Content Independence in Multimedia Databases

Arjen P. de Vries
CWI, Room M346, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
E-mail: arjen@acm.org

A database management system is a general-purpose software system that facilitates the processes of defining, constructing, and manipulating databases for various applications. This article investigates the role of data management in multimedia digital libraries, and its implications for the design of database management systems. The notions of content abstraction and content independence are introduced, which clearly expose the unique challenges (for database architecture) of applications involving multimedia search. A blueprint of a new class of database technology is proposed, which supports the basic functionality for the management of both content and structure of multimedia objects.

Introduction

Building large digital libraries challenges most fields in computer science. In their overview of "strategic directions" for digital libraries, Adam and Yesha (1996) identify numerous issues that require further research, touching on fundamental research questions as well as more practical software engineering problems. A huge volume of papers is relevant to at least some aspect of building digital libraries, and these papers are spread over many different fields such as operating systems, databases, information retrieval, artificial intelligence (both computational vision and reasoning under uncertainty), pattern recognition, and cognitive science.

Most digital library research takes place in a single field or discipline; looking back on the scientific literature that has appeared in the last decade, the researchers in different fields seem to have reached some local optima, while there is a clear need for integration of the different types of technology developed in these fields. There are certainly some obstinate problems with the current state of the art:

- The gap between the functionality required by real user tasks and the functionality provided in research prototypes is quite large;
- Developing advanced multimedia retrieval applications on top of existing systems is a complicated process;
- The current approach to integration of different components cannot be expected to scale up to data collections of realistic sizes.

The underlying hypothesis of research into the Mirror architecture (proposed in de Vries [1999]) is that the proper software architecture for digital libraries can facilitate the work of researchers seeking beyond the traditional boundaries of their disciplines. The goal of this research is to separate the user applications defined for a certain collection from the particular algorithms to process the data and to learn from the users. Mirror attempts to create order in the chaos and confusion about what a "multimedia database" is, and then uses this understanding to define a blueprint of such a system and to provide guidelines for the implementation of such systems.

This article analyzes the role of *data management* in the Mirror architecture; its main purpose is to explain the idea of *content (abstraction) independence* through the idea of *content abstraction*, and the role this may play in the study of information access in multimedia digital libraries. The next section identifies the key ideas of database technology that are emphasized in Mirror: *data abstraction* and *data independence*. Sections follow on what is special about multimedia, content abstraction, and multimedia data and database relate these issues to multimedia data management and illustrate the support for multimedia in extensible database systems. They, in turn, are followed by the introduction of the notion of *content independence* and explain how this idea can be supported in a database system. Finally, the last section concludes this discussion of multimedia databases with an outlook on future work.

Data Makes the World Go Round

We start with a (selective) overview of the fundamental notions in database technology, the primary inspiration for the Mirror architecture. A **database management system (DBMS)** is a *general-purpose* software system that facilitates the processes of defining, constructing, and manipulating databases for various applications (see, e.g., Elmasri & Navathe [1994]).

 • Published online 10 August 2001

The most important task of a DBMS is to reduce the amount of work required to adapt software systems again and again in a changing environment. It provides this ability to evolve by emphasizing **data independence:** programs that access data maintained in the DBMS are written independently of any specific files, ensuring that applications can continue to run (albeit perhaps at reduced performance) if the stored data is reorganized to accord other applications of higher performance.[1]

Data independence is achieved through **data abstraction:** raising the level of abstraction for data manipulation above the level of interaction with the file system. A DBMS provides users (which can be application programs) with a conceptual representation of the data, referred to as the database schema. This database schema is specified in its **data model,** a set of concepts that can be used to describe the structure of a database.

Data independence has always been the driving force behind the development of database technology. However, apart from data independence, a methodology to obtain efficient query processing is another big advantage of the database approach. Efficiency can be achieved using a divide-and-conquer strategy, in which the original information request is transformed into the final query plan using a number of intermediate representations. Each step uncovers another piece of abstraction, and gets closer to the machine level. These transformations enable the application of optimizations based on set-oriented processing to avoid inefficient nested-loop processing and simplify the support of scalability through parallelism and distribution.

What Is So Special About Multimedia?

In recent years, various software systems that handle multimedia data have been presented in the literature as multimedia database systems. The author argues that a large proportion of these systems should not be classified as multimedia database systems. On the one hand, many multimedia software systems support retrieval in collections of multimedia data, but these do not provide data independence or support for efficiency and scalability. On the other hand, extensible database systems with multimedia extensions support querying insufficiently, and do not even approximate the retrieval functionality provided in the first type of multimedia software systems.

Characteristics of Multimedia Data

Computer scientists use the term "multimedia" to refer to anything that is not conventional alpha-numeric data. Sometimes, the term is made more explicit by an enumeration of data types, appealing to an intuitive notion of multimedia: image, audio, video, and text. Defining multimedia more precisely than by a list of media types is surprisingly difficult. Grosky, Fotouhi, and Jiang attempted to define multimedia data by the human activity involved in creation of the data (Grosky et al., 1998). But long before multimedia data is eventually inserted in the database, movies and music have been created by humans as well, usually with specific care to communicate the artist's message. A claim that the semantics of multimedia data are implicit in the data is debatable as well, because the semantics of an employee number in its numeric representation are no less implicit than the semantics of a stop sign in its iconic representation.

Mirror distinguishes **multimedia data** (including text) from "traditional data" (usually alpha-numeric) from a query perspective:[2]

> **Definition 1.** *The information conveyed by multimedia data may represent anything from the real world, while the information conveyed by traditional data is a symbolic representation of facts restricted to the database's (limited) universe of discourse.*

It follows directly from this definition that queries in a multimedia database can be unrelated to the context of the applications for which the data has been collected: not the form in which the data is represented, but the scope of the semantics captured in the data determines whether it is considered multimedia data or not.

Although this definition has its own weaknesses, it is particularly useful for the study of multimedia databases: it identifies applications that use specific characteristics of multimedia data, and therefore may impose new requirements on database systems. Using icons to represent male or female in some employee database does not require extra functionality from the DBMS, except possibly an abstract data type (ADT) to handle bitmaps; the only "new" aspect is that the application developer encoded the male/female distinction with a different set of symbols than the more common choice for a boolean or enumeration type. In a multimedia digital library, an application developer cannot elicit all aspects of the data that may be relevant for its users, and therefore the information encoded in the data cannot be represented using a limited set of encodings that is (or could have been) known beforehand.

Multimedia Data With a Limited Context

Not all applications handling multimedia data require functionality that goes beyond standard DBMS services. Sometimes, it makes sense to limit the context of queries that can be processed to a restricted universe of discourse. As an example, consider a video directory service special-

[1] Tsichritzis and Klug describe data independence in database systems as follows: "Data independence is not the capability to avoid change; it is the capability to reduce the trauma of change" (Tsichritzis & Klug, 1978).

[2] Admittedly, the term multimedia is somewhat misleading, as it usually denotes *single-media* objects. The term multimedia has been chosen over alternatives such as "media item" to conform to the vocabulary that has become common in database literature.

ized in soccer, like the one described in Velthausz (1998). Its architecture consists of two subsystems: a video analysis and a video retrieval component. The video analysis subsystem uses a conceptualization of a limited set of *predefined* objects and events, including "ball," "player," "goal," and "corner," that are extracted from the data automatically, whereas the retrieval component makes the extracted concepts available to the users. As such, the possible queries are limited to the domain knowledge encoded in the analysis component. Because only alpha-numeric data is used for retrieval, this application is served perfectly well by a DBMS with a video ADT.

Storing multimedia data, as well as querying "normal" data that has been derived from multimedia data, does not turn a database system into a multimedia database system. A *real* multimedia DBMS should not restrict its users to predefined access patterns. In the case of the soccer service aforementioned, it could assist developers of the analysis component with the specification of queries that correspond to a predefined conceptualization. Yet, these predefined conceptualizations only capture the information required to answer a subset of all possible queries about the soccer videos; end users should be able to formulate queries for unanticipated information needs as well (e.g., marketeers of some multinational firm may be interested to see how often their advertisement in the stadium has been shown on television in the last year). Of course, it may be impossible to construct proper queries for complex information needs, given the digitized data and the currently known automatic analysis techniques. However, the decision whether it is worthwhile to spend more time to try and find relevant objects for an ad hoc query should be left to the user.

Composite Objects

For the sake of clarity, we consider only atomic multimedia objects in this article. An "atomic multimedia object" is defined as the smallest amount of data that can be retrieved as one object from a collection without losing its meaning, e.g., a piece of text, an image, a video segment, or a sound fragment. It cannot be divided into smaller objects without losing its meaning. The granularity of this atomicity is determined by the object's semantic content, and is therefore a subjective opinion of the creator of the object. This definition of a multimedia object corresponds to the notion of "media item" in Hardman's multimedia document model (Hardman, 1998), while ignoring its other structuring primitives. We conjecture that the structure of composite multimedia objects over its media items can and should be modeled using the logical type constructors of an appropriate (nested) data model.

Content Abstraction

Summarizing the previous section, a multimedia DBMS should assist the user to access multimedia objects based on their semantics. Unfortunately, these semantics are implicit to the raw media data. Grosky distinguishes between the digitized multimedia data and **content-based metadata,** user-recognizable surrogates for these objects which comprise their content (Grosky et al., 1998). The latter can be used to infer information regarding the former's content.

By analyzing and processing the raw data, semantics can be made explicit to some extent on different abstraction levels, varying from feature values (e.g., color histograms) to knowledge-based concepts (Boll, Klas, & Sheth 1998); other metadata can only be added by human annotators. From the perspective of managing the multimedia data, it does not matter which particular method is used to obtain metadata. Thus, let **content abstraction** denote the process of adding content-based metadata to the raw data.[3]

Definition 2. *Content abstraction is the process of describing the content of multimedia objects through metadata, either assigned manually, or extracted (semi-) automatically.*

There exists a gap between the content as perceived by our senses and the content that has been interpreted by our brain. For video, Hampapur and Jain call these the audiovisual content and the semantic content, respectively (Hampapur & Jain 1998). When we watch a video, our mind manages to derive the semantic content from the audiovisual content. This process requires a significant amount of background knowledge; it cannot be performed automatically independent of the domain. A similar distinction between two types of content can be made for other media types: images have pure visual content, and sounds can be experienced without knowing what makes that sound, or ever having heard it before. Borrowing terms from linguistics, we refer to the content of a multimedia object at the perceptual level as its **syntactic content,** and to its interpretation in concepts from the real world as its **semantic content.**

These different types of content cause a problem for manual annotation. Substantial neuropsychological evidence exists that some properties of audiovisual data cannot unambiguously be expressed verbally. In his book, Iaccino reviews psychological research in differences between the two hemispheres of the brain with respect to perception (Iaccino, 1993). While the left hemisphere is verbal and analytic, the right hemisphere is nonverbal and holistic. Each hemisphere is specialized for a different kind of thinking or *cognitive style.* Although a verbal-nonverbal dichotomy associated with the two sides of the brain is still considered speculative, the vast amount of research with split-brain patients and people with cerebral lesions reported shows convincingly that different areas in the brain are responsible for different perceptual processing. The areas of the brain that handle language are not always involved in this processing. Some of the perceptual information is not

[3] The result of this process is referred to as a content representation.

mediated verbally, and therefore difficult to express in words.

Accepting the idea of different cognitive styles leads to the observation that the usage of textual descriptions alone to search the database may always be too restrictive: the user's valuation processes are different from the query evaluation process that has been modeled in the system. Multimedia retrieval has to take into account more than just textual metadata; the metadata has to capture the syntactic content as well.

For this reason, content-based retrieval systems such as QBIC (Niblack et al., 1993) use a different type of content abstraction: automatically derived properties from the content of the multimedia objects called "features." Features mainly represent syntactic content of the stored objects. An example feature, often used in image retrieval systems, is a measure expressing the color distribution of the image, such as the color histogram. Other possible features are based on the texture and composition of the image. In the Musclefish system (Wold, Blum, Keisler, & Wheaton, 1996), features based on pitch, energy, and more advanced audio properties support the content-based retrieval of sounds.

Shardanand and Maes (1995) have approached the problem of content abstraction from a slightly different angle. The underlying idea is that the group of all users of a digital library can be divided into subgroups having similar interests; the group labels can then be used to describe the objects themselves using one level of indirection. The technique is commonly found in online stores selling books or compact discs: upon login, the system asks you to judge a selection of compact discs by several artists. This profile of your taste is then used to find people that like the same discs. If most "similar" people also judged another record highly, the system then recommends it to you. This idea (known as "social information filtering") overcomes the problems with identifying suitable features for objects like music and art, and deals implicitly with qualitative aspects like style, which would hardly be possible with automatically derived features.

Multimedia Data and Databases

A multimedia object (e.g., a video or a photograph) can be manipulated in a DBMS by addressing the following three aspects in the schema:

- The digitized object representation;
- Data abstraction;
- Content abstraction.

A multimedia object (existing in the real world) is represented in the computer by its digitized object representation: the "raw data." We ignore the practical problem that one real-world object may have several digitized representations, produced by varying types of hardware, or stored in different data formats of varying quality. Data abstraction refers to traditional data modeling, representing attributes such as the title of a movie, or the photographer of a picture. Unfortunately, it is difficult to define precisely what distinguishes data abstraction from content abstraction. Borrowing some terminology from Boll et al., (1998), data abstraction relates to content-independent metadata whereas content abstraction refers to the content-dependent metadata. A movie's title is, however, not content independent, though handled perfectly by "traditional data modelling." A discussion of the true reason for distinguishing between data and content abstraction is deferred to the next section, in which the notion of content independence is explained. At this point, it suffices to view both aspects as very similar, addressing the process of describing the multimedia object with its metadata in such a way that it can be stored and manipulated in a DBMS.

The simplest approach to the management of multimedia data stores a multimedia object as a tuple of its digitized object representation and its metadata. The media type of the digitized object representation is supported as an atomic data type. The data is viewed and manipulated outside the database, or alternatively, basic operations on these data types have been added as well. In our prototype DBMS for example, we could model a video as a tuple of two elements, the video object itself and a string representing the title, so we may easily play the video titled "Romeo and Juliet." Using the more powerful aspects of our nested data model, we may also represent directly the structure of a composite video object, e.g., as a 3-tuple with an audio element, a text element for the captions or subtitles, and a collection (or, maybe more appropriate, a list) of key frames summarizing the visual content.

Now consider extending such a schema with content representations for these key frames. Manual annotation may result in a set of keywords, or some natural language description of the scene. Feature extraction can be performed by calling the appropriate operations defined for the image datatype. The set of images in the original schema is then replaced with a set of tuples, containing the image and its metadata as elements. Approximate retrieval is performed by computing distances to a query vector, and sorting the images in order of their distance to the query object. Thus, key frames can now be queried by boolean retrieval using the manual annotations, or by approximate retrieval on color, texture, or shape.

Given these examples, it seems like an extensible database system with a nested data model provides sufficient support for querying on textual data or by content. The nesting could be removed in a wrapper (by flattening the schema), such that any object-relational DBMS would do. Some problems have been ignored, such as the inhomogeneity in collections of composite multimedia objects, requiring support for deep inheritance hierarchies and polymorphic operators, neither of which is supported well in current DBMSs. Also, integration of approximate retrieval techniques with a (relational) DBMS motivates the development of new algorithms for efficient query processing: the DBMS should support multidimensional index structures, and it should take advantage of the fact that often it is not required to rank all objects. Still, these are developments that fit perfectly well within the traditional database architecture, and other application domains (like GIS) would benefit equally well.

Content Independence

The real problem with the previous examples is that multimedia search requires a more complex query process than the exact one-shot queries common in administrative applications. Any search process takes place in four phases that are usually iterated: formulation, action, review of results, and refinement (Shneiderman, Byrd, & Croft, 1997).

Especially in multimedia search tasks, users cannot express their information need as (a series of) search requests. As explained before, a query cannot always be expressed verbally, as nonverbal aspects of multimedia like emotional and aesthetic values are hard to capture in words. Because such values are more easily recognized and compared than described or expressed, the search process should be iterative: users *can* tell us which of the retrieved objects are relevant for their (internal) information need. After an initial query has been processed, the user is asked to judge the retrieved objects. These relevance judgments can be used to adjust the query, making it better reflect the user's information need. Query refinement based on relevance feedback has been proven effective for both text retrieval and image retrieval (e.g., Sclaroff, Taycher, & Cascia, 1997 and Squire, Müller, Müller, & Raki, 1999).

Also, using a combination of multiple content representations is crucial for searching a multimedia database system effectively. Manually added descriptions are not sufficient for multimedia retrieval; using features such as those in content-based retrieval systems overcomes some of the problems with textual descriptions, but introduces new problems because most features have only a syntactic value. Although a single representation of syntactic content is usually not sufficient to capture semantic content, the combination of several representations may be more successful. Experiments in image database research, as reported in Minka and Picard (1997), support the hypothesis that the combination of several feature representations improves the results of retrieval. Preliminary results with image retrieval in the Mirror DBMS confirm this as well (van Doorn & de Vries, 2000).

Because these two new requirements—query formulation using relevance feedback and matching with multiple content representations—are not supported by the DBMS, these processes are necessarily implemented inside the user application. Now recall from the section "Data Makes the World Go Round" the main motivation for the development of DBMSs: data independence. Data independence ensures that applications keep running when the internal representation of data changes. Drawing an analogy with data independence helps to pinpoint an underestimated problem with the database support for multimedia as outlined in the previous section: it lacks **content independence:**

Definition 3. *Content independence is achieved when programs that access multimedia data maintained in the DBMS are written independently from the metadata available.*

Applications in a multimedia digital library should be able to use the best available techniques to fulfill information needs. But, in current systems, in which there is no separation between the retrieval application and the metadata extraction techniques used internally, applications have to be adapted every time a new technique for content abstraction has been developed and is added to a media extension. For example, when an ADT is added that has a feature space that is highly effective for face recognition, an application programmer has to know about this new feature space, and adapt already existing programs that could benefit from this new feature space. But the retrieval process modeled in the application does not really change; only the internal representation of content is changed.

Content independence is a notion similar to data independence, but related to the process of content abstraction through metadata. It separates the actual metadata available at some particular moment in time from the query evaluation process. Content independence can be obtained by expressing the part of a query that specifies constraints on the content of the objects only through examples and relevance judgments, without using the metadata explicitly in the query. A more detailed discussion is deferred to the next section.

Like data independence, a multimedia DBMS that provides content independence increases the value of applications by making them less vulnerable to an ever-changing environment. Although retrieval applications would not really break down and stop working without content independence, old applications that do not optimally use newly available metadata do not perform as well as new applications; in a commercial setting, this may encourage customers to try their luck with another digital library. Conversely, in a multimedia DBMS that provides content independence, the client applications will always use all the available metadata; when the DBMS is extended with better multimedia retrieval techniques, its applications benefit automatically from these improved techniques.

The other big advantage of content independence is the opportunities it provides to advance research in multimedia digital libraries. Experimenting with new content representations does not require any changes in user applications, nor in the algorithms that assist the user with query formulation, process the relevance feedback, or combine the evidence from various content representations. Because all interaction involving query formulation is located in a single place in the architecture, logging of interaction patterns becomes trivial, increasing the opportunities to improve our understanding of user actions, as well as the algorithms deployed in the ranking process.

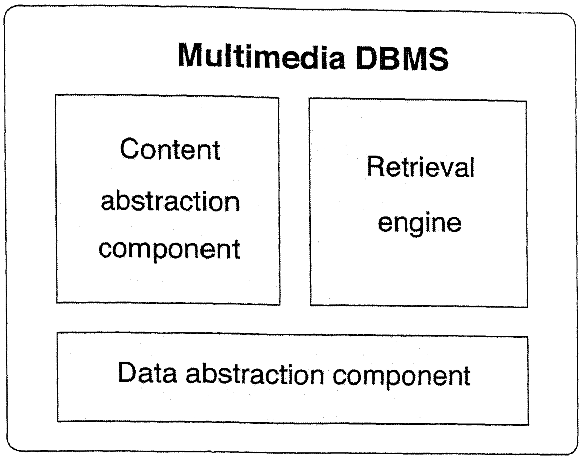

FIG. 1. Multimedia database architecture.

Supporting Content Independence

Figure 1 shows the Mirror architecture, a blueprint of a database system that provides multimedia search while satisfying the notion of content independence. The design is divided into three components. A DBMS provides the basic functionality for data abstraction and efficient query processing. The content abstraction component controls the content-based metadata used to represent the content of digitized object representations, including both descriptions entered by human annotators as well as automatically extracted features. The third component is the retrieval engine, which supports the user with query formulation, using multiple representations for query evaluation. Strict separation between the content abstraction component and the retrieval engine enforces content independence.

Conceptually, the multimedia DBMS operates in two modes (not necessarily mutually exclusive): maintenance and retrieval. In maintenance mode, digitized object representations can be added or deleted, and the content abstraction component can be extended with functionality for new types of metadata. In retrieval mode, the retrieval engine interfaces between the internal models of multimedia content and the user. It keeps track of the different metadata that may participate in the retrieval process. Subtasks of the retrieval process may be delegated to the content abstraction component, such as approximate retrieval methods. The retrieval engine "knows" how to combine evidence from different content representations. It also processes relevance feedback provided by the user, and uses this feedback in the further iterations to refine the initial query.

The retrieval engine can be based on the theory and techniques developed in information retrieval, as demonstrated by de Vries (1998). The resulting query evaluation process is closest to that supported in the (custom-built) Viper image retrieval system (Squire et al., 1999). Its relationship to the content abstraction component and the integration of these components with the underlying DBMS component (taking specific care not to affect negatively the set-oriented nature of query evaluation in database systems) are the research questions addressed by de Vries (1999). An example image retrieval application developed using this framework has been presented by van Doorn and de Vries (2000). Apart from "conventional" content-based retrieval on features, it supports query expansion using text-to-image thesauri associating words in the textual annotations to clusters in the image content representation as an approach to combining the information from different content representations.

The interface between the user (which, again, may be a client program) and the DBMS consists of normal data definition and manipulation languages, augmented with some new primitives. First, the data definition language needs a directive that tells the DBMS which entities of multimedia data types that occur in the schema should be treated as content-bearing objects. This information is passed to the content abstraction component, which will collect metadata about these entities. As explained before, the retrieval engine maintains a dialogue with the user. The interface should provide directives that declare the start and the end of a query session that is related to some information need. Also, the user needs primitives to specify example objects and relevance judgments about retrieved objects. Furthermore, the data control language should have a construct for the specification of operations that implement metadata extraction techniques.

Conclusions and Future Work

Not concurrency, security, recovery, nor any other "goodie" *really* tells a database system from other software artifacts; databases are different from other programs because they put the data in the spotlight by utilizing data abstraction. First, this provides data independence, which is important to reduce the efforts of writing and maintaining application programs. Second, declarative query languages enable efficient query evaluation, which is probably the best explanation for the success of relational database systems: it guarantees ease of use without inducing a performance penalty.

This article has extended the basic ideas behind traditional database management systems for the new domain of multimedia digital libraries. Here, content replaces data as the most important resource. Generalizing the essence of database technology, we proposed the new notions of content abstraction and content independence, and outlined how the design goal of content independence can be realized in the actual implementation of multimedia database management systems, thus providing an integrated approach to content management and (traditional) structured data management.

An open research question is whether content independence can be provided without hiding the processes in the retrieval engine from the user. Occasionally, content-based metadata may be better used directly, most notably for a feature like color distribution, allowing a user to ask explicitly for images with some particular colors in it. This functionality is necessary for the processing of information needs such as: "I want this rose, but in yellow instead,"[4] or, similarly, "I want an animal like this, but with stripes." Especially after some initial iterations, the user may well be able to correct decisions made by the retrieval engine after inspection of the constructed query, similar to text retrieval systems that consult the user before actually expanding a query with potentially useful new terms. We call this approach *query articulation,* in contrast to the process of query formulation now taking place mostly under cover. In principle, support for query articulation does not necessarily conflict with the notion of content independence, for the metadata is not used to address the query formulation problem, but as conventional (derived) attributes of the image. Further research is necessary to develop these ideas.

Other future work focuses on the realization of more complete prototype systems to test our ideas, in particular by experimenting with machine learning algorithms in the retrieval engine. Another important issue in our research is the efficient implementation of the retrieval engine, such that it can scale up to handling millions of images and thousands of videos. The continuing development effort of the Mirror DBMS prototype should (eventually) result in a tool that facilitates the collaboration among researchers from the full spectrum of fields contributing to acquiring a better understanding of multimedia digital libraries.

References

Adam, N., & Yesha, Y. (1996). Strategic directions in electronic commerce and digital libraries: Towards a digital agora. ACM Computing Surveys, 28, 818–835.

Boll, S., Klas, W., & Sheth, AP. (1998). Overview on using metadata to manage multimedia data. In A.P. Sheth & W. Klas (Eds.), Multimedia data management. Using metadata to integrate and apply digital media (pp. 1–24) New York: McGraw-Hill.

de Vries, A. (1998). Mirror: Multimedia query processing in extensible databases. In Proceedings of the Fourteenth Twente Workshop on Language Technology (TWLT14): Language Technology in Multimedia Information Retrieval (pp. 37–48), Enschede, The Netherlands.

de Vries, A. (1999). Content and multimedia database management systems. PhD thesis, University of Twente, Enschede, The Netherlands.

Elmasri, R., & Navathe, S. (1994). Fundamentals of database systems (2nd ed.). Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Grosky, W., Fotouhi, F., & Jiang, Z. (1998). Using metadata for the intelligent browsing of structured media objects. In A.P. Sheth & W. Klas (Eds.), Multimedia data management. Using metadata to integrate and apply digital media. (pp. 123–148). New York: McGraw-Hill.

Hampapur, A., & Jain, R. (1998). Video data management systems: metadata and architecture. In A.P. Sheth & W. Klas (Eds.), Multimedia data management. Using metadata to integrate and apply digital media, (pp. 245–286). New York: McGraw-Hill.

Hardman, H. (1998). Modelling and Authoring Hypermedia Documents. PhD thesis, University of Amsterdam.

Iaccino, J.F. (1993). Left brain-right brain differences. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Minka, T., & Picard, R. (1997). Interactive learning using a "society of models." Pattern Recognition, 30(4), 565–581.

Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., & Faloutsos, C. (1993). The QBIC project: Querying images by content using color, texture and shape. Technical Report RJ 9203, IBM Research Division.

Sclaroff, S., Taycher, L., & Cascia, M.L. (1997). Imagerover: A content-based image browser for the world wide web. In Proceedings IEEE Workshop on Content-based Access of Image and Video Libraries, (pp. 2–9) San Juan, Puerto Rico. Los Alamitos, CA: IEEE.

Shardanand, U. & Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth." In CHI'95 Conference on Human Factors in Computing Systems, May 7–11, (pp. 210–217) Denver, CO: ACM.

Shneiderman, B., Byrd, D., & Croft, W.B. (1997). Clarifying search: A user-interface framework for text searches. D-Lib Magazine of Digital Library Research. Available online: http://www.dlib.org.

Squire, D.M., Müller, W., Müller, H., & Raki, J. (1999). 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.

Tsichritzis, D. & Klug, A. (1978). The ANSI/X3/SPARC DBMS framework report of the study group on database management systems. Information systems, 3, 173–191.

van Doorn, M., & de Vries, A. (2000). The psychology of multimedia databases. In Proceedings of the 5th ACM Digital Libraries Conference (DL'00) (pp. 1-9), San Antonio, Texas: ACM.

Velthausz, D. (1998). Cost-effective network-based multimedia information retrieval. PhD thesis, University of Twente.

Wold, E., Blum, T., Keisler, D., & Wheaton, J. (1996). Content-based classification, search, and retrieval of audio. IEEE Multimedia, 3(3), 27–36.

[4] This example query is due to Tamar Öszu (in Dagstuhl, September 1999, personal communication).