1997

Smart Video Text: An Intelligent Video Database System

F. Kokkoras

H. Jiang

I. Vlahavas

Ahmed K. Elmagarmid
Purdue University, ake@cs.purdue.edu

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

Report Number:
97-049

SMART VIDEO TEXT: AN INTELLIGENT VIDEO DATABASE SYSTEM

F. Kokkoras
H. Jiang
Ioannis Vlahavas
Ahmed K. Elmagarmid
Elias N. Houstis

CSD-TR #97-049
October 1997

Smart VideoText: An Intelligent Video Database System

F.KOKKORAS* H.JIANG** I.VLAHAVAS*[1] A.K.ELMAGARMID**[2] E.N.HOUSTIS**

* Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54006 GREECE
{kokkoras, vlahavas}@csd.auth.gr

** Computer Science Department
Purdue University
West Lafayette, IN 47907 USA
{jiang, ake, enh}@cs.purdue.edu

Abstract

In this paper, an intelligent annotation-based video data model called *Smart VideoText* is introduced. It utilizes the Conceptual Graph knowledge representation formalism to capture the semantic associations among the concepts described in text annotations of the video data. The aim is to achieve more effective query, retrieval and browsing capabilities based on video data's semantic content. Finally, a generic and modular video database architecture based on Smart VideoText data model is described.

Keywords: video databases, Conceptual Graphs, information retrieval

---

[1] The author was on sabbatical leave at Purdue University when this work was carried out.

[2] Partially supported by a grant from the Intel Corporation

1. Introduction

Video data, with its unique characteristics such as huge size, rich content, the temporal and spatial nature, has posed many interesting challenges to the multimedia database research community. One critical problem is the modeling of video data for effective content-based indexing and user access capabilities , such as query, retrieval and browsing.

Video data can be modeled in terms of its visual content (such as color, motion, shape, intensity etc.) [14], audio content [2,18,23] and semantic content in the form of text annotations [12]. Because machine understanding of the video data is still an unsolved research problem, text annotations are usually used to describe the content of video data according to annotator's comprehension and the purpose for the expressed data. Although such content descriptions may be biased and incomplete, they still depict the amount of semantic content that can not be obtained by current image processing or voice recognition techniques. Video annotation is suitable for applications such as distance learning and news video databases, but is inadequate for surveillance video databases, where data access through face recognition is often performed. Visual content-based models are more appropriate for such applications [5].

Although they [9,12,30] take into consideration the temporal characteristics of video data which also exist in its annotations, some video annotation-based models fail to model and use the semantic relationships among the concepts expressed in the video and its annotations. A video database user may want to browse the video data in terms of the temporal relationship between video clips as well as the semantic association among them. The importance of capturing semantic associations in a video data model is increase because of the fact that human beings always have multiple expressions or terms for the same or similar semantics. Such capability in a video database is highly desired, but has not been explored so far.

In this paper, we introduce *Smart VideoText*, an intelligent, annotation-based video data model. The goal is to achieve more effective query, retrieval and browsing based on the semantic associations existing in the video data. Here, the effectiveness is the degree of relevance of the query results to what the user had in mind. This is achieved by combining VideoText video data model [12] with the Conceptual Graph (CG) knowledge representation formalism [25] to model the semantic associations among video annotations. The semantic association knowledge as well

as other information about video data is encoded as Conceptual Graphs (CGs) and, along with a proper inference mechanism, is used to support more flexible and effective video data access.

The paper is organized as follows: Section 2 introduces the current trends in Information Retrieval systems, the Conceptual Graph knowledge representation formalism and the Knowledge-Based Information Retrieval model based on this formalism. Section 3 gives a summary of current approaches to video data modeling. It also motivates and outlines our approach. The Smart VideoText data model is presented in Section 4, and in Section 5, different video data access methods supported by the model are discussed, using some examples. A system architecture based on Smart VideoText model is proposed in Section 6. Finally, Section 7 concludes the paper and suggests possible future work.

2. Knowledge-based Information Retrieval

2.1 Trends in Information Retrieval

Information retrieval (IR) systems retrieve textual documents using a partial match between a user query and a proper document representation. There are three fundamental issues in building an IR system: the choice of document representation, the query formulation, and a suitable ranking function that determines the extent to which a document is relevant to a query. Different categories of retrieval models have been developed based on how these issues are addressed in the system [7]. There are four main IR models: *Boolean*, *cluster-based*, *probabilistic* and *vector-space* models [17,21,31]. The efficiency of a model is usually determined by the so-called *precision* and *recall* measures. Precision is defined as the proportion of a retrieved set that is actually relevant. Recall is the proportion of all relevant documents that are actually retrieved.

Various studies have attempted to combine the strengths of the above categories of retrieval models and formulate a unified approach [3,19,22]. Recent research suggests that significant improvements in retrieval performance requires techniques that, in some sense, "understand" the content of documents and queries [21,29] and can thus infer probable relationships between documents and queries. From this point of view, information retrieval is an inference process. The aim of the inference is to facilitate flexible matching between the terms or concepts mentioned in queries and those contained in documents. The poor match between the vocabulary used to express queries and that used in the documents appears to be a major cause of poor recall. Recall

of an IR model can be improved by using domain knowledge in the concept representation and query processing without significantly degrading precision.

One knowledge-based approach to information retrieval is described in [13] where the Conceptual Graph knowledge representation formalism is used to encode the semantic associations among the various terms within a collection of text documents. This approach leads to a knowledge-based hypertext model in which the links between text documents are implicitly defined through the relationships among the elements (concepts, conceptual relations, concept-type hierarchy etc.) of the knowledge base (KB). This approach could be considered to be a knowledge-based IR model (KB-IR).

One advantage of using CGs in IR is that the complex semantic nets need not be manually created to encode the semantic associations of the documents. CGs are already semantic nets; the conceptual relations correspond to the links and the concepts correspond to the nodes. Moreover, there are techniques that create CGs automatically through document parsing, at least for highly organized documents such as programming language manuals [19] or for cases where a controlled vocabulary is used.

2.2 Conceptual Graphs: Primitives and Definition

The CG model of knowledge representation is not only a general framework for expressing natural language semantics but also a practical way to express a large amount of pragmatic information through assertions. All of the algorithms are domain-independent and every semantic domain can be described through a purely declarative set of CGs. In addition, this model can present high-order logical relations that are difficult to represent in a simple first-order logical formalism [6].

The elements of CG theory [25] are *concept-types*, *concepts* and *conceptual relations*. Concept-types represent classes of entity, attribute, state and event. Concept-types can be merged in a lattice whose partial ordering relation < can be interpreted as a categorical generalization relation. Thus, CAR<VEHICLE represents that CAR is a type of VEHICLE. A concept is an instantiation of a concept-type and is denoted by a concept-type label inside a box or between brackets. To refer to specific individuals, a referent field is added to the concept. Allowable kinds of referent fields include: generic ([book:*] - a book); individual ([person: John] - John); generic set ([book:{*}] - books); counted set ([book:{*}@3] - three books). Other

types of referent fields are also allowed. Conceptual relations show the relation between concepts. Each relation is constrained to which concepts it can connect. As for the concepts, there should be a pre-defined but expandable set of relation-types in any given system. A conceptual relation is denoted as a relation label inside a circle or between parentheses.

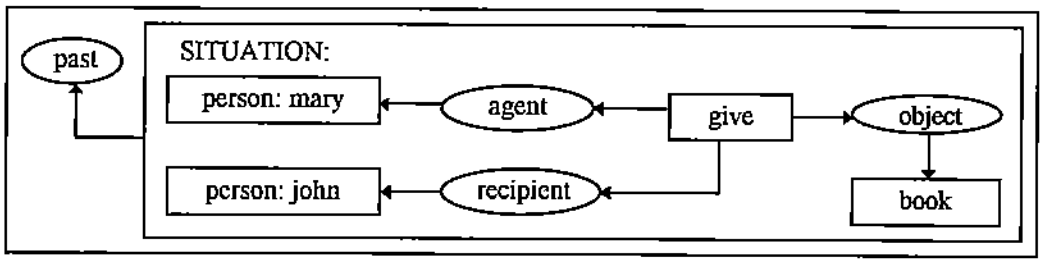

Figure 1: Conceptual graph of the sentence "Mary gave John a book".

A CG (Figure 1) is a finite, connected, bipartite graph. The nodes of the graph are either concepts or conceptual relations. Each relation is linked (only) to its requisite number of concepts, and each concept to zero or more relations. A CG represents information about typical objects or classes of objects in the world and can also be used to define new concepts in terms of old ones.

In the CG formalism, every context (situation, proposition, etc.) is a concept. Thus, contexts are represented as concepts whose referent field contains a nested CG. We will refer to these kinds of concepts with the term *contextual concepts*. A number of operations (canonical formation rules) are also defined on CGs, by which one can derive allowable CGs from a *canonical basis* [25]. The canonical basis is a set of CGs from which all other CGs are derivable and it is manually constructed. The canonical formation rules enforce constraints on meaningfulness; they do not allow nonsensical graphs to be created from meaningful ones. The canonical formation rules are:

- *Copy* creates a copy of a CG.
- *Restriction* takes a graph and replaces any of its concept nodes either by changing the concept-type to a subtype or adding a referent where there was none before.
- *Joining* joins two graphs with a common concept over it, to form a single graph.
- *Simplifying* removes any duplicate relations between two concepts.

Other operations on CGs include:

- *Contraction* tries to replace a sub-graph of a given CG with a single, equivalent concept (or relation), using the CG definition of this concept (or relation).

- *Expansion* is the opposite of the contraction operation.
- *Maximal Join* is a join of two CGs followed by a sequence of restrictions, internal joins and simplifications so that the maximum amount of matching and merging of the original graphs is achieved. The maximal join can be regarded as a generalized unification operation [10]. Note that the term usually refers to both the resulted CG and the operation.

Restrict, join and simplify operations are depicted in Figure 2. The maximal join, as a compound operation, is straightforward.

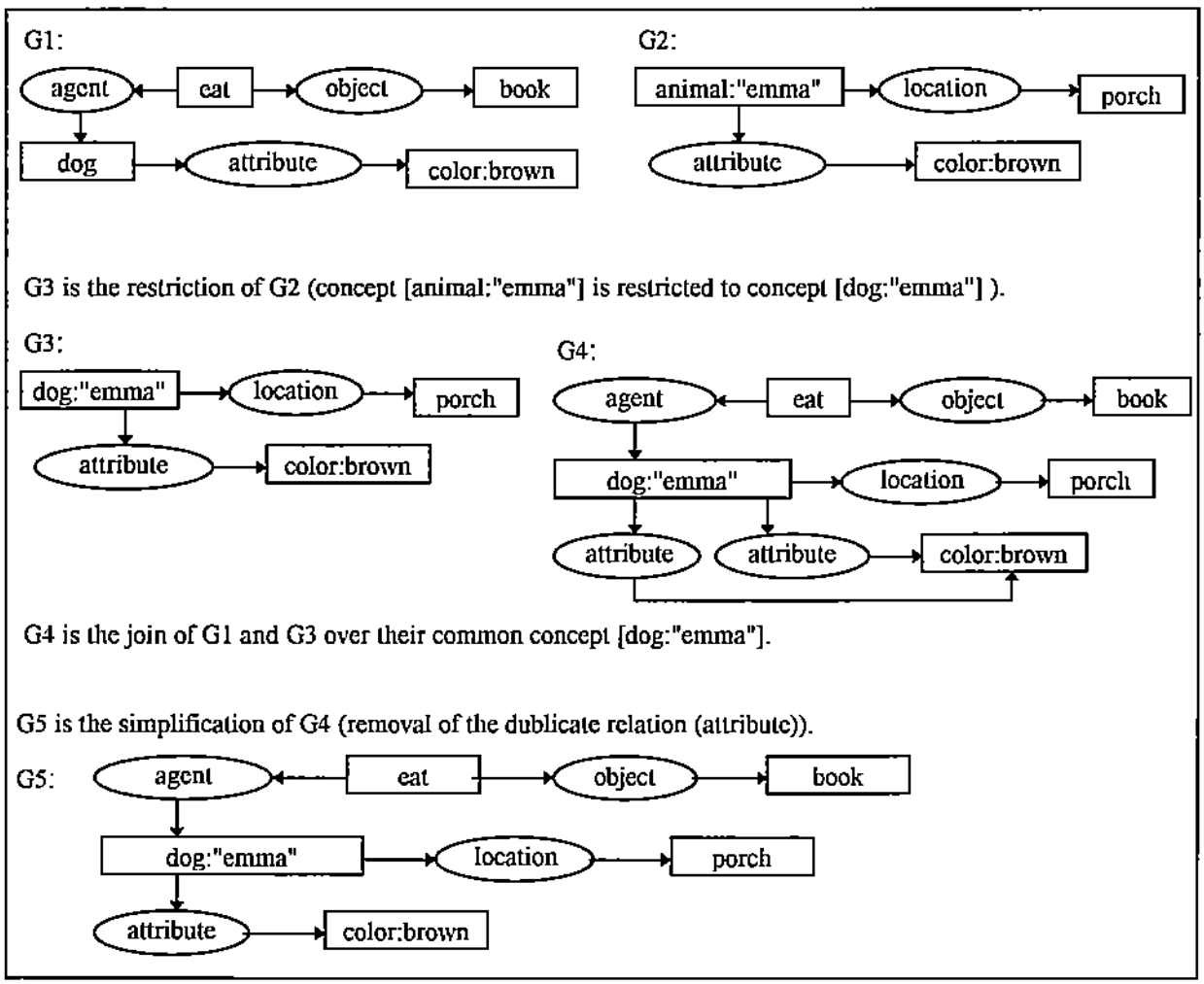

Figure 2: Examples of restrict, join and simplify operations.

Deduction with CGs is performed via a top-down resolution algorithm. A query expressed as a CG can be answered either by a direct matching with a CG of the knowledge base or an indirect matching using inference rules. The classic Sowa's *Oz* example [25] is given in Figure 3.

A person is a citizen of Oz if and only if any of the following conditions are true:
1) This person is born in Oz.
2) One of his parents is a citizen of Oz.
3) This person is naturalized in Oz.

clause (inference rule):
[CITIZEN: *x] ← (MEMB) ← [COUNTRY: 'Oz']
⇐ [PERSON: *x] ← (AGNT) ← [BORN] → (LOC) → [COUNTRY: 'Oz']
clause (inference rule):
[CITIZEN: *x] ← (MEMB) ← [COUNTRY: 'Oz']
⇐ [PERSON: *x] ← (CHLD) ← [PERSON: *y] and [CITIZEN: *y] ← (MEMB) ← [COUNTRY: 'Oz']
clause (inference rule):
[CITIZEN: *x] ← (MEMB) ← [COUNTRY: 'Oz']
⇐ [PERSON: *x] ← (RCPT) ← [NATURALIZE] → (LOC) → [COUNTRY: 'Oz']
clause (fact)
[PERSON: 'Tinman'] ← (AGNT) ← [BORN] → (LOC) → [COUNTRY: 'Oz']
↓
(CHLD) → [GIRL: 'Dorothy']

goal clause : "Who is a citizen of Oz country ?" ⇐ [CITIZEN] ← (MEMB) ← [COUNTRY: 'Oz']
result :
[CITIZEN: 'Tinman'] ← (MEMB) ← [COUNTRY: 'Oz']
[CITIZEN: 'Dorothy'] ← (MEMB) ← [COUNTRY: 'Oz']

Figure 3: Deduction with CGs. The classic Sowa's "Oz" example.

3. Video Data Models

A video data model is a representation of video data based on its characteristics and content, as well as the applications it is intended for. Some desired capabilities of a video data model include multi-level video data abstraction; video annotation support; spatial and temporal relation support and video data independence. Video data models can be based on the idea of video segmentation or video annotation layering [5].

3.1 Segmentation-based Models

For a given video stream, segmentation-based models [8,9,27,28,32] usually use scene change detection algorithms [11] to parse and segment the video stream into a set of basic indexing units called *shots*. These shots can be matched or classified against a set of domain specific templates (patterns) to extract higher level semantics (such as CNN Headline News) and structures (such as

episode) contained in the video. A hierarchical video stream representation can thus be built through this process.

The main advantage of these models is that the video indexing process can be fully automated. But they also have some disadvantages, such as:

- lack of flexibility and scalability since video streams are pre-segmented by the scene change detection algorithms;
- unreliable template matching because the similarity measure between two frame images is ill-defined and limited;
- lack of applicability for video streams that do not have well defined structure. For example, in a video stream of a class lecture, where there is no clear visual structure in terms of shots, segmentation using scene change detection algorithms is difficult; and
- very limited semantics can be extracted from template matching process which are application specific.

3.2 Annotation-based models

The basic idea of the annotation-based models is to layer the content information (depicted by annotations) on top of the video streams, rather than segment the video data into shots. Each annotation is associated with a logical video segment, which is, in general, a subset of a video stream and is defined by the starting and ending frame numbers. Logical video segments can be overlapped or nested [15,16,30] in an arbitrary manner.

One of the earliest annotation-based model is the stratification model proposed by Davenport et al. [4,24], which is based on the idea of annotation layering. Other annotation-based models like, the generic video data model [9] and the Algebraic Video model [30] have been developed since then. The annotation layering and the notion of logical video segment have the following advantages:

- Various video access granularities can be supported, i.e. annotations can be made on logical video segment of any length, from a single frame to the whole video stream.
- The annotation information can be easily handled by previously existing, sophisticated information retrieval and database techniques.
- Various annotations can be linked to the same logical segment of video data to provide multiple views of the video data [30].

- Annotations can be added and deleted independent of the underlying video streams. This supports dynamic and incremental creation and modifications of video annotations.
- Video retrieval and queries based on semantic content can be performed at a level that current image processing and computer vision techniques can not achieve.

An annotation-based video data model called *VideoText model* was recently proposed by Jiang et al [12]. The model integrates IR and video databases to support free text video annotations. This model is no longer limited by attribute/value pair type of annotations, and it incorporates all possible interval relationships between two logical video segments. It not only supports IR operators like AND, OR, NOT and ADJ, but also supports user queries that are based on interval relations (OVERLAP, AFTER etc.) among logical video segments. The VideoText model is briefly introduced in the next section.

3.3 VideoText Data Model

The VideoText model [12] is a video data model based on the concepts of logical video segment and free text video annotations with arbitrary mapping between them. The model is defined as

$$VT=(V, T, Map)$$

where V is a set of video streams $v^i \in V$, $i=1,...,n$, and also a set of logical video segments; a *logical video segment* is a consecutive video frame sequence $[f^i_j, f^i_{j+1}, .., f^i_k]$ that has meaning by itself for indexing and query purposes. A logical video segment can span from a single frame to the whole video and can also overlap with other logical video segments in arbitrary ways. T is a set of video annotations that are free text segments that describe the content of the corresponding logical video segments. Map is the mapping that defines the relationship between logical video segments and video annotations. The relationship is, in general, many-to-many, that is, a logical video segment can correspond to multiple annotations (different user views), and an annotation can be assigned to several logical video segments (annotation sharing).

Since V and T are relatively independent from each other, the model also supports dynamic creation and incremental updates of the video annotations. The VideoText model can be used to implement a video database system with a modular architecture that consists of a video data storage sub-system, an information retrieval sub-system and an integrator sub-system which is used to coordinate the other two sub-systems [12].

Also defined by the VideoText model is a query language which supports queries based on semantic content (video annotation documents) of video data. This query language uses Boolean (AND, OR and NOT), temporal (ADJ) and interval operators (DURING, OVERLAP etc.). The VideoText query language enables users to formulate complicated video queries which involve temporal characteristics of video data. One such query expression can be [12]:

((((George AND chair) AND (chair ADJ window)) OVERLAPS raining) BEFORE Chicago)

This query expression requires finding video segments whose annotations contain "George", "chair", "window", with "chair" appearing before "window". Also, the segments should overlap with a video clip that contains the annotation term "raining" and precedes a video clip with the annotation term "Chicago".

3.4 Our Approach

Although the VideoText model [12] captures the temporal characteristics of the video data, it does not consider the semantic associations among video annotations. A similar problem exists with other annotation-based models [9,30]. Semantic association, which refers to the complex relationships between different concepts and words, is important due to the fact that human beings tend to express the same or similar meaning in multiple ways or through different concepts and/or words. Modeling such associations will greatly improve the effectiveness of the video query, retrieval and browsing capabilities. For an example, a user query such as "Clinton AND Welfare" is semantically related to a logical video segment which annotated with "... the President ... Medicare ...". The reason is that the terms "President" and "Clinton" refer to the same person, and "Medicare" is actually one type of "Welfare".

Our approach proposed in this paper is a knowledge-based video data model called *Smart VideoText*. It extends the VideoText model [12] by utilizing the conceptual graph knowledge representation formalism to capture the semantic associations among the concepts described in the video annotations. This will enable the basic annotated-based video data model to provide functionality beyond the simple operator-based video query and retrieval. Namely, the CG layer allows hypertext-like browsing and natural language querying on the video data based on the semantic relationship among video clips or logical video segments. Furthermore, the effectiveness of the operator-based retrieval will be greatly improved because the CG layer will provide semantic term matching. The details of our approach are presented next.

4. The Smart VideoText Model

This section describes a new Video data model, called *Smart VideoText*. It extends the ideas of the VideoText model [12] by applying knowledge-based IR techniques to capture and model the semantic associations in the video annotations and support intelligent video query, retrieval and browsing based on the semantic content of the video data.

4.1 Definition Of The Model

The Smart VideoText can be defined as a 5-tuple:

$$\texttt{SmartVideoText} = (V, \texttt{Map}_1, T, \texttt{Map}_2, KB)$$

where:

V is a set of video streams ($v^i \in V$, $i=1,...,n$) and also a set of logical video segments. It can be simply denoted as $[v^i_{j-k}]$ where j and k are the starting and ending frames respectively, of the logical video segment which is part of the video stream i.

T is a set of video annotations ($t^i_m \in T$) which are text segments that describe the contents of the logical video segments of video stream i.

The mapping relation $\texttt{Map}_1$ defines the correspondence between annotations and logical video segments. For instance, the mapping $\texttt{Map}_1([t^3_5], [v^3_{40-980}])$ defines an one-to-one relationship between video annotation 5 of video stream 3 and the frame sequence 40 to 980 of the same video stream. This mapping is, in general, a many-to-many relationship since, in this way, an annotation could be shared among logical video segments or a logical video segment could be multiple annotated (perhaps by different users) to fulfill different application needs and to reflect possible different understanding of the same video data. In addition, logical video segments implicitly define the temporal relationship between any given two annotations within the same video.

KB is the knowledge base that is encoded according to the CG knowledge representation scheme. It includes system knowledge, application knowledge and domain knowledge. More details are given in Section 4.2.

$\texttt{Map}_2$ is a mapping relation that maps a subset (application knowledge) of the KB to video annotations T since only this knowledge is directly derived from the video annotations. $\texttt{Map}_2$ is a many-to-many relationship according to the mapping scheme described in Section 4.2.

4.2 Video Knowledge Representation

The KB in the Smart VideoText model logically consists of three parts, namely system, application and domain knowledge:

- *system knowledge*: this mostly includes rules about how to handle CGs (formation rules, inference rules etc.),
- *application knowledge*: this is the knowledge related to the content of the video database; it is derived from video annotations,
- *domain knowledge*: this includes knowledge related to but not explicitly defined in the video database, such as the type hierarchy and concept definitions which are expressed as CGs.

Although all three parts are used during knowledge-based video access, only the application knowledge is involved in the $\texttt{Map}_2$ mapping since only this knowledge is explicitly derived from video annotations. Extend of the automation of this knowledge derivation process varies, and it could be considered, in general, as a semi-automatic one [26]. Usually, the more organized and better structured the source text documents are, the more automation could be achieved.

The basic building blocks in the knowledge base are CGs, concepts and conceptual relations. Following is the definition of these terms in the Smart VideoText data model using a Prolog-like notation. Conceptual Graphs are defined as predicates of the form:

cg(ID, RelationList)

where ID is a unique identifier associated with this CG and RelationList is a Prolog list that stores the conceptual relations of the specific CG.

A conceptual relation is defined as:

cgr(RelationName, ConceptIDs)

where ConceptIDs is a list of concept identifiers that this specific conceptual relation joins and RelationName is the name of the conceptual relation.

Concepts are represented as predicates of the form:

cgc(ID, VideoAnnotationIDList, Context, ConceptName, ReferentField)

where ID is a unique identifier associated with this concept, VideoAnnotationIDList is a Prolog list of identifiers of the video annotations that contains this concept, Context is either normal for the case of normal concepts or special for the case of contextual concepts. ConceptName is the type-name of a normal concept or the context name of a contextual

concept (situation, proposition etc.). `ReferentField` is a Prolog list that holds the referent field of the specific concept. For instance, for the concept `[Book:{*}@3]` (three books), the `ReferentField` has the value `[{*}@3]`.

The concept and the relation type hierarchy are defined using `is_a` relationships. For example, `is_a(car,vehicle)` denotes that car is a kind of vehicle. Note that other kinds of hierarchical relations such as equivalent, scope and association are not required to be explicitly defined since they can be expressed using relations of the conceptual graph formalism.

The above Prolog oriented notation can handle complex CGs without information repetition since concepts are indexed separately. Furthermore, a CG can be traversed starting from any of its concepts [20].

```
cgc( 10, [va_id(4,5)], normal, person, [`mary`] ).
cgc( 20, [va_id(4,5)], normal, person, [`john`] ).
cgc( 30, [va_id(4,5)], normal, give, [ ] ).
cgc( 40, [va_id(4,5)], normal, book, [`*`] ).
cgc( 50, [va_id(4,5)], special, situation, [100] ).
cg( 100, [ cgr( agent, [30,10] ), cgr( recipient, [30,20] ), cgr( object, [30,40] ) ] ).
cg( 110, [ cgr( past, [50] ) ] ).
```

Figure 4: "Mary gave John a book" expressed in the notation of the KB in the Smart VideoText model.

It is obvious that the Map_2 mapping scheme is included into the above representations of the various elements in the CG formalism. The `VideoAnnotationIDList` argument in the `cgc/5` predicate[3], is a list that holds this information as tuples of the form `va_id(i,j)`. Such a binary tuple in a `cgc` means that, the concept represented by this `cgc` is mapped to the video annotation `j` of the logical video stream `i`. This argument has the value `va_id(-1,-1)` and `va_id(0,0)` for concepts belonging to the system and domain knowledge respectively since these are not derived from video annotations.

In Figure 4, we illustrate the concepts, conceptual relations and CGs for the example of Figure 1 in the way they are stored into the KB of the Smart VideoText model. Note that context is a flag; when it has the value *special*, then the next argument defines the context of a concept (*situation* here) and not a usual concept-type name.

[3] The symbol `/N` after the name of a Prolog predicate denotes its arity (number of arguments).

5. Video Data Access in Smart VideoText Database

Smart VideoText model supports multiple-strategy, knowledge-based video data access which includes operator-based queries, natural language queries and hypertext-like video browsing, based on semantic associations provided by the KB.

5.1 Operator-based Video Queries

An operator-based user query is an expression formed through terms and zero or more operators [12]. Terms are the strings that a user wants to find in the annotations of the target logical video segments. The operators define the relationships between the terms and can be Boolean (AND etc.), temporal (ADJ) and/or interval operators (AFTER etc.). The detailed description and syntax of such operator-based query expression can be found in [12]. A video query in the Smart VideoText model, which is an extension of the VideoText query, is defined as:

`Q(Expression, Scope, KBFlag, MaxResults)`

where `Expression` is the Smart VideoText query expression described above, `Scope` defines the granularity of the answer and can be video streams (`v`) or logical video segments (`s`), `KBFlag` is a flag (`true/false`) denoting whether to use the knowledge base or not, and `MaxResults` is the maximum number of returned query results.

Notice that if the `Expression` contains interval operators, then the `Scope` is always logical video segments (`s`). This stems from the nature of this category of operators.

The utilization of the KB is expected to increase the effectiveness of the operator-based video data access. This is due to the fact that exact term matching suffers in the following cases:

- *poor recall*: in this case, useful logical video segments are not retrieved because their annotations contain a synonym or something semantically similar rather than the exact terms presented in the video query, and
- *poor precision*: too many video annotations contain the given term(s), but not all the retrieved logical video segments are actually semantically relevant to the video query.

The use of the KB (particularly the concept and relation type hierarchy) in the video data model alleviates the poor recall problem. The following example will demonstrate why: an attempt to match the term `car` with video annotations containing the term `BMW-325i` will succeed because a concept `[car:BMW-325i]` exists in the KB, which implies that `BMW-325i` is a

specific individual of the concept-type car. A discussion about how the KB can be used to improve the video query precision is given in Section 5.2.

The existence of the KB in the Smart VideoText model provides two modes of operator-based query evaluation: *raw term matching* and *semantic term matching*. The mode is determined by the value of the `KBFlag` argument (`false` or `true`, respectively) in a Smart VideoText query expression. The first case is straightforward: a text annotation (and its logical video segment) answers to a video query if it contains all of the query terms and satisfies all of the constraints introduced by the operators. In the second case, a *similarity measure* is required to be able to determine the extent to which two concepts may be labeled "similar". Calculation of the similarity of two concepts depends upon the prior identification of appropriate "sources" of similarity associated with the concepts. Such a source is the concept-type hierarchy. Having the form of a lattice, the type hierarchy of a CG system allows the computation of the distance between related nodes in the lattice, which is often called *semantic distance*. The semantic distance can be used as the measure to rank the results of an operator-based video query.

For a semantic match, if two concepts are syntactically different from each other but they belong to the same branch of a concept-type hierarchy, the more specific one can be repeatedly generalized to shorten the semantic distance between them. Between two semantic matches, the one that uses fewer successive generalizations is more important since the semantic distance between this one and the matching concept is shorter. Thus it has higher rank. We restrict to generalization since specialization does not always preserve truth [25]. For example, specializing the concept [`building`] to [`hospital`] is not correct in all contexts.

The operator-based query evaluation is performed by recursively decomposing it into sub-expressions and processing them along the lines described in [12]. The only difference is that in the Smart VideoText model, if a query term does not match a term in a video annotation, a try to generalize the query term using the concept-type hierarchy is performed. On successful generalization, the matching try is repeated, this time for the term which is the result of the generalization. This is depicted in Figure 5. It is up to the user to decide whether to use raw or semantic term matching in a video query. It would be better to have the semantic term matching method automatically invoked when the raw term matching fails to give results or the number of results is below a user defined threshold.

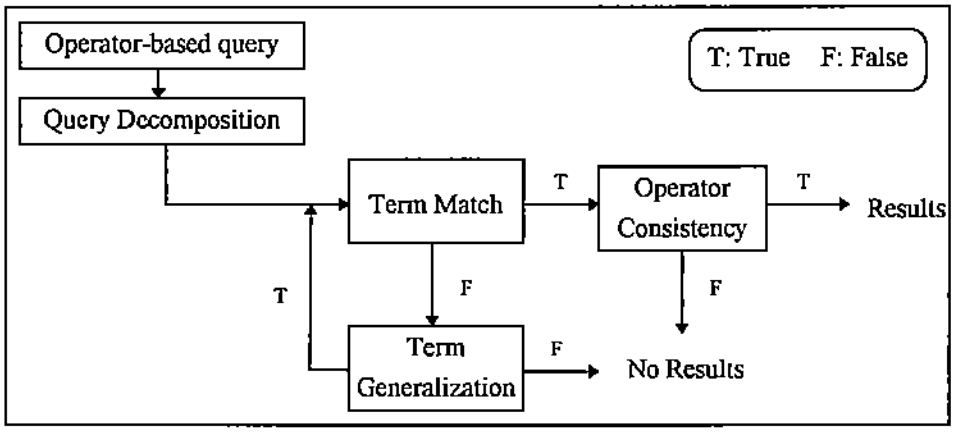

Figure 5: Block diagram of the term matching part of the operator-based query processing. The "Term Generalization" block provides the semantic term matching.

Notice that the effect of an operator-based query to a Smart VideoText video database (SVTDB) is the derivation of a ranked subset SVTDB'. This allows users to recursively refine their queries. Stated it in another way, the query evaluation process is recursive.

5.2 Natural Language Video Queries

Since the derivation of CGs from natural language text can be automated, it is possible to allow video queries to be expressed in natural language. Such a query is converted into a query CG which is then compared to the conceptual graphs in the KB using CG-based inference rules. Depending upon the application, it is possible to have predefined query templates and let the user to construct queries by filling in "slots". A query template corresponds to a semi-structured CG in which the user is requested to precisely define it by either specifying one or more concepts or replacing a generic referent marker of a concept with a more specific one.

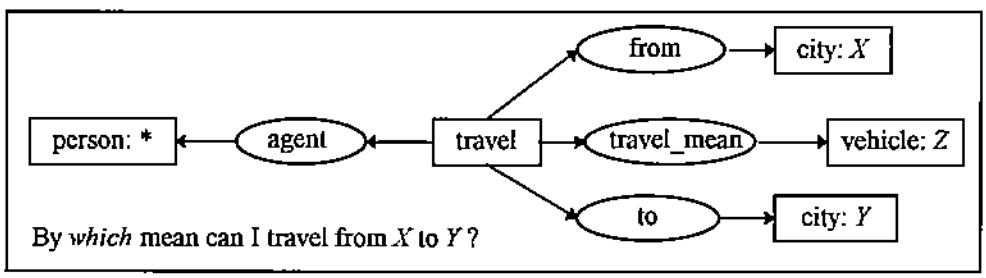

Figure 6: A query template expressed in natural language and in CG form.

An example of a query template coupled with a semi-structured CG is illustrated in Figure 6. The user fills in the empty fields of either a natural language query or a CG expression (these empty fields are represented as variables in italics in Figure 6) to make the question complete. In

both cases, the result is a CG that the system tries to "prove" with the elements (CGs, concepts etc.) of the KB. Upon successful matching, the query CG will be augmented (possibly a modified version of it due to an expansion operation, for example) with icons that will allow the user to invoke a video player to play the related video clip(s). This is possible through the $\texttt{Map}_1$ and $\texttt{Map}_2$ mapping schemes which map CGs and concepts to logical video segments.

It is a good idea to distinguish between *directly related* videos and *indirectly related* ones. A logical video segment is said to be directly related to a user query, if its derived knowledge matches syntactically the information need of the user, that is, without any use of other knowledge. Consequently, for indirectly related logical video segments, additional knowledge has been used (for example, the concept-type hierarchy) to complete the match which, as a result, is a semantic match. The Dempster-Shafer theory [1] can be used to combine various "sources" of similarity evidence associated with CGs to compute the *total similarity* between two CGs. Such "sources" can be the maximal join, matching between relations, concepts and conceptual referents, the concept-type hierarchy, and the ratio of arcs in the maximal join CG to the total number of arcs in the larger of the two CGs that participate in the maximal join operation [1] (the size of a CG is equal to the number of arcs that join the building blocks (concepts and relations) of it). The contribution from any of the above sources of evidence of similarity can be equal or weighted. In general, the total similarity can be defined as:

$$\texttt{TotalSimilarity} = w_1*\texttt{Evidence}_1 + w_2*\texttt{Evidence}_2 + \ldots + w_N*\texttt{Evidence}_N$$

where w_i are the weights and $\Sigma w_i=1$. According to [1], this combined similarity allows for superior retrieval to that obtained by any individual form of evidence. This `TotalSimilarity` value can be used to rank the results in the case of natural language video queries.

Since the end user is assumed to be unfamiliar with CG representation formalism, a sophisticated graphical user interface (GUI) is necessary to make natural language video query formulation as user friendly as possible. In a real application with a clearly defined content, a pre-existing, controlled vocabulary can further alleviate the problem of user unfamiliarity with the CG formalism.

The KB can be used in both operator-based and natural language queries to improve precision. If a Smart VideoText query has produced too many returns, it is possible to use this knowledge to construct system queries, that is, queries constructed by the system and presented

to the user, to improve the precision of the returned logical video segments. For example, if searching for video clips about cars has returned too many logical video segments, then, the system should try to collect various instances of cars and ask the user if he/she is interested in any particular brand name or model. According to the notation we have adopted for the concepts such an operation is relatively easy, since all the information required for this task is enclosed into the referent fields of concepts.

5.3 Video Browsing

All existing video browsing techniques are based upon the visual content of video data; although useful in certain applications, they fail to follow the semantic content of the video data. The Smart VideoText model supports an efficient and effective way for the user to browse a large collection of video data as well as the corresponding video annotations based on their semantic content. This is achieved by using the semantic associations between the concepts of video annotations and dynamically representing them as "hyperlinks" between the corresponding logical video segments. The users can follow these links when they browse the video database.

A term in an video annotation, which is also a concept in the KB, could serve as a hypertext-like link that points to a logical video segment that contains the same concept or a semantically similar one. For instance, two video annotations containing the term "Mars" are directly associated (so are their corresponding logical video streams), but any of them is indirectly related to a third annotation containing the term "Pathfinder" and vice-versa. Thus, semantic associations between the concepts in the KB can be the vehicle to browse over semantically related video clips and video annotations. This functionality is supported by the Map_1, Map_2 and KB components in the Smart VideoText model.

Figure 7 gives an example of the browsing capabilities of the Smart VideoText model. Let's assume that there are three logical video segments, namely LVS_i, (i=1,2,3) with the corresponding video annotations VA_i (i=1,2,3). CG_i (i=1,2,3) are conceptual graphs derived from those annotations and they are part of the KB. Let's assume that the first set of data LVS_1, VA_1 and CG_1 is presented to the user. The user can either ask the system to find information related to one of the designers (say designer b) or ask for information related to skyscrapers. All can be done in a hypertext-like fashion, i.e. some elements displayed have the potential to function as hypertext links to other video data. The existence and functionality of these hyperlinks are provided by the

model rather than explicitly defined by the database user. Hence, these KB-based links are generated "on the fly" rather than explicitly predefined and fixed as in hypertext documents like the HTML pages on the World Wide Web.

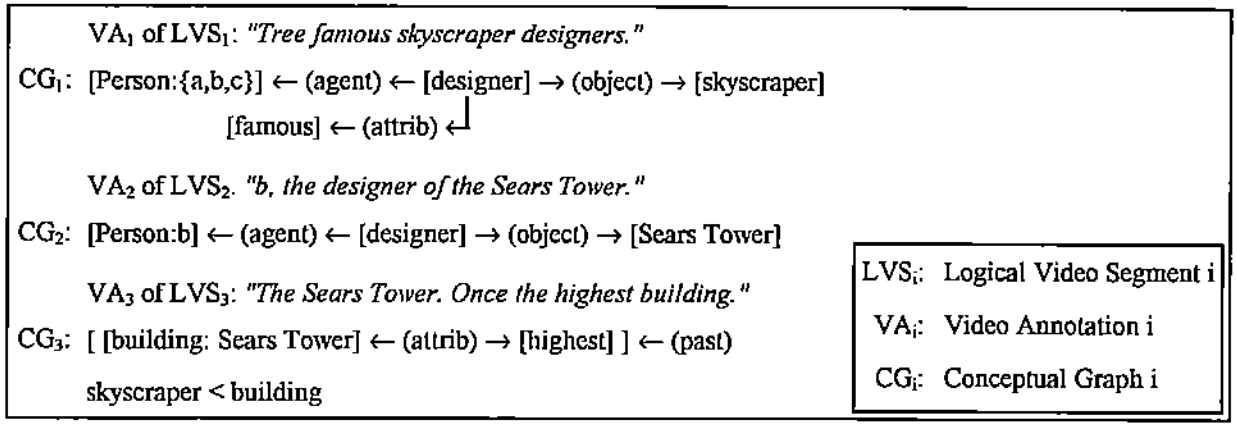

Figure 7: Video Annotations (VA_i) of some logical video segments (LVS_i) and their corresponding CGs.

In the above example, a request for additional or related information about designer b, can lead the user to VA_2 or CG_2 through a direct concept matching over `[Person]` and a membership check over the referents. At any time, the user can also ask the system to play the corresponding logical video segment, LVS_2.

On the other hand, an information request about skyscrapers will drive the user to the third set of data (VA_3, CG_3 and LVS_3) given the fact that the concept-type hierarchy in the domain KB includes a relation like `is_a(skyscraper,building)`. Moreover, from the third set of data, the system can suggest the second set as very related information.

6. Smart VideoText System Architecture

The Smart VideoText model introduces a modular architecture for implementing video databases. Although video segments, text annotations and application knowledge are related in the data model, they can be managed by a different component of a Smart VideoText based system. Such a system is outlined in Figure 8 and its components are described latter on.

- Databases: The *Video database* stores the video data while the *Video Annotation database* stores annotations of the video data. Each annotation is stored together with a reference to the logical video stream it annotates.
- Knowledge: The *system knowledge* includes the canonical formation rules and knowledge

that is supposed to be application independent. *Application knowledge* is consisted of CGs, concept-types, concepts and relations derived from video annotations. Although multiple application knowledge bases can be allowed (one for each application), only one of them is used at any time. This means that knowledge consistency checking is performed at the application level. *Domain knowledge* includes the concept-type hierarchy together with the concept and conceptual relation definitions of the various concepts and conceptual relations used in the application. In other words, it includes all the knowledge related to the application, but not explicitly defined in the video annotations.

- A *kernel* integrates the various modules into a single video database system. The kernel includes a Prolog inference engine that is responsible for the inference.
- Several modules for the user/author including a *parser*, *a semantic interpreter*, a *query handler*, a *knowledge manager*, a *linkage assistant*, an *editor* and a *video player*.

The *parser* uses syntactic rules to generate parse trees corresponding to all (or the user selected) sentences of the video annotations, as well as to the queries expressed in natural language.

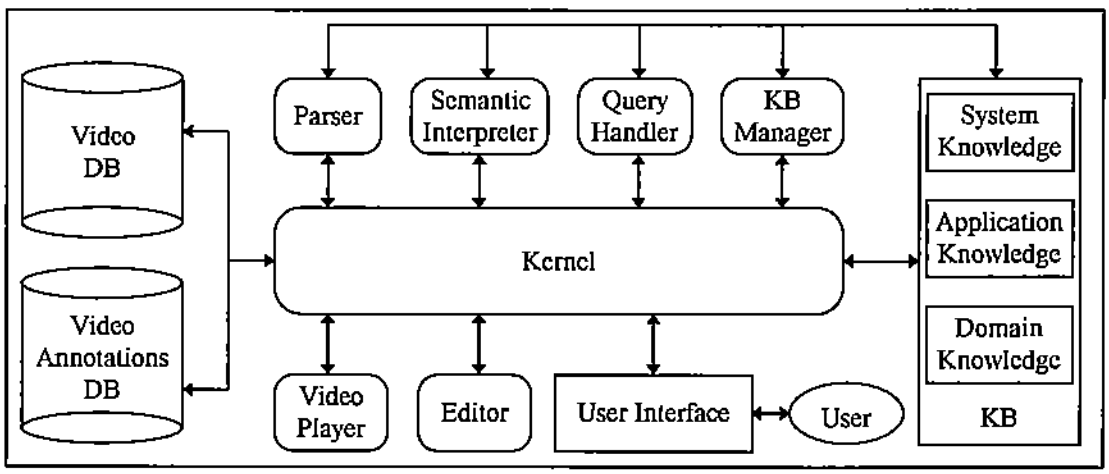

Figure 8: Smart-VideoText System Architecture

The *semantic interpreter* translates the above trees into CGs and, in the case of knowledge construction, asserts them into the KB if they fulfill the canonical formation rules. In the case of natural language query, the resulted CG is passed to the query handler module. In both cases, the semantic interpreter can be manually forced to abandon some parts of a sentence with no interest.

The *query handler* lets the user construct queries concerning the video database. A query is expressed as a CG and is constructed by the user either directly or indirectly. In the first case, the user selects the appropriate items (concepts and relations) from a combination of selection

components (menus, listboxes, etc.). It is also possible for the user to select a query from a set of previously defined query templates, expressed in natural language. For each of them, there exists a pre-constructed CG that is used to answer the query. Creating a CG-query indirectly means that the user expresses it in natural language and the proper system modules (parser, semantic interpreter) convert it into CG form. Operator-based query processing has already been discussed in Section 5.1.

The *knowledge manager* supports operations such as review of the KB and assertion of knowledge in any of the three parts of the KB. The review of the knowledge base is performed at CG level. The user opens the KB file and displays any CG in graphical form. CGs that satisfy the user-defined criteria can also be displayed. These criteria are filters that allow the user to inspect CGs with a common property. For example, in a geographical video database application, the user may want to see all the CGs concerning the population of capital cities in order to update the population numbers. Furthermore, the knowledge base can be modified if necessary. For an example, when a video annotation is updated or deleted, the corresponding CGs in KB are also modified or deleted by the system. In addition, knowledge consistency techniques could be applied to ensure that the knowledge base remains consistent after modifications.

The main advantage of the above architecture is its modularity which stems directly from the video content and the knowledge representation methods that are used. Using textual annotations to describe the content of the video data allows us to replace video data with any media form. Furthermore, the KB related modules are independent of each other as soon as they conform with the representation standards of CGs, which are well founded. Hence, advances in any of these domains could be easily utilized. Moreover, modularity introduces distributability, which makes it easier for SmartVideoText to be implemented as a distributed video database system

7. Conclusion and Future Work

In this paper, we have proposed an annotation-based video data model called Smart VideoText. This model utilizes the conceptual graph knowledge representation formalism to capture and represent the semantic associations among the concepts described in video annotations. The model can support, among others, dynamic video annotation manipulation, annotation sharing and

multiple interpretation of the same logical video segment. The model also includes a query language that supports complicated video queries based on the semantic content of the video data. Operator-based (Boolean, temporal and interval query operators) as well as natural language video queries are supported. The CG layer also allows transparent, hypertext-like content-based browsing on the video data. Furthermore, the functionality of the operator-based video query and retrieval is significantly enhanced because the CG layer provides semantic term matching. In this paper, we have concentrated mainly on the video data model and its knowledge-based aspects.

Some issues in the Smart VideoText need further research. For example, the naive representation of the `ReferentField` argument of the `cgc` tuple in Section 4.2, although provides the functionality required to establish the Smart VideoText model, could be further improved, since it is a crucial factor in the calculation of the semantic distance between two concepts. A more sophisticated treatment of `ReferentField` can be found in [20]. Another interesting work is to use the KB for guided navigation in a video database. The presence of the knowledge base and the inference mechanism allows for the implementation of intelligent agents in order to provide guided navigation in a video database. For example, such an agent will be able to incrementally create (as the user browses the Smart VideoText database) a user related KB (user profile) and use it to provide suggestions or to filter out some links that the user consistently ignores.

We are currently in the process of implementing the Smart VideoText along the lines of the system architecture proposed in Figure 8. The system will be web-based and provide distributed query processing and video access. More issues concerning implementation aspects, such as query optimization and various performance measurements, will be studied in the near future.

Applying the ideas of Smart VideoText model to other media forms (sound, graphics etc.) is fairly straightforward since our model deals directly with an intermediate, textual representation of the content of such data. Thus, extending our model to multimedia documents is feasible. This is very important since, given the popularity of the World Wide Web and the constantly increasing amount of multimedia data available online, knowledge-based IR systems will play important role in the near feature in the effort to implement more efficient and effective multimedia information systems.

References

1. Brkich F. and Wilkinson R., (1994). A Conceptual Graph Matching Algorithm for Information Retrieval. *First Australian Conceptual Structures Workshop.* University of New England, Armidale, New South Wales.
2. Brown M.G., Foote J.T., Jones G.J.F., Sparck K. and Young S.J., (1996). Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval. In *Proceedings of the Fourth ACM International Multimedia Conference.* pp.307-316
3. Croft W.B. and Harper D.J., (1997). Using probabilistic models of documents retrieval without relevance feedback. *J. Doc.*, 35: pp.285-295.
4. Davenport G., Smith T.G.A. and Pincever N. (1991). Cinematic primitives for multimedia. *IEEE Computer Graphics & Applications*, pp.67-74.
5. Elmagarmid A.K., Jiang H. and et al. (1996). *Video Database System: Issues, Products and Applications.* Kluwer Academic Publishers.
6. Fargues J., Landau M.C., Dugourd A. and Catach L., (1986). Conceptual Graphs for semantics and knowledge processing. *IBM Journ. of Research and Devel.*, 30:1, pp.70-79.
7. Fuhr N., (1996). Models for integrating retrieval and database systems. *IEEE Data Engineering Bulletin*, 19.
8. Hampapur A., Jain R. and Weymouth T., (1994). Digital video indexing in multimedia systems. In *Proceedings of the Workshop on Indexing and Reuse in Multimedia Systems.*
9. Hjelsvold R. and Midtstraum R. (1994). Modeling and querying video data. In *Proceedings of the 20th International Conference on Very Large Data Bases.*
10. Jackman M.K. (1988). The Maximal Join for Conceptual Graphs. In Sowa, J.F., Foo & Rao (Ed.), *Conceptual Graphs for Knowledge Systems.* Reading, MA: Addison-Wesley.
11. Jiang H., Helal A., Elmagarmid A.K. and Joshi A. (1996). Scene change detection techniques for video database systems. *ACM Multimedia Systems*, (accepted).
12. Jiang H., Montesi D. and Elmagarmid A.K. (1997). VideoText Database Systems. In *Proceedings of the Fourth IEEE Int. Conference on Multimedia Computing and Systems.*
13. Kokkoras F. and Vlahavas I., (1995). COMFRESH: A Common Framework for Expert Systems and Hypertext. *Information Proces. and Management*, Vol.31, No 4, pp.593-604.

14. Lei Z. and Lin Y.T., (1996). 3D Shape Inferencing and Modeling for Semantic Video Retrieval. In *Proceedings of Multimedia Storage and Archiving Systems.* pp.224-235
15. Little T.D.C. and Ghafoor A. (1993). Interval-based conceptual model for time-dependent multimedia data. *IEEE Transac. on Knowledge and Data Engineering*, 5(4) pp.551-563.
16. Little T.D.C., Ahanger G., Folz R.J., Gibbon J.F., Reeve F.W., Schelleng D.H. and Venkatesh D. (1993). A digital on-demand video service supporting content-based queries. In *Proc. of First ACM Intern. Conference on Multimedia*, Anaheim, CA, pp.427-436.
17. McGill M.J. and Salton G. (1983). *Introduction to Modern Infor. Retrieval.* McGraw-Hill.
18. Patel N.V. and Sethi I.K., (1995). Audio Characterization for Video Indexing, *IST/SPIE, Proceedings: Storage and Retrieval for Image and Video Databases IV*, San Jose, CA.
19. Petermann H. (1994). Automatische Generierung von Wissensreprasentationen aus Manualquelltexten. *Workshop der GI-Fachgruppe "Intelligente Lehr-/Lernsysteme"*, FAW an der Universitat Ulm, FAW-TR-940003.
20. Petermann H., Euler L. and Bontcheva K. (1995). CGPro - a PROLOG Implementation of Conceptual Graphs. Technical Report, University of Hamburg, FBI-HH-M-251/95.
21. Rijsbergen (van) C.J. (1997). *Information Retrieval.* Butterworths. 2nd Edition.
22. Salton G. and et al. (1983). Extended Boolean information retrieval. *Communications of the ACM*, 26, pp.1022-1036.
23. Smith M.A. and Hauptmann A. (1995). Text, speech and vision for video segmentation: The Informedia project. In *AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision.*
24. Smith T.G.A. and Davenport G. (1992). The stratification system: A design environment for random access video. In *Workshop on Networking and operating System Support for Digital Audio and Video*, Association of Computing Machinery.
25. Sowa J.F. (1984). *Conceptual Structures: Information Processing in Minds and Machines*, Reading, MA: Addison-Wesley Publishing Co.
26. Sowa J.F. and Way E.C. (1986). Implementing a Semantic Interpreter using Conceptual Graphs. *IBM Journal of Research and Development*, 30:1, pp.57-69.
27. Swanberg D., Shu C.F. and Jain R. (1992). Architecture of multimedia information system for content-based retrieval. In *Audio Video Workshop*, San Diego, CA.

28. Swanberg D., Shu C.F. and Jain R. (1993). Knowledge guided parsing in video database. In *Electronic Imaging: Science and Technology*, San Jose, California, IST/SPIE.

29. Turtle H.R. (1991). Inference Networks for Document Retrieval. *Ph.D. Thesis*. Department of Computer and Information Science, University of Massachusetts.

30. Weiss R., Duda A. and Gifford D. (1994). Content-based access to algebraic video. In *IEEE International Conference Multimedia Computing and Systems*, Los Alamitos, CA.

31. Wong S.K.M. and Yao Y.Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Transaction on Information Systems*, 13:1 pp.38-68.

32. Zhang H.J., Gong Y., Smoliar S.W. and Tan S.Y., (1994). Automatic parsing of news video. In *Proceedings of IEEE Conference on Multimedia Computing Systems*.