New Generation Computing, 18(2000)323-339
Ohmsha, Ltd. and Springer-Verlag

Information Access in Multimedia Databases Based on Feature Models

Arjen P. DE VRIES[1,2], Menzo WINDHOUWER[1],
Peter M.G. APERS[2] and Martin KERSTEN[1]
[1] CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
[2] University of Twente, 7500 AE Enschede, The Netherlands
Arjen@acm.org, Apers@cs.utwente.nl,
Menzo.Windhouwer@cwi.nl, Martin.Kersten@cwi.nl

Received 04 February 2000
Revised manuscript received 10 April 2000

Abstract With the increasing popularity of the WWW, the main challenge in computer science has become content-based retrieval of multimedia objects. Access to multimedia objects in databases has long been limited to the information provided in manually assigned keywords. Now, with the integration of feature-detection algorithms in database systems software, content-based retrieval can be fully integrated with query processing. We describe our experimentation platform under development, making database technology available to multimedia. Our approach is based on the new notion of feature databases. Its architecture fully integrates traditional query processing and content-based retrieval techniques.

Keywords: Information Access, Multimedia Databases, Query Processing, Content-Based Retrieval.

§1 Introduction

Large-scale multimedia information retrieval is one of the major scientific challenges of this decade. This focus of attention seems natural given the significant advances in technology to capture and store raw material in databases and in files on the world-wide web.[2] As a result, the research field has entered a third stage, improving upon the foundation laid in two previous stages.

First generation multimedia database systems focussed on kernel support for **blobs** (binary large objects), to efficiently store the sizeable objects. Since the early 90s, database vendors have provided support for these non-interpreted byte streams in their core products, leaving timing, synchronization, and quality-of-service to specialized co-processors. These developments paved the way for video-on-demand applications, which are slowly making their way into the homes.

The second phase concerned techniques for annotation and linking media objects. Most of this activity found itself a breeding ground in user interface research and multimedia authoring systems. The database merely contains textual annotations, made accessible efficiently using conventional information retrieval (IR) techniques. Still, multimedia objects remain non-interpreted with respect to retrieval.

The third generation of multimedia database retrieval research focuses on effective techniques for indexing and retrieval by content.[14,27] The ideal searched for are algorithms to automatically index objects according to a semantic framework. Unfortunately, realization of this vision will not become feasible in the foreseeable future; and, it may very well never be achieved for more generic domains, as semantic descriptions are too tightly coupled with the frame or reference (domain) of the intended audience. In the shorter term, the best we can hope for is to make progress in the effective usage of automatically derived (syntactic) features, that aid pre-selection in a large multimedia database. Progress in this area has already demonstrated the feasibility of this approach for retrieving still images, using features based on low-level perceptual image properties such as color distribution and texture. Early experiments seem to indicate that a similar approach can be taken for audio as well[25]; and, combining techniques for these two media with speech recognition and scene segmentation can be used to provide access to the information stored in digital video archives.[23]

The importance of these research challenges is illustrated by the abundance of funding available worldwide. Within The Netherlands alone, the authors are involved in the following large national programs:

AMIS, Advanced Multimedia Indexing and Searching This national project, bringing together researchers from image processing, computer graphics, database technology, and operating systems, focuses on indexing and searching of multimedia databases.[1]

DMW, Digital Media Warehouses This project, which runs under the umbrella of the Telematics Institute, brings together academia and industry to look at the usage of multimedia databases in a cooperative environment.[11]

MIA, Multimedia Information Analysis As part of a national program to stimulate high-speed internet applications, the activities of CWI in the area of feature databases and query articulation are matched with strong research on multimedia document and video analysis. The key scientific innovation is to search for *invariant* properties, which can be used to retrieve multimedia images irrespective of the lightning conditions, viewpoint and scale.[15]

The research activities performed in these projects meet in the joined development of the **Acoi experimentation platform**, a national experimentation platform for research into multimedia databases. Its main goal is to build and maintain an extensible index to multimedia objects on the world wide web. Our initial target is the creation of a collection that consists of one million still images, hundreds of video sequences, and thousands of audio tracks. This large collection will subsequently be indexed using various feature extraction algorithms published in the literature, and developed by partners in the various affiliated research projects.

1.1 Contribution of This Paper

In this paper, we present our ongoing research in the area of multimedia information retrieval, based on a novel architecture of a multimedia database search-engine. The key scientific questions driving our research are:

Information access How to *effectively* satisfy a user's information need using the multimedia objects stored in a database, bridging the gap between the low-level internal representations of multimedia content and the high-level cognitive processes of the user.

Data management How to *efficiently* derive simple and complex multimedia features for widely distributed sources of raw material and making this available as an index for query resolution.

The research contribution of our project lies in the integration of both aspects in a single software architecture, focusing from the start on flexibility, extensibility, as well as efficiency and scalability. This sets it apart from previous work, which focused either on information access (such as the Informedia project[23]) or on efficient processing of similarity queries.[14] We believe an integrated approach is necessary to facilitate the management of very large collections, such as the million still images aimed for in Acoi.

The work presented in this paper is based heavily on[7] and.[16,20,24] It is organized as follows. We start with a motivating example in Section 2. In Section 3, we introduce and discuss the blueprint of a multimedia database architecture, geared at content-based multimedia information retrieval. In the two subsequent Sections, we elaborate on the unique aspects of this architecture: Section 4 discusses its relationship to our information access research line, and Section 5 presents the novel data model and processing scheme especially suited to construct a database with metadata. Finally, Section 6 indicates the challenges ahead and secondary roads to be explored.

§2 Motivating Example

The following informal example illustrates the scope of problems that we address in our approach. Imagine a journalist in the near future, working on a TV news item on El Niño and its effect on weather; assume she is looking for some video fragments and background data to support her story. It is reasonable to assume that she will have access to a distributed, multimedia database at her work, containing news items collected over the years. Also, we expect people to maintain reasonably sized 'multimedia scrapbooks' with pictures and recorded television broadcasts in their homes: the multimedia facilities in today's personal computers are impressive, and the integration of television and multimedia computer equipment into 'intelligent televisions' (as we prototyped in[10]) is *already* gaining popularity as a commercial product.*1 This Section outlines the various ways she may search these personal and professional archives, and relates these to the type of functionality provided in the three generations of multimedia database systems recognized in the Introduction.

In the simplest case, video fragments are only labeled manually with a small number of relevant keywords. In this case, our journalist would search for fragments having the keyword El Niño assigned, which can be supported with first generation multimedia database systems. Although such a boolean search on the presence of certain keywords usually gives a good **precision**, the result most likely evaluates badly on **recall**: most fragments labeled with El Niño will be relevant, but the fragments that had not been labeled with this keyword explicitly cannot be found.

A more advanced database system would support retrieval using subtitles extracted from the recorded video data; for, many broadcasts nowadays are annotated with subtitles (also known as 'captions') for the hearing impaired and/or foreign-speaking audiences. The database system uses IR techniques to query the collection for occurrences of El Niño. This should increase the performance on recall, as all videos that mention El Niño explicitly will be retrieved. By expanding the query automatically with related terms, recall can be further improved, but precision will drop as videos in which only related terms occur are less likely to be relevant.

In a true multimedia search engine, the journalist could further restrict the type of videos she wants by specifying the desired properties of their content as well, hopefully improving the precision of the query results based on the extra information about her particular *information need*. For example, she may want to limit the results to those video fragments that actually contain shots of the ocean, in which the corresponding audio track contains spoken text regarding the abnormally warm waters off the Peruvian coast. The problem at hand is how to express the high-level notion of 'ocean' in the – usually unintuitive – low-level representations of content on which state-of-the-art image retrieval techniques are based. While recognizing the limited semantic value of these representations, we hypothesize that the semantic gap (expressing high-level concepts with low-level features) can be reduced by putting more effort in the dialogue between the user and the search engine.[8]

The holy grail of multimedia retrieval is for the retrieval engine to conclude from a video that it concerns the effects of a warm ocean stream before the coast of Peru on the weather in other parts of the world. Unfortunately, we consider this final example beyond our capabilities, as it requires the attachment of high-level semantics to video fragments and the machine understanding of relationships among various fragments.

*1 See e.g. the recent product marketed by TiVo Inc., which allows you to pause your television for up to thirty minutes, and recommends and records programs of your taste automatically.

§3 A Novel Database Architecture

The essence of existing database management systems (DBMSs) is the process of data abstraction: raising the level of abstraction for data manipulation above the level of interaction with the file system. A DBMS provides the user with a **data model**, a set of concepts that can be used to structure the data. This Section argues that extending database systems to support multimedia search requires more than 'just' the abstraction from data structures: for, the problem underlying multimedia search is that the DBMS has to reason about the *content* of multimedia objects as well. We therefore propose that a multimedia DBMS should explicitly support the modeling of content, a process we identify as **content abstraction**:

Definition 3.1
Content abstraction is the process of describing the content of multimedia objects through metadata, either assigned manually, or extracted (semi-)automatically.

We use the term **detector** to denote software artifacts that implement algorithms for (semi-)automatic content abstraction.

Our motivating example has illustrated various techniques for content abstraction: manually assigned keywords, subtitles, or automatically derived features such as color and texture. The boolean search used in the first example of Section 2 is a search strategy that is perfectly supported in traditional (relational) database management systems: the query is formulated in terms of the data available in the database, and the database system decides how to process the query most efficiently.

The second example requires the integration of database query processing with information retrieval. Both the query and the contents of the database (the subtitles) undergo some processing, such as replacing query words by their stems. The extra functionality required can be supported easily in modern extensible database management systems.

The third example uses content abstractions that are derived from the image data itself. The iconic example of handling this situation in extensible

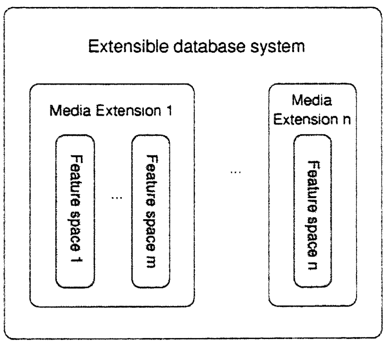


Fig. 1 An Extensible DBMS Architecture with Multimedia Extensions

database systems is the Chabot picture retrieval system[19]; its generalized system architecture is outlined in Fig. 1. In this 'standard' approach, the usage of content abstractions in retrieval is limited to the *explicit* use of user-defined operations on multimedia data types. But, how should end-users know what content abstractions are available for each object, what these abstractions mean at a higher, semantic level, and, more seriously, how to express their queries using a combination of several content abstractions?

The gap between the query formulated by the user and the content abstractions that should be used to satisfy the information need is simply too large. The various content abstractions available provide different views on the data, but it is not clear *a priori* how to combine these views to find relevant objects. In theory, the best candidate to specify this combination is indeed the end user. Unfortunately, features such as color and texture have little semantic value for the average user (and hardly more for expert users). Furthermore, psychological evidence gives reason to doubt whether multimedia information needs can be described fully in textual, propositional form.[8]

In our view, the effective usage of the available content abstractions to satisfy a user's information need is a task for a multimedia DBMS. We have therefore come to the conclusion that multimedia retrieval requires a new design of multimedia database systems, in which the system actively participates in the process of formulating the query. Based on the relevance judgments provided by the user in subsequent iterations, the multimedia retrieval engine assists the user in refining the query, thus *implicitly* reducing the gap between the (low-level) representations of video content such as color and texture and the (high-level) concepts like 'ocean' familiar to the user.

A blueprint of the architecture pursued is illustrated on the left side of Fig. 2; it concerns the definition of interfaces between components, and not a complete instruction for implementation. The architecture separates the design of a multimedia database into three components: a data abstraction component, a content abstraction component, and a retrieval engine. Next to this blueprint, we outline its current implementation in the Acoi platform. Content abstraction

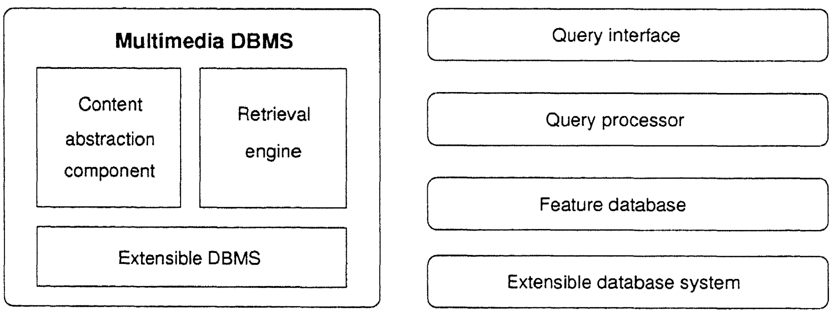


Fig. 2 The Mirror Architecture, a blueprint of our proposed multimedia DBMS architecture, next to an overview of the four layers in the prototype implementation of the Acoi platform.

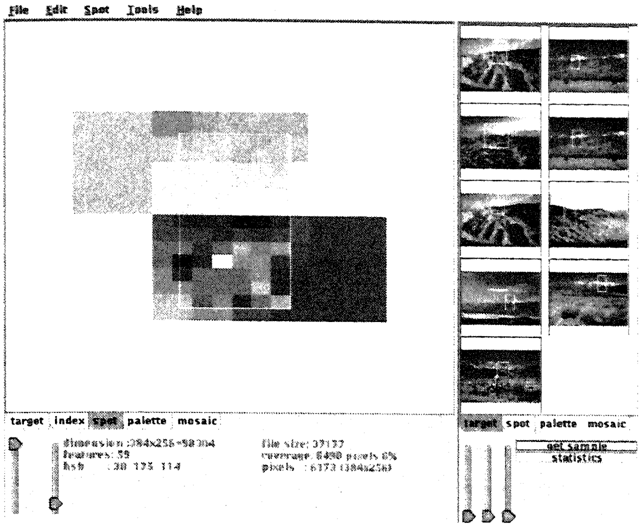


Fig. 3 A User Interface Supporting Spot-based Retrieval

and data abstraction are managed by the notion of a **feature database** (discussed shortly in more detail). The interaction with the user is managed by the query processor. The remainder of this Section introduces the four layers of the Acoi platform; next, Sections 4 and 5 elaborate on the two layers addressing our key research questions.

3.1 Query Interface

The query interface for multimedia databases differs considerably from the traditional straight line approach encountered in OQL or SQL. Query formulation involves a mix of textual descriptions, component clipping, and expression of temporal, spatial, as well as topological relationships between objects.

We have taken a pragmatic approach to assume that such interfaces are largely built on Java with an identifiable clean interface to a database. Except for occasional demonstrators, we do not invest in the design of user-interfaces per se; but, we do experiment with various interaction models to identify the primitive actions that need be supported in the interface between the GUI of user applications and the multimedia database system.

An example is our research into the idea of **spot-based retrieval**, see Fig. 3. A spot is a distinctive small region of a sample image used to steer the search process. The spot is used to localize similar images in the collection, using a few controllable parameters, such as their position in the target image, their scale, and their possible color distortions. For example, searching tigers in a database is translated into identifying a spot on the skin of a sample tiger image. This defines a small color set and their spatial relations. The user may relax the spot features by controlling the color range, brightness, textures, and their spatial locality in a target image, e.g. tigers do not fly therefore a spot is not likely of interest if it appears at the top. Subsequently, we inspect all color-space indices for similar spots. A subset of qualifying index clips are returned as an approximate answer. The user can then use this information to refine his query and steer towards the desired result.

3.2 The Retrieval Engine

The retrieval engine controls the dialogue with the user. Its implementation is strongly rooted in the theory and techniques developed in information retrieval. Although IR theory is usually only applied to text documents, it can be adapted for the case of multimedia.[7] Indeed, in their survey of probabilistic models in IR, Crestani et al. remark that a document can be 'any object carrying information.'[6]

Users first express their information needs in the form of an initial query, that consists of a combination of an explicit statement of properties of the objects desired, like keywords that should occur, with an implicit statement of such properties, using a (possibly empty) set of example objects with relevance judgments. The retrieval engine then determines which objects are likely to be relevant for the user's information need. Since the query cannot possibly capture all aspects of the information need, this process will be iterated.

This layer of the architecture supports the primitives to specify example objects, and relevance judgments about previously retrieved objects. Also, it supports directives to declare the start and end of a query session related to some particular information need. Using the information elicited from the interaction with the user, the retrieval engine estimates which content abstractions are most likely to explain the relevance judgments, and formulates a query accordingly. How the information retrieval theory is used to retrieve relevant objects is explained in Section 4.

3.3 Feature Databases

In principle, the retrieval engine may be implemented on top of any extensible DBMS, e.g. as an extra component in between the Chabot system and the user application. But, the process of content abstraction in multimedia databases imposes some constraints that are hard to handle effectively in existing database technology: most notably, (1) the ever-increasing set of data types and operations on these types, (2) a large number of is-a and has-a relationships between objects, content abstractions, and their components, and (3) the necessity to index the multimedia objects incrementally.

Multimedia data comes in many different formats for the raw data (such as JPG, GIF, etc.), and this number of data types is likely to keep on growing, e.g. as better compression algorithms are invented or new authoring tools enter the market. Most detectors can however only be applied to a small number of data types and formats. Similarly, because researchers continue to discover new types of content abstractions, we also face an increasing set of feature types and their costly detection. Naturally, we would prefer the capability to seamlessly integrate third-party feature extraction modules in the database architecture.

Furthermore, it is insufficient to derive a single feature value for a complete object (like its color distribution), but we should detect and store discriminative features for object components as well, including retention of temporal and spatial relationships. Finally, given the size, distribution and volatility of the data collections to be indexed, construction of a multimedia feature index is an on-going activity. Regardless, we should answer requests immediately; we have to circumvent the problem that the feature index is incomplete at any moment of time.

This complex environment motivates the introduction of an extra layer in between the retrieval engine and an extensible DBMS. We believe that the process of content abstraction is best understood in the parsing paradigm. Therefore, we propose to support the retrieval engine with a **feature database***2: a database system with a novel data model based on *feature grammars*. Feature grammars extend the notion of context-free grammars with active nodes to represent detectors. A data model based on this concept differs from traditional data models – object-oriented and relational – in its rich provision for partial- and multi-view descriptions of the underlying objects.

Feature grammars are inspired by SGML, and its derivative XML, to provide for a concise representation of large schemas. In particular, the grammatical notation provides a good handle to structure alternatives (union-types), which in a more traditional OO approach would lead to an explosion in class definitions. A data model based on feature grammars is not restrictive, and providing a mapping to a relational data model for storage or to XML-based schemes for output is straightforward. At a technical level, the grammatical flavor brings many of the techniques known in language parsing into the realm of multimedia indexing.[20)] In particular, it provides a simple scheme to support evolutionary parsing of a multimedia document as more feature detectors are added to the feature grammar. But, whereas conventional parsers just *validate* an input sentence against the language defined by the grammar, we exploit parsing techniques to *expand* the input sentence to a set of alternative valid sentences: in other words, instead of producing a single parse tree, we generate the union of all alternative parse trees. A concrete example of the use of this novel data model is discussed in Section 5.

3.4 Database Support

Realization of the feature database calls upon the facilities offered by modern extensible databases. These provide both the facilities to deal with multimedia data items and the mechanisms to implement the necessary search accelerators for boosting the metric search in feature spaces. The underlying database system deployed in our projects is Monet, a novel and powerful extensible DBMS. It supports a binary relational data model, using vertical decomposition to represent complex data; which is (see[5)]) a particularly useful basis for the storage of the inhomogeneous data captured in the feature grammars. Monet

*2 Notice that the term 'feature' in this context is *not* restricted to multi-dimensional feature vectors, but may denote any type of content abstraction that reveals some aspect of the meaning of the multimedia object.

also supports modular extension, a technique in line with **data cartridges** and **data blades**, which encapsulate the routines and data structures for a particular data type. The system provides already modules that support image applications and GIS, and modules for the management and analysis of video data are currently under development. Results on their functionality and performance have been reported elsewhere.[3,4,18]

§4 Information access

Our approach to the implementation of the retrieval engine is based on the architecture of information retrieval systems. The underlying philosophy of the retrieval engine has been inspired by theories from cognitive psychology, in particular Paivio's dual coding theory.[13] Aspects of its design are similar in spirit to both the Viper[21] and the FourEyes[17] image retrieval systems.

Following Reference[26], the three fundamental issues in probabilistic information retrieval are: representation of documents, query formulation, and a ranking function. These three issues are reflected in the three layers of the retrieval engine (shown in Fig. 4): the concept layer manages the basic 'concepts' derived from multimedia objects, the evidential reasoning layer implements the ranking function, and the relevance feedback layer steers query formulation through a dialogue between the user and the system.

Words in natural language correspond more or less to concepts from the real world. Conversely, the computation of a feature value is a – usually unique – point in an abstract, multi-dimensional feature space. We therefore cluster the feature values based on their proximity in feature space; very likely the significance of this proximity reveals an underlying 'concept' that can be useful at query time. We further use the identified clusters as if these are words in text retrieval: the clusters become the basic blocks of 'meaning' for multimedia information retrieval.

Of course, these semantically rather poor 'concepts' are not suited well for direct interaction with the users of the digital library. As a solution, we automatically construct a thesaurus, which associates words in textual content

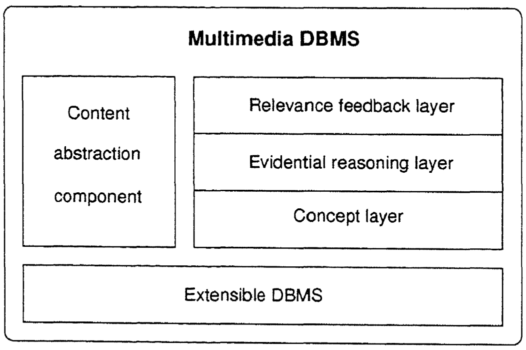


Fig. 4 A Three-layered Design of the Query Engine in Our Proposed Architecture

abstractions to these concepts, clusters in the image content. The thesaurus is used for automatic query expansion. An interesting aspect of this approach, is that this thesaurus can be considered an implementation of Paivio's dual coding theory, see.[22]

Querying the digital image library now takes place as follows. First, the user enters an initial (usually textual) query. Next, we use the thesaurus to select clusters from the image content representations that are relevant to this initial query. The multimedia objects are ranked in the evidential reasoning layer using a standard probabilistic retrieval model, and the best results of this query are presented to the user. The user may provide relevance feedback for these images; this relevance feedback is then used to improve the current query based on Rocchio query term reweighting and expansion. An image retrieval application using this approach has been demoed at VLDB '99.[9]

A problem for the current retrieval system is that the thesaurus sometimes associates words in the annotations to irrelevant clusters, or the clustering process chooses clusters having little semantic value. To alleviate these problems, we will use machine learning techniques to adapt the thesaurus and the clustering, by combining the relevance feedback across several query sessions.

§5 Using Feature Grammars

Feature grammars describe the relationships between the various feature values and the detectors which are used to calculate them, not unlike the relations between metadata and daemons used to construct the AltaVista index.[12] Their introduction is motivated, besides the reasons already mentioned in Section 3.3, by the observation that indexing an arbitrary multimedia object leads to a hierarchical structure that describes the components of interest for the retrieval engine. We have shown in Reference[20] that such hierarchical structures can be described concisely with formal grammars. The architecture of the current implementation of a feature database on top of Monet is shown in Fig. 5; XML is used as the exchange format between the various components of the system. In this Section, we present the running example (partially shown in Fig. 6) of indexing a set of images collected from the WWW, using a specification of this process in a feature grammar.

Before going into this concrete example, we will take a short look at the formal definition of a feature grammar. A feature grammar G is a quintuple $G = (V, D, T, S, P)$, where V are the variables, D a set of special variables called detectors (which will be described in the following paragraphs), T the terminal symbols, $S \in (V \cup D)$ is the start symbol and $P \subseteq (V \cup D) \rightarrow (V \cup D \cup T)*$ are the production rules.

The most prominent extension of feature grammars to context-free grammars are grammar rules whose left-hand side symbols are programs, called detectors. These programs interrupt the parsing process, and read from the same input as the parser. They also write their output back onto the parser input, which then again consumes it and evaluates it against the right-hand side of the detector rule. The Acoi platform allows detectors to be **blackbox detectors** writ-

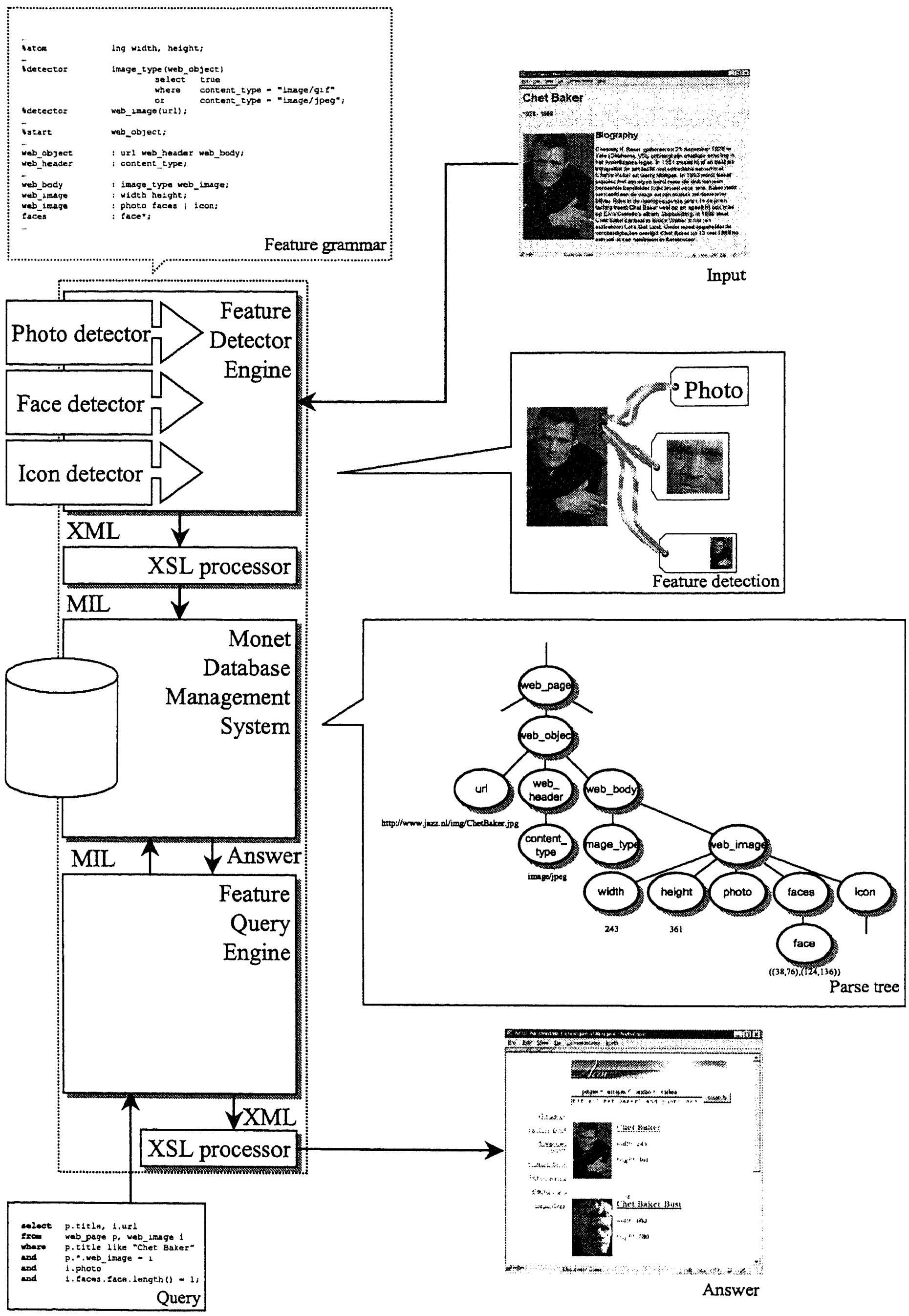


Fig. 5 Acoi Architecture

```
...
%atom          lng width, height;
...
%detector      image_type(web_object)
                 select true
                 where content_type = "image/gif"
                 or content_type = "image/jpeg";
%detector      web_image(url);
...
%start         web_object;

web_object  →  url web_header web_body?;
web_header  →  content_type;
...
web_body    →  image_type web_image;
web_image   →  width height;
...
```

Fig. 6 (Partial) Example of a Feature Grammar

ten in any programming language, or (preferably) be specified in a declarative specification language similar to SQL. The latter are called **whitebox detectors**, and can be used by the feature database system to determine the optimal schedule for processing the set of multimedia objects.

The parser input is initialized with the *url* of the image to be indexed (*e.g.* `http://www.jazz.nl/img/ChetBaker.jpg`). The feature grammar defines the goal of the parsing process as the acceptance or rejection of the given *url* as a legitimate *web_object*. The proof for acceptance is attempted by proving the right-hand side of the *web_object* rule. The first required symbol is an atom, the *url*. For an atom the parser looks for a matching symbol in the front of the input; so it will extract the matching initial value. The next symbol, the *web_header*, is a black box detector (so the program associated with this detector is called). The *web_header* takes the *url* from the parse tree, sends a HTTP `HEAD` request to the specific server, filters out the *content_type* of this *url*, and puts it – correctly labeled – in front of the input. When the program control returns to the parser it will try to prove the right-hand side of the *web_header* rule, thus consuming the *content_type* token. The parsing process will continue in this fashion until the *image_type* symbol is reached. This symbol is a whitebox detector and associated with it is a query over the constructed parse tree. On the basis of the value of the *content_type* leaf this detector steers the parser into the *web_image* part of the grammar. The parser continues in this fashion until the right-hand side of the start rule has been proved or rejected. The resulting parse tree is stored in the underlying DBMS, and can be used for retrieval by the retrieval engine.

§6 Conclusions and Future Work

One of the main challenges is to bridge the gap between the concepts in the real-world environment of the end users and the low-level features that can be computed from the raw data of multimedia objects. We presented the use of IR theory for this purpose, and proposed a novel architecture for multimedia database management systems that integrates these techniques in all levels of the system.

We also demonstrated how the new notion of **feature database** can manage the heterogeneity in the multimedia objects and their associated metadata, using feature grammars in the data model. This allows us to capture the inter-component structure and provide the semantic basis for evidential reasoning in the retrieval engine. Its direct mapping to either a relational or object-relational scheme makes it an attractive intermediate model in the Acoi platform.

Currently, a large scale experimentation platform is under construction to demonstrate our approach against a database consisting of over a million images gathered from the web. Future work in the information access research line consists of the application of machine learning techniques to improve the representations of multimedia objects in the concept layer using feedback across sessions. Also, we expect significant improvements in the quality of multimedia information retrieval, by obtaining finer grained information from the dialogue with the user. In the data management research line, we will apply techniques from parallel and distributed database technology to increase the efficiency of the overall system architecture.

Acknowledgements

We would like to thank our researchers working in our multimedia projects; they are the ones that have to realize our ambitious goals. A special word of thanks goes to Mark van Doorn, Niels Nes, and Albrecht Schmidt for their research contributions to this paper.

References

1) *AMIS, Advanced Multimedia Indexing and Searching*, URL http://www.cwi.nl/~acoi/Amis/, 1999.

2) Apers, P.M.G., Blanken, H.M. and Houtsma, M.A.W., eds. *Multimedia Databases in Perspective*, Springer Verlag, 1997.

3) Boncz, P.A., Quak, C.W. and Kersten, M.L., "Monet and its Geographic Extensions," in *Proc. of the 1996 EDBT conference*, 1996.

4) Boncz, P.A., Wilschut, A.N. and Kersten, M.L., "Flattening an Object Algebra to Provide Performance," in *Fourteenth International Conference on Data Engineering*, pp. 568–577, Orlando, Florida, February 1998.

5) Copeland, G.P. and Koshafian, S.N., "A Decomposition Storage Model," in *Proc. of the SIGMOD Conference*, pp. 268–279, 1985.

6) Crestani, F., Lalmas, M., van Rijsbergen, C.J. and Campbell, I., 'Is This Document Relevant? ... Probably': a Survey of Probabilistic Models in Information Retrieval," *ACM Computing Surveys, 30, 4, pp. 528–552*, December 1998.

7) de Vries, A.P., *"Content and Multimedia Database Management Systems," PhD thesis,* University of Twente, Enschede, The Netherlands, December 1999.

8) de Vries, A.P., van der Veer, G.C. and Blanken, H.M., "Let's Talk About it: Dialogues with Multimedia Databases," *Database Support for Human Activity, Displays, 18, 4, pp. 215–220,* 1998.

9) de Vries, A.P., van Doorn, M.G.L.M., Blanken, H.M. and Apers, P.M.G., "The Mirror MMDBMS Architecture," in *Proc. of 25th International Conference on Very Large Databases (VLDB '99)*, pp. 758–761, Technical demo, Edinburgh, Scotland, UK, September 1999.

10) de Vries, A.P., "Intelligent television: A Testbed for Multimedia Information Filtering," *Technical Report CTIT 97–35*, Centre for Telematics and Information Technology, 1997.

11) *DMW, Digital Media Warehouses*, URL http://www.cwi.nl/~acoi/DMW/, 1999.

12) Eberman, B., Fidler, B., Lannucci, R.A., Joerg, C., Kontothanassis, L., Kovalcin, D.E., Moreno, P., Swain, M.J. and van Thong, J.M., "Indexing Multimedia for the Internet," *Technical Report CRL 99/2*, Cambridge Research Laboratory, 1999.

13) Eysenck, M.W. and Keane, M.T., "Mental Representation," *Cognitive Psychology: A Student's Handbook*, Lawrence Erlbaum Associates, 3rd ed., 1995.

14) Faloutsos, C., *Searching Multimedia Databases by Content*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1996.

15) Gevers, T. and Smeulders, A.W.M., "A Comparative Study of Several Color Models for Color Image Invariant Retrieval," in *Proc. of the First International Workshop on Image Databases and Multi-media Search (IDB-MMS '96)*, pp. 17–26, Amsterdam, The Netherlands, August 1996.

16) Kersten, M.L., Nes, N. and Windhouwer, M.A., "A Feature Database for Multimedia Objects," in *ERCIM Database Research Group Workshop on Metadata for Web Databases*, pp. 49–57, Bonn, Germany, May 1998.

17) Minka, T.P. and Picard, R.W., "Interactive learning using a 'society of models'," *Pattern Recognition, 30, 4*, pp. 565–581, 1997.

18) Nes, N. and Kersten, M., "The Acoi Algebra: A Query Algebra for Image Retrieval Systems," in *Advances in Databases. 16th British National Conference on Databases, BNCOD 16*, pp. 77–88, Cardiff, Wales, UK, July 1998.

19) Ogle, V.E. and Stonebraker, M., "Chabot: Retrieval from a Relational Database of Images," *IEEE Computer, 28, 9, pp. 40–48*, September 1995.

20) Schmidt, A.R. Windhouwer, M.A. and Kersten, M.L., "Feature Grammars," in *Proc. of the International Conference on Systems Analysis and Synthesis*, Orlando, FL, USA, August 1999.

21) Squire, D.McG., Müller, W., Müller, H. and Raki, J., "Content-based Query of Image Databases, Inspirations from Text Retrieval: Inverted Files, Frequency-based Weights and Relevance Feedback," in *The 11th Scandinavian Conference on Image Analysis*, pp. 7–11, Kangerlussuaq, Greenland, June 1999.

22) van Doorn, M.G.L.M. and de Vries, A.P., "The Psychology of Multimedia Databases," to appear in *ACM Digital Libraries (DL'00)*, San Antonio, Texas, USA, June 2000.

23) Wactlar, H., Kanade, T., Smith, M. and Stevens, S., "Intelligent Access to Digital Video: The Informedia Project," *IEEE Computer, 29, 5*, May 1996.

24) Windhouwer, M.A., Schmidt, A.R. and Kersten, M.L., "Acoi: A System for Indexing Multimedia Objects," in *International Workshop on Information Integration and Web-based Applications & Services*, Yogyakarta, Indonesia, November 1999.

25) Wold, E., Blum, Th., Keisler, D. and Wheaton, J., "Content-based Classification, Search, and Retrieval of Audio," *IEEE Multimedia, 3, 3*, 1996.

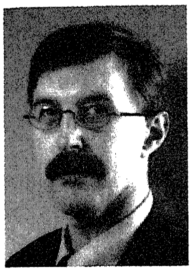
26) Wong, S.K.M. and Yao, Y.Y., "On Modeling Information Retrieval with Probabilistic Inference," *ACM Transactions on Information Systems*, 13, 1, pp. 38–68, January 1995.

27) Wu, J.K., Narasimhalu, A.D., Mehtre, B.M., Lam, C.P. and Gao, Y.J., "CORE: a Content-based Retrieval Engine for Multimedia Information Systems," *Multimedia Systems, 3*, pp. 25–41, 1995.

Arjen P. de Vries, Ph.D.: He received his Ph.D. in Computer Science from the University of Twente in 1999, on the integration of content management in database systems. He is especially interested in the new requirements on the design of database systems to support content-based retrieval in multimedia digital libraries. He has continued to work on multimedia database systems as a postdoc at the CWI in Amsterdam as well as University of Twente.

Menzo Windhouwer: He received his MSc in Computer Science and Management from the University of Amsterdam in 1997. Currently he is working in the CWI Database Research Group on his Ph.D., which is concerned with multimedia indexing and retrieval using feature grammars.

Peter M.G. Apers, Ph.D.: He is a full professor in the area of databases at the University of Twente, the Netherlands. He obtained his MSc and Ph.D. at the Free University, Amsterdam, and has been a visiting researcher at the University of California, Santa Cruz and Stanford University. His research interests are query optimization in parallel and distributed database systems to support new application domains, such as multimedia applications and WWW. He has served on the program committees of major database conferences: VLDB, SIGMOD, ICDE, EDBT. In 1996 he was the chairman of the EDBT PC. In 2001 he will, for the second time, be the chairman of the European PC of the VLDB. Currently he is coordinating Editor-in-Chief of the VLDB Journal, editor of Data & Knowledge Engineering, and editor of Distributed and Parallel Databases.



Martin Kersten, Ph.D.: He received his PhD in Computer Science from the Vrije Universiteit in 1985 on research in database security, whereafter he moved to CWI to establish the Database Research Group. Since 1994 he is professor at the University of Amsterdam. Currently he is heading a department involving 60 researchers in areas covering DBMS architectures, datamining, multimedia information systems, and quantum computing. In 1995 he co-founded Data Distilleries, specialized in data mining technology, and became a non-executive board member of the software company Consultdata Nederland. He has published ca. 130 scientific papers and is member of the editorial board of VLDB journal and Parallel and Distributed Systems . He acts as a reviewer for ESPRIT projects and is a trustee of the VLDB Endowment board.