Multimedia Databases

A Literature Study:

Store and Retrieval Methods

Inge La Rivière, 9052100

Sluis Noord 8
3961 ML Wijk bij Duurstede
0343-578667 (privé)
020-4098444 (werk)

CONTENTS

1 Introduction

Multimedia is one of the hypes within Information Technology. Multimedia is just one in the series of ’new’ developments and trends like Client/Server (C/S), Object Orientation, Rapid Application Development (RAD), Data-warehousing and Internet. Everyone is talking about it, various articles appear in journals and numerous books on these subjects are published. Although everyone is talking about it, it isn’t always clear what is meant by it.

Less than a decade ago Client/Server was the topic. There was a tremendous push for it. You just had to enter the C/S era if you didn’t want to miss the boat. When having a discussion about this subject, one can discover that the participants in the discussion don’t understand each other. This is because they have a different idea about the concept. The reason for this can be that there are several gradations of application partitioning which are all called Client/Server.

The same confusion of tongues is the case with Multimedia. Everyone is talking about it, but when asked what it means, you may find out different persons have a totally different idea of the concept. Maybe all one is thinking of with multimedia is a CD-ROM with pictures, audio and video files.

Almost the first thing to do when discussing a subject is to get a clear idea about it. When I talk about Multimedia in this paper, what do I mean by it? The first thing is to describe what will be achieved or attempted within this Master’s Thesis Paper. What is the goal of this project?

Multimedia itself is a broad subject. It is impossible to cover this in full detail. When deepening a specific subject within a certain timeframe, it is mandatory to delimitate it by clearly defining what is and what isn’t part of the study.

Multimedia Systems can cover several areas, like:

- Integration and synchronisation of digital video and audio.
- Information encoding and data interchange formats.
- Operating system mechanisms for handling multimedia data.
- Distribution using networking and communication (digital video and audio).
- Multimedia databases, storage models and structures.

- Methodologies, paradigms, tools and software architectures for supporting multimedia applications.
- Multimedia applications and application program interfaces, and multimedia end-system architectures.

Because of my personal interest in databases, the direction of my study naturally led to multimedia databases. Even multimedia databases alone is a broad subject.

Depending from which perspective one is looking at it, areas within this subject are e.g.:

- Data modelling and metadata.
- Application framework.
- Physical storage and compression techniques.
- Logical store and retrieval methodologies.
- Specific applications (GIS, Medical Image DBMS).
- Multimedia datatypes.

I want to lead this search for information in the direction of content-based retrieval, because that is the area I'm the most curious about.

2 Objectives

As already pointed out in the introduction (chapter 1), in the first instance the general context and meaning of the concept multimedia will be described. The next step will be to discuss the notion multimedia databases.

What I find the most intriguing about multimedia databases is if, and how, questions like *'Give me all pictures that represent airplanes?'* can be solved. This is what I will try to answer in this thesis.

What is the state-of-the-art within information technology concerning this subject? What information and achievements can be found among the existing research?

For these objectives several other concepts related to them have to be discussed first. The discussion of the concepts: multimedia, multimedia database, query and retrieval and content-based retrieval, will be handled in the third chapter.

To obtain an overview of the state-of-the-art, existing research results and operational multimedia systems will be discussed and compared. The review of articles about the researches and two commercially available products is handled in separate sections in chapter 4.

In chapter 5, I will subsequently analyse and discuss the work in chapter 4 by the different aspects they cover. The aspects that have been identified are: metadata, data-driven versus model-driven approach, model and architecture and indexing and matching.

I will end this paper with my conclusions in the final chapter. In this sixth chapter, I will also make a proposal of what a multimedia system should comprise in reality and ideally.

This thesis is completed with the literature references, the list of abbreviations and an epilogue. The list of abbreviations consists of the abbreviations that occur in this thesis. Generally adopted abbreviations are not part of this list unless their common use was doubted. The epilogue contains my personal evaluation of the coming about of this document.

3 Concepts

3.1 Multimedia

A clear and unambiguous definition of multimedia cannot easily be found and given. When going through books and articles, you'll discover that the same underlying meaning isn't always intended, when the term multimedia is used. Most definitions, however, seem to convert to the same meaning.

Multimedia is used by everyone. When someone is telling about an experience and uses both speech and gestures, then this is a form of multimedia communication. Within some branches, like telecommunication and publishing, there exists a different meaning for the term medium and consequently also for the word multimedia. This might cause confusion about the meaning of the word multimedia.

According to Negroponte [Jansen], multimedia is the coming together of 3 business branches: the media world, the telecommunications branch and the computer industry. Because of this diverse historical background multimedia applications cannot easily be characterised. When studying several multimedia applications, features from these branches can be discovered in various proportions.

The background of the media is providing information. Publishing companies mainly supply text, graphics and images, while the television broadcasts sound and moving images. The telecommunication facilitates communication between people, and the computer industry directs to structured data, like numbers and text.

The market for multimedia applications is, among others, the IT-industry, publishing, amusement, health-care, education and marketing. The nature of these applications is as diverse as the market. Often fast communication plays a crucial role.

Multimedia applications you may be able to think of are electronic publishing like multimedia-encyclopedias, computer games, medical information systems (patient records with X-rays), computer based training (CBT) and tele-education, company and product presentations, and not forgetting surfing the Web: Internet. The latter has almost become synonymous with multimedia.

These applications sometimes fill a new market based on new needs. Multimedia also operates on existing markets, like the one of computer games with ’better’ and flashier computer games. For marketing it serves as extension for the current advertisement media. Interactive home shopping is just an extension to the existing mail ordering services.

Multimedia is intended to make communication more clear. Illustrations are used for this purpose, and also to make it more attractive. It shouldn’t, however, distract from the real message. Often multimedia techniques are used without beforehand considering the benefits compared with traditional means.

The use of multimedia techniques within an application should create a certain surplus value. Otherwise one takes the risk that it will hardly be used, or even not at all. Unfortunately the effects of multimedia are hard to quantify.

Sometimes it is even impossible to express them in terms of reduced costs or increased profits. The prevailing opinion is, however, that multimedia causes a more effective transfer of information by integrating data in different presentation forms [Gurchom].

When analysing the meaning of the word multimedia, it should mean something like consisting of many or multiple types of media. This definition is comparable with the one Kay and Izumida give for multimedia information in a database context.

According to the Master’s Thesis of Suijker on IBM’s Digital Library [Suijker], multimedia information is text (alphanumeric information), images, video (fragments) and audio (fragments). He also points out that, according to some, multimedia data excludes the alphanumeric type.

Until now the definition of multimedia hasn’t been more than the series of datatypes it can consist of. What distinguishes Multimedia from a collection of datatypes? Suijker describes the concept Multimedia itself as computers using different kinds of audio-visual means to let people use information as natural a way as possible.

In *’Toward Multimedia’*, Cheyney *et al.* put the emphasis on the integrated whole of text, graphic, audio and video information. A multimedia application should contain extensive provisions for random access and hypermedia linking. In this article the process of creating the proceedings of an academic conference in multimedia format is described [Cheyney].

With this multimedia format they want to overcome the shortcomings of the traditional means. They also want to capture the conference atmosphere as much as possible. By describing the process of creation and the problems that occurred during that, they hope it will serve as a model for future conference publications. Their method can also be applied to multimedia textbooks and learning environments.

In last year’s special of the journal *’Informatie’* on multimedia, both van Gurchom and van Rijssen, Hoogeveen and Noordzij [Gurchom, Hoogeveen, Noordzij] agree with each other on what multimedia is. The essence of multimedia relates to the integration of different presentation forms or information types. The types are the same as the aforementioned.

With integration one means synchronised and interlarded. Another will define integration in this context as a strong relation between the data by means of links and references (e.g. hyperlinks). In this sense they seem to agree with Cheyney.

Hoogeveen and Noordzij also consider the interactivity for the user as an important aspect of a multimedia application. Users should be able, up to a certain level, to decide which information they want to receive and which they want to respond to.

In the same issue of the journal *’Informatie’* Jansen and Koster [Jansen] characterise the same two phenomena (integration and interactive) that can be distinguished for multimedia applications. They also identify informative, imagination, independence, individualising and intuitive as characteristics for multimedia applications. In their opinion, these seven I’s can be used to determine the extent of multimediality.

Multimedia applications will always be informative by supporting information interchange processes. Imagination is merely a consequence of integration, because the message can appeal to different organs of sense. The combination of telecommunication and computers creates the independence of place and time for information.

The characteristics individualising and intuitive are both strongly related with how Hoogeveen and Noordzij define interactivity. The individual demand for information and the intuitiveness of the user interface are already part of that concept.

According to the above information, multimedia varies from a collection of multiple kinds of information types via this same collection presented as an integrated whole to a natural and human-like interaction with the computer.

3.2 Multimedia Databases

Part of a multimedia application is the data. The diversity of multimedia data can be categorised in several ways:

time-dependency time-dependent data has a duration in time like sound and video, this is not the case with time-independent data like images or text.

dimension spatial (3D) occurring with GIS and CAD/CAM and non-spatial data (2D) can be distinguished from each other.

by sense by which organ of sense it can be perceived, like ear (sound), eye (image), both or possibly other senses.

Another classification of data, by Lorie, is based on how the data is formatted. All these kinds currently exist in the application areas of advanced database systems [Gudivada96]:

formatted traditional alphanumeric data

structured heterogeneous data about an object is stored and retrieved together as with structures or records in programming languages.

complex structured data with a variable number of components.

unformatted string data whose structure is not understood by the DBMS, like the BLOB (Binary Large OBject).

A very rough definition of multimedia information within a database context is given by Kay and Izumida [Kay]: consisting of one or more (according to some: two or more) of the following forms: image, text, diagram, graphical animation, sound and moving pictures.

According to some definitions, a multimedia database is a database which contains one or more types of information. According to others, at least two types are needed, because otherwise it isn’t possible to use ’multi’. In my opinion, this definition should be extended with the addition that a multimedia database potentially can contain multiple media types.

When storing images and text there are already two types involved. Most people will associate multimedia with flashy presentations with sound, video, images and supporting text.

Ideally multimedia data will be stored in a suitable DBMS in a standardized and integrated manner. A multimedia database should provide support for multimedia applications as well as possible. This can be by offering fast search coupled with the ability to handle a large variety of queries. According to Blanken and Apers [Blanken] databases provide more and more support for multimedia applications.

Until recently there was hardly any multimedia database support available. More than the BLOB wasn’t available in most DBMS. Now this has almost become a standard feature.

A BLOB isn’t considered to be an object, but is unstructured binary data [Kay, Colton94, Colton95]. This raw data can be anything. The database system doesn’t know the underlying datatype and has no knowledge about the internal format of it. For this reason, the database system doesn’t know what operations are possible.

As no operations are provided by the DBMS, also no internal components can be retrieved and no indices can be created on it. The only thing a DBMS can do is store and retrieve it as a whole. In this sense, a BLOB seems to be an unsuitable datatype for storing multimedia information.

Another way of looking at multimedia databases is by which characteristics are important and how it distinguishes itself from traditional databases. Kay and Izumida [Kay], Blanken and Apers [Blanken] and Faloutsos *et al.* [Faloutsos] have described a number of them. Also the work of Huijsmans and Lew [Huijsmans] and that of Ma and Manjunath [Ma] more specifically look at image database characteristics. Smoliar and Zhang [Smoliar] look at the accessibility of video content.

isochronality and time-dimension of data	Continuous and synchronised delivery of the data is important for moving pictures and sound. Further, synchronisation of sub-objects.

internal structure	Although multimedia information often is considered unstructured, it really has a very complex structure.
size of the objects (and databases)	Traditional database systems were designed to work with millions of records varying from a few bytes to kilobytes. The size of multimedia items can be immense (megabytes or even gigabytes) and results in large-scale databases (petabytes).
external structure	This consists of relations between different media, often via so called hyperlinks.
query	In a multimedia database it often isn't the purpose to retrieve facts, but to find documents in order to get at the facts. Combination of various query mechanisms.
navigate and browse	Access and ad hoc-retrieval based on links. User-guided navigation.
long transactions and high concurrence	Many users have access to the same large multimedia items at the same time, mostly by reading (updates are scarce) and viewing.
presentation	The importance of the presentation and a (visual) query and user interface is often neglected.

In the work of Huijsmans and Lew [Huijsmans] especially the time-dimension and internal structure characteristics play a role in their retrieval methodology for images for the following reasons:

- sound and images are difficult to separate in time and space;
- the signal is low compared with the noise;
- the diversity of the appearance of the same source-object is complex;
- the meaning and importance of (parts of) the images can be totally different for the observers.

These characteristics of multimedia data lead to several prerequisites and necessities for multimedia technology in order to cope with multimedia information [Colton94, Colton95, Kay, Smoliar, Suijker].

1. query and navigation, or indexing and retrieval,

searching for and access to data by content-based retrieval and via browsing.

2. independence of storage format, also proper classification and representation, flexibility by extensibility (user-defined datatypes and functions).
3. optimal and efficient storage and management of large objects. (availability, storage, performance, distribution and compression techniques)
4. partitioning (elemental units) and modelling of time-dependent media.
5. web publishing (integrity of links) and other interactive tools.
6. configurable provision of security for the information, protection against unauthorized use, by marking (watermark), encryption, metering and billing.

Many of the prerequisites for multimedia are supported by Object Oriented Databases (OODB) and Object Relational DBMS (ORDBMS). The suitability of OODB and ORDBMS for multimedia data lies in the following [Kay, Colton94, Colton95]:

ad 1 The combination of query and navigation is supported by Object Database Systems. These systems try to unify the declarative and navigable access on the same level for model and language. Examples of these systems are: ODB-II, UniSQL, Illustra and Odaptor (HP). These systems are sometimes called object relational.

ad 2 The object oriented techniques encapsulation and polymorphism are designed to deal with variety and change and also with independence of storage format. Encapsulation takes care that the stored data stays hidden behind a query interface.

Polymorphism makes it possible for implementations of methods to exist next to each other. Illustra is extensible, a developer can write functions of any degree of complexity that can be dynamically linked to the DBMS and executed against both user-defined as system datatypes.

ad 3 A large object will generally be stored as a chain of fragments. With ODB-II and Illustra, multimedia data can be managed externally and internally. Externally the data is stored as files and the names of these files are stored in the database. With internal management the fragments are stored as a series of separate objects.

ad 4 Time-dependent data also has to deal with point 2.

Further spatial or time-dependent relations can be defined between e.g. video and subtitling. Also complex manipulations can be defined like colour separation, noise reduction and the transition between video images. Object oriented techniques can also be used to represent these complex combinations and derivates of relations.

In 1995, a workshop supported by the National Science Foundation was held on the future of database systems research [Silberschatz]. One of the categories discussed in this workshop was multimedia. It resulted in five principal areas for research involving multimedia data. These are:

1. Tertiary storage (related to 3. of the prerequisites), like optical disks, CD juke-boxes or tape silos with the use of a mechanical arm to physically move the desired tape cassette or CD to a reader.

2. New datatypes for each form of multimedia information with its own collection of first-class concepts (operations and functions) along with a high performance implementation involving appropriate data structures and access methods.

3. Quality of Service by timely and realistic presentation of the data in its desired form.

4. Multi-resolution query (as in 1. of the prerequisites) by best match instead of precise match retrieval by content. Retrieval according to imprecisely defined characteristics (such as shape, colour and texture) creates a need for new query languages or extensions to old ones, that incorporate the degree of precision and modes of relaxing the requirements of a query as first-class notions.

   The advanced searching methods work mostly in two steps. First is content creation and capture by scanning, recognition, compression and transformation techniques. Second is the search for shapes, sounds, colours and texts within video scenes.

5. User interface support. Multimedia data often require new user interfaces that must be supported by DBMS, and new means for browsing, searching and/or visualizing the content of massive data objects quickly and efficiently.

3.3 Query and Retrieval

When you want to obtain information from databases, most of them offer a kind of query interface. This query interface can be a language, like SQL, but it also can be a structured and visual user interface. The interface is mostly specific for the database, but there are also tools which provide an interface to databases from various suppliers.

Such an interface helps and guides the user. It shows which tables or objects are present. It also offers a list of fields or attributes to select from. The user can also specify, by choosing from lists, by which criteria the data have to be retrieved and how they must be ordered. Such an interface is often called Query by Example (QBE).

For multimedia information, the query and retrieval of data is a completely different story compared with query and retrieval of the traditional alphanumeric datatypes. Alphanumeric data can be ordered, indexed and searched for in a straightforward manner.

For alphanumeric data, defining that an insurance number equals to a specific value, or a date of birth must be greater than a certain date and the ordering of addresses must be by the zip-code is clear for most people.

Handling multimedia information isn’t that easy. Techniques that have been developed to deal with alphanumeric data cannot simply be applied on non-alphanumeric data. The information content of images and other multimedia data is not explicit and does not easily lend itself for classification, indexing and retrieval.

How do you specify a query, if you want a picture with an airplane or a piece of music that contains a certain note pattern? How will these patterns be recognised? Also sorting the retrieved pieces of music in the order that the first is the most alike and the last is the least alike. The selection and order criteria are the most significant and important in this respect.

Gudivada *et al.* worked on picture retrieval systems. In that respect they distinguish and describe five retrieval classes for image databases [Gudivada96]:

1. Retrieval by BRowsing (RBR);
   By using a user-friendly interface one can browse through (’thumbnail’) images.

2. Retrieval by Objective Attributes (ROA);
   Formulating a query (SQL) based on the meta and/or logical attributes. Retrieval of images which match perfectly.
3. Retrieval by Spatial Constraints (RSC);
   Query based on the relative spatial relationships between objects in an image. Within this two categories can be distinguished: relaxed (best match) and strict (precise match).
4. Retrieval by Shape Similarity (RSS);
   Query based on shapes of domain objects in an image.
5. Retrieval by Semantic Attributes (RSA);
   Query by specifying an exemplar image, just as QBE, all conceptually, semantically similar images are retrieved.

The last three classes are considered to be forms of content-based retrieval (CBR). For that reason they can also be considered to be one retrieval class with three subclasses. Querying by the content of a multimedia object or content-based retrieval is what I am most interested in. Content-based retrieval will be discussed in more detail in the next section.

The opposite of such a form of querying are the first 2 classes. In order to query multimedia information, a user should be provided with some sort of tool or mechanism to do so. Often multimedia systems work with logical attributes or keywords, as the second of the five classes described above.

With this form, descriptive information about an item is stored together with the multimedia information. Instead of querying for a picture with an airplane, a picture of which the keyword contains ’airplane’ is requested. In this sense the querying of multimedia information works as with alphanumeric data.

Keywords are useful but several problems occur with this approach [Faloutsos]. The originally chosen keywords do not allow for unanticipated search in applications that want to use the data as well. More important is the inadequacy of defining uniform textual descriptions for categories like colour, texture, shape, etc.

All this deals with the underlying meaning of a description different persons give about a scene. The interpretation of a situation can vary a great deal from person to person. With police work this also occurs when questioning or interviewing witnesses of a crime. Their stories can differ significantly.

Some visual properties are nearly impossible to describe. Also, there is no commonly accepted vocabulary for describing image properties. As result a ’curvy’ item may not match a ’wavy’ item [Niblack].

With respect to these keywords or logical and meta attributes, I found an interesting article from Kashyap *et al.* [Kashyap], which discusses several aspects of metadata. They are working on the problem of the ’semantic bottleneck’. This problem has to do with the lack in multimedia systems of the capability of correlating information at a semantic level across different representations.

For their article they have made a classification for the various kinds of metadata used by researchers based on whether or not they are based on data or information content of the multimedia items. The basic kinds of metadata they identify are:

Content-dependent — Depends only on the content of the data.

Content-descriptive — Special case of content-dependent metadata. It describes the content in some way, but it cannot be extracted automatically. An interactive cognitive process or an intellectual derivation with support from a tool is needed.

Content-independent — Does not depend on the content. Can be derived independently from the content of the data.

Next to the five pictorial retrieval classes and the distinction of the three basic kinds of metadata, there are other ways of categorising the approaches of retrieval. Another ground of categorising is by whether the retrieval is based on a model or on the data. This way is described by several researchers [Jagadish, Kashyap, Smoliar].

The first, model-based (also called model-driven or application-driven), assumes some a priori knowledge, namely the model or the application area, about metadata or on how images are structured.

The other approach, data-based or data-driven, requires a more general model of which features should be examined and how to be compared for proximity. Features that have shown to be most effective are: colour, texture, shape of objects and relationships among edges.

3.4 Content-Based Retrieval

As already stated in the introduction, I want to direct this thesis in the area of content-based retrieval. Immediately the question arises: what is content-based retrieval? As the term simply indicates, it is retrieval based on the content of an object, in this case a multimedia item.

With ’the content of multimedia items’ I mean: which features or properties can be distinguished within a multimedia object. The features that can be recognised within such an object depend mostly on the type. For images or other visually perceptible objects, there are different patterns and characteristics than for sound.

For images, features one can think of are colour ratio, colour pattern, texture, shape and spatial relationships between objects in an image. For sound the occurrence of a note pattern or melody within a piece of music is such a feature. Also phrases in sung music or spoken words are features that are likely to be recognised.

Ma and Manjunath [Ma] focus in their paper on image feature extraction based on local intensity patterns (texture). They also give a summary of recent research in pictorial queries in a broad classification. They make a distinction in low level features, shape primitives and high level features.

For the first category no domain specific or context information is required. Examples of these are based on image attributes like colour/histogram and texture. For the last one, context and domain specific information is required, and sophisticated image analysis tools for detecting and extracting feature information are needed. These are used for human face features, medical image data and satellite images.

The shape primitives category uses an intermediate representation in relation to the other two categories. The shape and contour information require good segmentation algorithms to detect object/region boundaries. Polygonal approximation is used, but mostly limits the application to highly structured objects. This is the case in the work of Jagadish on rectilinear shapes (see the next chapter).

As mentioned earlier, by referring to the work of Kashyap *et al.* there are also features that cannot be extracted automatically. They need a cognitive or intellectual process to be derived. There could be thousands of features which could be recognised. What one person finds remarkable or important can be totally irrelevant to another.

When someone wants to buy a music CD, it is possible to search for it by composer, performing artist, title and so forth. When you do not know all of these, but are able to hum or whistle a few bars of it, this pattern could have been the key to the item you are looking for.

In the past and present you could see if the salesman recognised your tune. If it is classical music and the seller is only interested in popular music there is a good chance he won’t recognise or know it. Even if he knows it, the whistled part can be insufficiently recognised by him to determine what you want.

If one had a proper multimedia system, this should be able to record your tune. Based on the characteristics of that recording, music with similar themes could be retrieved in order of their correspondence.

The same is the case with images. What one person finds striking in a picture, like a house, could be a totally irrelevant detail to another, because the other person may notice the colours of the sunset or the tree in the front-yard.

What some researchers call the ’semantic bottleneck’ [Kashyap] has another aspect. In addition to those items a person finds interesting or striking in an image, attaching a semantic meaning to an object is also a problem. The process of grouping image features into meaningful objects and attaching semantic descriptions to scenes through model matching, called perceptual organisation, is an unsolved problem in image understanding [Flickner, Smoliar].

It is possible to ’teach’ a system what a fish-shaped object is. The system is able to find images with a shape close to the given shape, but fishes have many shapes and fishes with a different shape won’t be found. The semantic query of finding all pictures with fish is still beyond our reach [Flickner].

As an example a situation in an artificial intelligence conference was described. The audience was challenged to write a program that would identify all dogs pictured in a children’s book. This is a task that most 3-year-olds can easily accomplish. Nobody in the audience accepted the challenge.

Another aspect of content-based retrieval which distinguishes itself from traditional searches is that it isn’t suitable for an exact match. Although there could be an exact match, one is more interested in, e.g. an image that looks like a given image. In this manner CBR acts as an information filter.

Content-based searches work by approximation, the retrieved objects, like images, are typically sorted by similarity to the posed query. Usually only a (specifiable) fraction of the top ranked images are displayed [Faloutsos, Niblack].

4 Existing Research and Operational Products

In this chapter the work of various researchers will be introduced and reviewed. Also some of the available products that present themselves as tools for content-based retrieval will be dealt with. Each research or product will be handled briefly in a separate section. In the next chapter the similarities and differences will be discussed.

Most of the discussed work in the sections hereafter will only deal with images. Some are also more generic or also deal with video information. During the collection of the ground materials for this thesis, I mainly gathered information about images. This was mainly because of my interests. Also most of the available information and product handle is about images.

In the first sections the more theoretical research articles will be summarised. They merely deal with the modelling of multimedia information. Following on from that the more domain-specific applications will be treated. This will be rounded off with sections about two commercially available products.

4.1 Metadata

Kashyap *et al.* [Kashyap] present a three layer architecture to enable the correlation of information at a semantic level across multiple forms and presentations. Humans have the capability to abstract information efficiently and can accordingly correlate information at a higher semantic level. This capability is lacking in current multimedia systems and has been characterised by them as the *semantic gap*.

Their three layer architecture consist of:

1. ontology — content of the information in a database irrespective of the media type
2. metadata — information about the data, extension on the database schema/catalog concept
3. data (base) — actual (raw) data

The metadata typically represents constraints between individual media objects, content-independent information (like location, time of creation) and content-dependent information (relief of geographical area). The metadata level is the most critical level. It should be able to model the semantics (meaning and use) of the data.

In the perspective of the process of construction of metadata, an application-driven and a data-driven approach is distinguished by them. These approaches work in different directions; top down and bottom up.

In a top down or application-driven system the design of metadata is influenced by the concepts in the ontology. This approach refers to the concepts and relationships inspired by the class of queries for which the related information in the various media types is processed. With this approach mostly domain-dependent concepts are identified.

In a bottom up or data-driven system the metadata is extracted from the data. For this perspective the concepts and relationships are designed by interactive identification of objects, and the related information is stored in the database corresponding to different media types. Typically, domain-independent and, for the various media types, specific concepts are identified and generated by this data-driven approach.

They identify the domain-dependent, content-descriptive and media-independent metadata as those best suited to support the semantic correlation. This is because the metadata should model the meaning of the data and should therefore capture as much media-specific information as possible. Also the metadata should view the data independent of the representation medium.

In order to make these statements, they have compared their work with the research of many others. Still, many research challenges should be met to overcome the ’semantic bottleneck’.

4.2 Unified framework for a multimedia information system architecture

Marcus and Subrahmanian [Marcus] introduce a theoretical framework for multimedia database systems. Current multimedia systems in the commercial market have primarily been developed on a case-by-case basis. They think that the large-scale development of these systems requires an approach that is independent of a single application.

They view a multimedia system as a finite set of media instances. A media instance is an instance of a media source, which contains represented information that is unique for that medium. They use a formal language to specify the notion of a media instance.

A media instance may be thought of as *glue* residing on top of a physical media representation of a specific kind, such as video, audio, document, etc. With the use of the glue, it is possible to define a logical query language to query multimedia data for general purposes.

The glue consists of a set of *states* (e.g. video frames, audio tracks, etc.) and the (relevant) *features*, together with relationships and properties between states and/or features. In their opinion, any media source (e.g. video) has an affiliated set of possible features. An instance of a media source possesses some subset of these features.

The glue must be seen as general-purpose access structures. With the use of this notion of a structure, it is possible to define indexing structures for processing queries and procedures to answer these queries and their relaxations, when appropriate. The indexing structures and algorithms have shown to be relatively efficient (polynomial-time).

A general scheme is developed that, for the set of media-sources and a set of instances of those media sources, builds additional data structures, called *frames*, on top of them. Various interesting aspects or features can be represented by them. They can be used to access multimedia information efficiently.

The main advantages of their indexing scheme is the independence of the data structures used for the physical storage of an image and the irrelevance of the nature of the medium like audio, video, bitmap, etc.

4.3 A framework for an image retrieval model

As the titles of both the articles of Gudivada and his co-writers indicate [Gudivada94, Gudivada96], does this research involve a uniform approach for image database applications? A framework for an image retrieval model is described under the name AIR (adaptive image retrieval). The term *adaptive* is used, because the proposed framework can easily be adapted.

An image data model (IDM) is a scheme for representing entities of interest in images. Also the geometric characteristics and attribute values of entities, and associations among images are part of the model. An image retrieval model (IRM) consists of an IDM, a way for defining user queries, and matching and retrieval strategies.

The AIR model is designed to efficiently store and display images. It consists of the constructs image, image-object, image- and image-object-base-representation, image- and image-object-logical-representation, semantic attributes, rule-programs and meta attributes.

An image may contain many image-objects. These are both modelled. The notion of an image-object is domain-dependent. The relevant ones are determined by the users at the time of insertion of the image in the database.

The image- and image-object-base-representation are representations for the physical level and provide (multiple) persistent storage for raw or unprocessed images and image-objects. Also storage structures for logical attributes are provided.

The image- and image-object-logical-representation (respectively ILR and OLR) model various (simple) logical attributes and (complex) logical structures. They are derived directly and do not require domain expertise.

The ILR describes the properties of an image as an integral entity. An important aspect is the use of logical structures for implicitly modelling the spatial/topological relationships. Geometry-based logical structures are used to model the spatial/topological relationships among the image-objects.

The OLR describes the properties of an image as a collection of constituent objects. It is derived from the base representation using automated domain-dependent interpretation techniques, manual interpretation through human involvement, or a combination of both. After the identification of the objects, the logical representation and attributes are generated automatically, if they can be derived based on the objects geometry.

Generic logical structures that can be used are: Spatial Orientation Graph (fully connected weighted graph), Plane Sweep (2D Sweep Line), ΘR-String (radial Sweep Line), Skeleton, Minimum Bounding Rectangle (MBR, for determining of two objects intersect) and 2D-String.

Semantic attributes capture the high-level domain concepts that the images and image-objects manifest. They can be derived by applying user-defined transformations on the base representations, meta attributes and logical representations, either in an automated fashion or with considerable human involvement. Some subjectivity is involved, because the richness of information can lead to different interpretations of the same image.

The Personal Construct Theory (PCT) can be used for the elicitation of the semantic attributes by a domain expert. PCT assumes that people use *constructs* in evaluating their experiences for decision making. A construct or cognitive dimension is a property of an element that influences a person’s decision making process.

A set of rule-programs is used to synthesize the semantic attributes and to provide the transformation process at the semantic level. Besides these logical and semantical attributes, externally derived meta-attributes are also part of the AIR model. These do not depend on the content of the image and image-objects.

Observation of the AIR data model results in a framework that can be divided into three layers. From bottom to top, these layers are: the physical level representation, the logical level representation and the semantic or external level representation. It is also referred to as the AIR architecture.

As stated before, the lowest level in the architecture, the physical level representation consists of the image- and image-object-base-representation. The ILR and OLR comprise the middle level, the logical level representation.

The topmost layer in the AIR architecture hierarchy is the semantic level representation. It models the semantic views of the images from the perspective of individual users or user groups. It establishes a mapping mechanism for synthesizing the semantic attributes from meta attributes and logical representations.

4.4 Approximate matching for image databases

A generally applicable system for picture retrieval based on *approximate matching* is described by Sistla and Prasad [Sistla]. The contents of a picture is a collection of objects related by some associations. Images can thus be represented by ER diagrams:

- entities the identifiable objects in an image;
- attributes the characteristic or qualifying properties for these objects (colour, size, state of motion, etc.);
- relationships the associations among the objects (spatial, actions).

The metadata that represents the contents of the pictures are currently created manually. It is assumed that the metadata associated with each picture will be generated a-priori and is stored in a separate database. The generation can occur via image processing algorithms, manually or a combination of both. They expect that computer vision and pattern recognition techniques will help in automatically identifying some of the objects and the relationships.

The user interface must be kept as simple as possible in order to let the user interact easily with the picture retrieval system. An interface with icons has been developed that guides the user step by step in specifying the contents of the picture the user has in mind. Supporting features for identifying the objects, their characteristics and the associations among objects are provided. A user-query is specified by the properties of several objects and the relations between the objects.

The similarity between the query (*Q*) and an image (*P*) is given by a similarity value. The value (*f(P,Q)*) is given by a function (*f*). The higher the value, the greater the similarity. When multiple objects exist within *P* or *Q*, the maximum similarity value computed over all combinations of objects in *P* and *Q* is the result of *f*.

The similarity of each picture is computed individually with respect to the given query. With a large database of pictures this is not feasible. By using methods that make use of indices to facilitate efficient retrieval this can be solved. This is, however, not part of their article.

Based on these concepts a prototype has been built. The preliminary experimental results are encouraging.

4.5 Pattern-recognition with shape matching

In multimedia databases one is not usually looking for an object about which the exact conditions it meets are known. Objects are often queried based on conditions that approximately meet the specified query. Jagadish addresses the question 'how to construct an index structure that can enable efficient retrievals by *similarity*'.

Jagadish [Jagadish] uses a *data-driven* pattern-recognition technique for shape matching within his research. Two different types of pattern-recognition can be distinguished with the following consequences:

- model-driven — the given shape has to be compared individually against each shape in the database, or at least with a large number of clusters of features.
- data-driven — by constructing an index structure on the data, given a template-shape similar shapes can be retrieved in less than linear time.

Based upon the notion of similarity, the technique is to obtain an appropriate *feature vector* for each object. The research restricts itself to specific shaped objects, namely rectilinear shapes in two dimensions. Area-differences are used as the measure of similarity.

The 2D-rectilinear shapes can be covered fully by placing all kinds of rectangles over it. The types of rectangular covers used in the paper are additive (a union of several rectangles) and general (by addition and subtraction of rectangles). The shape of an object is described by the relative position of these rectangles.

The coordinates of the centre point and the size (length Y, width X) of the rectangles are determined. The position of the first rectangle serves as the origin for the calculation of the others. The X- and Y-values (for the size) of the first rectangle are used as a divisor for the others (size and position). The natural logarithm is taken from these normalised measures.

For an object existing of K rectangles the following information on the feature vector will be stored:

- For the first rectangle
  - shift factor (X, Y)
  - scale factor (X * Y)
  - distortion factor (Y / X)
- For the other K-1 rectangles (after shifting and scaling):
  - (X, Y) coordinates for the centre points
  - (X, Y) values for the size

The query types that can be performed based on the above structured information are:

- full match
  correct shape and position
- match with shift
  correct shape, position unimportant (without shift factor)
- match with uniform scaling
  size and position are unimportant, as with different distances from the camera (without shift- and scale factor)
- match with independent scaling
  independent scaling for X- and Y-axes, as with pictures taken from different angles (without all factors)

Approximate matching can be implemented by allowing error margins on the position and size factors of the query shape descriptions. A more subtle way for obtaining approximation is by not using all the rectangles in the description of the query shape. Mostly it is sufficient to use the first *k* of K for indexing. If the index of the shape database is constructed on *k* rectangles this will be an upper limit. The error margins and number of rectangles can be implemented by means of parameters.

The practical utility of the proposed technique has been verified in an experiment. A database of 16 thousand synthetic shapes was constructed. Each synthetic shape is a fusion of 10 randomly generated rectangles. The shapes returned from the database in response to an approximate match query (*k*=3) are somewhat similar.

The question is, are the returned shapes indeed the most similar shapes in the database? The answer to this question is subjective because the database was too large to be studied by a human. The technique, however, is expected to have found the four best matches out of a returned set of forty shapes.

4.6 Texture-Based Pattern Image Retrieval

Ma and Manjunath [Ma] describe their approach to image data retrieval by queries based on local intensity patterns. In their paper they focus on the extraction of image features.

The steps for transforming an image to a feature representation consist of:

1. Identifying salient image locations for extracting information manually or by some automated technique.
2. Computing local feature representations based on a Gabor wavelet decomposition of the local intensity pattern for each of the locations from step 1.
3. Clustering the features of step 2 to obtain a compact representation. For the clustering, an algorithm by Duda and Hart is used.

Salient image locations are locations where a human observer would look. Developments in computer vision include the detection of low level features like edges, corners and line endings. Further, algorithms based on local gray level statistics and Gabor wavelet based feature detection can be used. In some cases domain-specific knowledge must be used. For textures a set of various locations can be selected randomly.

Two experiments are described as an illustration that the proposed approach provides a powerful tool. In both experiments 48 different texture images are used. The image data set is obtained by digitising texture images (512x512) from the so called Brodatz album. These textures are e.g. cane, coffee beans, grass, raffia weave, reptile skin, woven matting, etc.

The first experiment demonstrates that the local perceptual similarity is preserved by the use of the Gabor wavelet decomposition and the grouping strategy. A dapabase with 7680 feature vectors obtained from the 48 texture images is used.

For each texture image 160 different image locations with a size of 40x40 were selected randomly out of the 512x512 images. The Euclidian distance between each test vector and the other vectors in the database is computed. The top 10 of the features with the smallest distance are identified and ranked.

The results show that in nearly 85% of the cases the first feature in the ranking is a correct match. When more hits are considered, this percentage of success even reaches 90, 95 and 97.5 by respectively 2, 5 and 10 top rankings of which at least one is a correct match. The retrieved patterns, that are not correct, are visually very similar to the input pattern.

In the other experiment, the performance of feature representatives is evaluated. Feature representatives are obtained from a large image segment (128x128) by local feature clustering and are characteristic for the different local intensity regions of the larger texture image.

By clustering the local feature vectors, all intensity patterns in the same cluster will have a similar structure. From each of the ten largest clusters, the image pattern closest to the cluster centre is selected as representative for that cluster.

The usefulness of the clustered feature representatives is tested by dividing each of the 48 texture images into 16 subimages (128x128). This results in a set of 768 images from the 48 classes.

The results show the top 7 feature representatives and, associated with each of them, the top 3 candidate matches of the retrieved local intensity patterns, which are feature representatives of the textures in the database.

The worst cases show that the different local regions are not very homogeneous. This does not occur with the images with homogeneous, regularly repetitive patterns. In the latter case only a small number of feature representatives is enough to effectively characterise the entire image. In some cases the top matches look similar, although they are not from the same texture class.

A quantitative measure of the performance is calculated to obtain the retrieval accuracy. For any of the 768 images the top 15 matches are retrieved. Because each of the images has 15 other images with the same texture, the evaluation is based on the presence of these in the top matches.

When all 15 matches are considered a classification accuracy of 81% is obtained. When considering 12, 5 and only 1 (the top match), these percentages become respectively 85, 90 and 94. With another evaluation is calculated, that in only about 40.5% of the cases all of the 15 matches were correct patterns. When a minimum of correct patterns of 13, 7, 5 and 1 is considered out of the 15, an accuracy percentage of respectively 60, 90, 96.5 and 100 is achieved.

The results of these two experiments for the proposed feature representation scheme show it has a very good performance with retrieving the correct texture class for most of the homogeneous image patterns. The experiments also illustrate that, unlike with alpha-numeric search, the results are imprecise by nature.

Their future work includes investigating methods for efficient multidimensional indexing instead of the current implementation of sequential search.

4.7 Content-based Image Retrieval

Huijsmans and Lew describe in their article [Huijsmans] an opening to image queries based on the visual contents rather than thematic search patterns. In their opinion databases are increasingly used for the storage of digitised speech, images and video sequences.

A growing number of researchers try to find image query equivalents of the successful text-pattern matching and text-indexing techniques. A generalisation of the text-retrieval methodologies is hindered by the aforementioned characteristics of multimedia data. These circumstances make it difficult to represent and specify the contents of an image for effective and efficient querying.

Their opinion is accordingly that efficient retrieval by indexing and compact specification of the search image can only be done for a specific application area. Only then is it possible to determine which features describe the contents of an image.

The chosen pictorial objects for their experiment are 19th-century black and white (B/W) studio portraits. These portraits are highly standardized and produced in mass and are for that reason very suitable for this purpose. They reduced the search space by dividing the *copy location* method into two phases; pre-processing and matching.

The method is called the copy location method, because it is used within the problem area of locating the copies and the former copies of a portrait photo in a large picture database. The (near-) copies were originally copies, but through time they now have a different appearance. Factors like fading, dirt, stains, writing, labels and cutting are the reason for this.

The goal of the pre-processing phase is to eliminate most of the effects of scanning by normalisation. The scanning effects like position (translation), rotation, scale (resolution in dpi) and lighting (intensity, contrast) may otherwise play an important role in the matching phase.

To obtain an automatic normalisation, objects should be imaged against a uniform background, which has sufficient contrast. In this case, the thickness of the photo carton produces a small shadow on the background, which makes it easy to find the minimum enclosing rectangle.

Once a uniform and sufficiently contrasting background has been established, the following boundary conditions must be met during scan-in:

- highly rectangular shape of the pictures, eventually with rounded corners;
- gray-values within the range of the scanner, no under- and over-exposed parts;
- recording the scanning resolution.

The normalisation phase consists of the following steps:

1. finding the sides of the rectangular object
2. rotating the object in upright position (portrait)
3. removing the background
4. normalise the lighting conditions
5. calculating gradient magnitude image
6. threshold and binarise gradient magnitude image

These last 2 steps are only performed when gradient or binarised gradient images are to be compared.

With the matching phase the same and similar pictures are retrieved by ranking the pictures by a measure of difference. With the ranking the least different comes first.

To make the comparison of the different methods quicker, a visual QBE interface is developed: VSPDB (Visual Search Photo DataBase). This is built around the portrait database and the precalculated ranking results of all images against every other image. According to Huijsmans and Lew it is important that the quantitative measure of similarity should be an integral part of this interface.

As a measure of difference (or similarity) the magnitude of the average difference in intensity between 2 pixels or a gradient space is used. This measure was normalised to fall within the [0,100] range, like a percentage. Other picture content characterisation vectors are tried and compared with the pixel to pixel results in order to find more efficient image indices.

The tried vectors are:

- histogram
- row and column line integrals: horizontal and vertical (H/V) projections

- 3x3 B/W spatial pattern statistics vector.

The initial idea of the search itself consists of three stages, each stage delivering a candidate set or subset for the next stage.

1. ranking by histogram fraction comparison
2. ranking by H/V projections, the best shift is recorded in case of size differences
3. pixel by pixel distance comparison at the best shift

Each stage has a duration that relates to its complexity. According to Huijsmans and Lew, the first stage takes a constant time. The second and last ranking stage respectively take linear and quadratic time. How this is achieved isn’t clearly mentioned.

Besides the different indices the scale and the intensity pattern is varied to find the best conditions. The scales of 300, 100 and 33 dpi are used. Gray-values, gradient magnitude and binary thresholding gradient magnitude are the different intensity patterns for the search.

As test sets for the initial search the following objects were used: same, faded copy and cut copy with water-colouring. Also same person with and without hat and same studio background were used. Further, several variations in location, rotation, scale and lighting are used during scanning.

The tests resulted in a more efficient, faster search strategy with an equivalent ranking. For the index a multi-resolution representation of the H/V projections is used. For the different scales almost identical rankings were obtained and the resolution can therefore easily be lowered. As intensity patterns the gradient magnitude versions outperform the gray-value comparison. The results of these tests show that the thresholded one is even slightly better in finding faded copies and images with an alike layout.

They compared their results with those of QBIC and VIMSYS, respectively the IBM and the Virage Inc systems for querying by image content. These results strengthen them in their choice for a well-defined application area. For the future they are looking for a strategy for subimage searches. Also they intend to build binary or quad-tree search structures to lower the search time from linear to logarithmic.

4.8 Video Indexing and Retrieval

With the discussion of the article of Smoliar and Zhang [Smoliar], I will only relate to the specific aspects of video. Because video data can be looked at as a set of images, most of its aspects are also dealt with by the work of other authors that bestow this area. This related work is discussed in great depth in relation to their own work.

Smoliar and Zhang relate in their article to video indexing and retrieval. According to them, the effective use of video content is still beyond our grasp. The automatic extraction of semantic information is outside the capability of current signal analysis technologies. Manual classification on the other hand is problematic, because it’s a subjective act.

As stated before, they also base their work on the fact that video material is structured. The task to achieve is to characterise the nature of this structure. This is called parsing by them.

Some basic concepts underlying their work are:

- shot one uninterrupted run of the camera to expose a series of frames.
- frame image unit.
- key frames one of more frames to capture the content of a shot, representative frames.

The representation of a camera shot is by abstracting it to a set of images, the key frames. Still, the content of those images must be represented. One approach is by identifying some characteristic set of content features, such as colour, texture, shape of image objects and sketch (relationships among edges). This is a data-based approach but can also be seen as based on a more general model.

The parsing technique for detecting boundaries between the consecutive camera shots is by the use of histograms of intensity levels. These histogram bins for the temporal segmentation can be in ranges of intensity values but also in intensity ranges of colour components. Unlike two frames from different shots, two consecutive frames of the same shot will show little difference.

Some form of content-parsing, based on domain knowledge, is possible. With e.g. television news, there tends to be a spatial structure between the news reader (called an anchor-person), with his/her name, a news icon and the background. Further a temporal structure can be identified. This consists of news items with the anchor-person shot at its beginning and/or end, possibly interspersed with commercials.

The model-based parsing relies on the classification of each shot according to categories of the model. Categories of television news may include: anchor-person, commercial, weather forecast, news, sports and business shots.

Content-based parsing and developing representation and classification techniques for images and video data form the basis for the design of a suitable indexing scheme. Content-based image retrieval, in the opinion of Smoliar and Zhang, entails reducing the search from a large and unmanageable number of images to a few that a user can quickly browse. Only then can it be effective.

The notion of a feature vector is distinguished by them in relation to the effective retrieval of image data. For image features as those mentioned before, each image can be represented by a feature vector which corresponds to a point in a multidimensional feature space. The image features are computed before the images are added to the database.

For the fast retrieval, two different techniques are described: multidimensional indexing and filtering. Three approaches to multidimensional indexing are currently popular: R-trees, linear quad-trees and grid files. These methods tend to explode geometrically as the number of dimensions increases. With a higher dimensionality, the technique is not better than sequential scanning.

With these higher dimensions filtering can be best used to solve this problem. Filtering acts as a preprocessing step and reduces the search space. Filtering may allow false hits but won’t allow false dismissals.

Queries may be specified by an example image. This can be either selected or created by the user. It may provide information about any of the image features. There are three basic approaches on how this can be accommodated in a user interface: template manipulation, drawing and selection.

Template manipulation uses template maps. A template map is based on the division of an image in a 3x3 array of 9 sub-areas. A query image can be composed from selections from a menu. The resulting assignment serves as a query image.

Unassigned areas serve as "don’t care" specifications.

A natural way for letting a user specify a visual query is by painting or sketching an image. A feature-based query can be performed by extracting the visual features from the painted image. A coarse specification is mostly sufficient, because the query ban be refined based on the retrieval results.

The last approach for specifying a query is by selecting an image from a result set of an earlier query. Visually similar images to the entire example image or a specific region are to be returned.

The retrieval techniques that apply to images can also be performed on video frames (the key frames). Because the user who executed the query is most likely interested in the whole video fragment, the information about that should be available as well.

Some research problems in their work are still open. The audio part of the video has not been taken into account. This may, however, provide additional interesting information to understand video. Audio is a much more unexplored area of research than that of visual information.

They consider the news program material as relatively easy to parse. Developing tools and techniques for modelling different kinds of video material is a major area of research.

4.9 The VIRAGE model

The work of Virage Inc. about Visual Information Retrieval (VIR) is described in a document under authorship of Dr Amarnath Gupta [Gupta]. This work is referred to by several authors in the area of multimedia. Although Gupta acts as a single author, it reflects the work of the whole company team and as such I might refer to them as ’they’.

The document was available through internet as a white paper together with other information about the company and their technology. The web site (*www.virage.com*) also contained an online demo in which they demonstrate the power of VIR. In this section, I’ll first discuss the background of their VIR technology before I give my view on the demo.

4.9.1 The Visual Information Retrieval Technology

The VIR technology meets the need of the market to handle large amounts of visual information. A new model was developed on the basis of extensive academic research in multimedia information system technology. The model is called the Visual Information Management System (VIMSYS) model.

They think that the foremost benefit of their technology is that it gives the user the power to ask a query like 'give me all pictures that *look like this*'. The query is satisfied by the system by comparing the content of the query picture with all the target pictures in the database. They call it Query By Pictorial Example (QBPE). This is just a variation of QBE and thus a form of content-based retrieval.

Unlike traditional database systems in which information is searched by keywords or descriptions in association with the visual information, they claim that this model recognises what the image or video actually contains. According to the author, this is what most users prefer to search visual information by.

In a traditional DBMS an image can also be stored as a BLOB. As mentioned earlier, this datatype is not useful for describing the contents of it. They even consider textual descriptions inadequate, because the same image can be described differently by different people. Instead they consider the use of image analysis technology as the only proper method.

The content of an image or video is extracted with the use of image analysis technology. This extraction results in a very high information compression. The extracted contents represent most of what the user needs in order to search and locate the necessary visual information. These are mainly generic image properties with the following definitions:

■ colour	global colour impression of the image
■ composition	spatial arrangement of colour regions in the image
■ texture	pattern and textural characteristics in an image, like wood grain, granite, marble and clouds
■ structure	general shape characteristics of the objects in an image

The system extracts the contents at the moment that an image is inserted into a Virage database. The size of the extracted information will be in the order of 1 or 2 kilobytes, regardless of the original image size. The extracted information is used for all subsequent database operations. The original image is only used for display.

As mentioned before, an important concept of CBR is to determine how similar two pictures are. Instead of strict or exact matching, relaxed or approximate matching is used. They consider the notion of similarity as appropriate for visual information, because multiple pictures of the same scene may have an identical content although they do not match exactly. In their experience, the overall similarity between two images lies "in the eye of the beholder". As with other methods of CBR, the retrieved pictures are ranked in order of their similarity.

The VIMSYS model consists of four layers of information abstraction. In the order of low to high levelled information these layers are:

1. Image Representation Layer	Raw image (pixels)
2. Image Object Layer	Processed image (primitives)
3. Domain Object Layer	User’s features of interest
4. Domain Event Layer	User’s events of interest for video

The top three layers (2-4) form the content of the visual information.

The Image Object Layer contains the image objects. These are computable generic properties such as colour, texture and shape. These computed features, called primitives, can be computed globally, over an entire image, or locally, over smaller regions of an image. The primitives can be localised in several domains like spatial (arrangement of colour) or frequency (sharp edge fragments) or by statistical methods (random texture).

The primitives are extracted by different computational processes. Several different primitives are necessary to express the content of an image. The search space for an image is therefore multidimensional. The metric to reflect the distance or similarity needs to combine all of these in a composite metric.

How these individual metrics contribute to the composite metric is not fixed. At query-time, a user can change the relative importance of each primitive by adjusting a set of weighing factors and herewith the relative importance of the visual features. This way the visual similarity depends on the context.

The Domain Event Layer models the time-dependent aspects of video and image sequences. Time-dependent features are object motion, object discontinuities, scene breaks, cuts and editing features like dissolves, fades and wipes.

The model is central for the architecture of the Virage technology. Other aspects like keywords associated with an image also play a role. The software architecture that supports the model consists of a core module (Virage Engine) that operates at the Image Object Level of the model. The three main functional parts of the engine, Image Analysis, Image Comparison and Management, are invoked by an application developer on image insertion, image query and image re-query.

Considering the Image Analysis component of the engine as black box, it gets a raw image buffer as input and returns a pointer to a set of data containing the extracted primitive data. The calling application is responsible for storing the returned information.

Within this black box several processing steps take place.

Preprocessing Several preprocessing operations (smoothing and contrast enhancing) are performed on the raw image to make it ready for the different primitive extraction routines.

Primitive extraction Each routine works on the preprocessed image and computes a specific set of data for that routine. A vector of the computed primitive data is stored in a proprietary data structure.

The Comparisons part computes the similarity distance for a pair of primitive vectors. This part is also performed in two steps.

1. Computing of the similarity distance for each primitive (texture, structure, composition and colour).
2. Combining the distance measures with their respective weights. The weights determine the similarity and also if local or global aspects should be emphasised.

The latter results in a final score, by which they are ranked. The pair of primitive vectors and the set of weights are the input for the comparison. A pointer to a structure with the score data is returned.

For a re-query, the same structure can be used to recompute a new score for a different set of weights. Also a threshold value for a score can be supplied. When the distance is greater than the threshold value, it is considered not qualifying. This can result in a significant performance gain.

The Management component contains several supporting functions like initialisation, allocation and de-allocation of weights and scores structures and management of the primitive vector data.

The model is also embedded in the Extensible Virage Engine. This engine is mostly the same as the (Base) Virage Engine, except that it provides the application developer with the flexibility of creating and adding custom-made primitives to the system. Three aspects are important for this type of development.

1. A schema of primitives by which the visual matching mechanism can be application tailored by specification of which primitives should be extracted.
2. The definition of custom primitives and incorporating them in the schema by referencing the ID tag.
3. Support tools for image processing to assist in easy development of new primitives.

The support tools for image processing are supplied as a toolkit. They can perform common operations (contrast normalisation, scaling and cropping) and operations for more advanced features (convolutions, histograms, geometric transformations and masking). Virage also supplies a library of image file format readers and writers for various standard image formats (BMP, GIF, MAC, TIFF, JPEG, PCX, etc.).

When defining a new primitive, custom functions need to be supplied and registered with the system. For this purpose the Extensible Virage Engine contains another functional part; the Register. The new primitive is associated with a primitive ID tag. It can be incorporated in any schema just like a built-in primitive.

The supplied custom functions should be able to:

- compute the data associated with the previously extracted features.
- compute the distance between two sets of extracted feature data.
- if applicable, perform a byte swap of the feature data for Endian management.

- print the values of the primitive for debugging purposes.

For the future Virage wants to achieve full visual information management. To achieve this goal several directions are pursued:

- Applying their technology to video information retrieval (to support the Domain Event Layer of the VIMSYS model).
- Developing a method for parametrically specifiable primitives, a set of domain-specific primitives and a domain-specific mechanism of constructing domain-specific objects using these primitives. This to support the Domain Object Layer of the model.
- Enhancing the query specification mechanism (more expressive queries on the arrangement, more general re-query, more intelligent means of handling image browsing and query-time feature definition).

4.9.2 The Virage Online Demonstration

The demo on the World Wide Web is built with several Virage facilities. Next to the engine, a command line interface (VCLI) and a graphical user interface (GUI), are used resulting in the Visual Intelligence DataBlade. The term datablade is Illustra terminology. Illustra is an object-relational database company. The Visual Intelligence DataBlade is embedded in Illustra’s DataBlade product family. Illustra has been taken over by Informix. Currently Informix has a product called Universal Server that can handle images, sound and video-fragments.

I used the Virage demo several times. In the period May till June 1996 I tried the demo repeatedly. Each time, I only performed a few queries. Unfortunately, the performance of the demo wasn’t very well. It just seemed to ’hang’ and it was unclear to me if this could be fully blamed on the speed of the web, or if there were other causes.

Later, in May 1997, I tried the demo once again. The demo was intuitive and simple to operate. How this is experienced by a novice user is uncertain. At this time the demo performed much better than a year before. This time, it was possible to observe the effect of different queries. The demo was slightly changed compared to the version of a year earlier. In this section I will describe the version of May 1997.

According to the information supplied, the demo database contains over 10.000 images of various subjects and origins with a broad scope. They vary from portraits and scenic views to textures, clip arts and backgrounds. The demo is accompanied by a tutorial that explains how you should operate it.

The demo starts with an initial screen that presents a few randomly selected images. The user can influence the number of images that will be shown on a single screen. A total of 3, 6, 9, 12, or 18 images can be selected. Twelve is the default. These low numbers of returned images have been chosen to give an acceptable performance. In the tutorial they don’t hesitate to explain that the transmitting time causes most of the delay.

By simply clicking on an image, other visually similar images are returned in what is called a results grid. With the option ’random’, another random set of images will be generated.

The first of the returned images is the query image. Under each image is a hyperlink ’info’ for information about the image. The information consists of the image number, the file name, the width and height, the file size and the similarity. The similarity is the distance value between each image and the query image and lies in the range of 0.0 and 100.0. The smaller the value, the more different the image.

The relative importance of the image attributes (colour, composition, texture and structure) can be adjusted. The value representing the relative importance of a feature can vary from 0 to 10. The greater the value, the greater the importance of that feature. By default the relative importance is 10 for the colour image attribute and 5 for the other image attributes.

By setting the relative importance of one image attribute to 10, and the others to zero, a selection based on one single visual characteristic can be performed. I tried this for both structure and texture.

Clicking on a striped pattern results in other striped patterns with different colours. For a structure query the stripes are in the same direction. A horizontal striped pattern results in flags and bricks. For a texture query the stripes in the returned images are in all sorts of directions.

For another type of query I kept the relative importance of the texture attribute on 10 and put colour to 5. This query results in images with similar patterns than the query image, but they have mainly the same colour. Also images with similar colours but with different textures are returned.

For a long period of time I tried to find an image with an airplane. A possibility is that the database doesn’t contain any airplane-pictures. Many random selections never resulted in any airplane-picture. Also selecting similar images based on images with a white flying bird against a blue sky didn’t do the trick.

Eventually, I found the depiction of a double-wing airplane on the ground with a red car in front of it. Whatever combination of visual attributes I tried, I never got another airplane-picture as result. Based on this picture most of what I got, were pictures with cars on it. This isn’t very surprising, given the query image.

I tried all sorts of queries to get some insight in the working of the Virage system. The relationship between the results of a query and the query image in combination with the settings of image attributes isn’t always clear.

For pictures that are selected based on a query image with a clear structure (stripes), shape (triangular traffic sign), or colour composition (flag), it is clear to me why certain pictures are returned. By clicking on such an image, other similar images are retrieved. As far as I can perceive, the images have a similar structure, shape, or globally similar colour percentages in comparison with the query image.

For pictures that are less ordered in their staging, it is much more difficult to understand why certain images are returned.

Also visually similar images can have totally different semantics. One of the queries I performed, had a pink rose in a green surrounding. The query returned other pink and red flowers with green, but also a red car on the grass. Although the images had globally the same visual features, semantically a car is a very different from a flower. With this query the relative importance of composition and structure image attributes were set to 10.

4.10 Query by Image Content (QBIC)

QBIC (Query by Image Content) is an IBM developed technology to index and search pictorial information with. When I decided to investigate multimedia databases for my master thesis, I already knew about the existence of QBIC and knew that this technology would be part of it. IBM responded very cooperatively on my e-mail with the request for information about QBIC and related technology. It resulted in a package that contained six reports and one brochure about the subject.

All papers deal with QBIC. Some discuss QBIC in a general sense [Faloutsos, Flickner, Niblack]. Because the age of the reports differs, the technology has reached a further stage. Others discuss applications that have been developed with QBIC [Holt, Petkovic]. One report and the brochure discuss the Ultimedia Manager, a software product for the management and retrieval of image data. This has also been developed by IBM and combines the QBIC technology with traditional database searches [Brochure, Treat].

In this section, I will combine the information of all these reports. I’ll first describe the QBIC technical background and details. After that, I will discuss the applications in which QBIC is used. I will also, as with Virage, give my personal view of the QBIC demo available through the Web (at *wwwqbic.almaden.ibm.com*).

4.10.1 The QBIC Technology

As with other new developments, the QBIC project is an answer to the need to access multimedia information, in particular images and video data, more naturally. It makes use of their content as an addition to traditional SQL and text queries. The QBIC technique serves as a kind of ’database filter’ by reducing the search space for the user.

The QBIC method is developed for high volume image databases. It also anticipates the expected growth in importance and volume in the near future. More recently QBIC has been extended with video information.

The size of the database must be sufficiently large to justify the use of such a method. With small to medium sized databases one can use fast browsing of thumbnail images to select the required images. Thumbnail images are reduced images of a common and standard size. This will work sufficiently in many cases.

Of high importance, in the process of developing QBIC, is a natural visual query user interface, that allows for query refinement and navigation. A major challenge has been the determination of a suitable set of attributes or features which satisfies the conditions of describing the contents of an image, admitting appropriate similarity measures and forming the basis of an index.

With QBIC, they have carefully distributed the different tasks between human and machine. Tasks that are hard for machines, like identifying objects in a scene and giving a semantic meaning to a scene or an object, are left for the user to handle. On the other hand, computing quantitative features is done best by machines. This has been the guiding principle in the development of QBIC.

Given the state of the art in computer vision and pattern and image understanding technology, they restrict the content to parameters that are feasible to compute. No attempt has yet been made to derive more complex semantic descriptions like "dog", "cat" or "house". Many attributes are available for the description of the image content. An example is the colour histogram, which describes the set of colours in an image.

For each visual feature, representations and associated distance functions have been determined and developed. They have been selected based on their capability to capture the similarity that a human perceives. The properties are selected based on their broad intuitive applicability.

Originally, QBIC was restricted to databases of still images. With images, the two main datatypes are scenes, which are full images, and objects. The latter are subsets of an image. The objects are determined by an outline, a closed contour over an image area. Each scene contains one or more objects. Examples of objects are a person, an animal, a texture area or an apple.

When video information was added to QBIC, the data model needed to be extended to support it. With video information, there are three datatypes, namely shots, scenes and objects. The original still images and video self are also part of the data model.

A video is broken into several parts called clips or shots.

A shot consists of a set of contiguous frames and contains motion objects. For each extracted series of frames, or shot, a single frame is generated. It represents the contents of the whole shot and is called representative frame or *r*-frame. The *r*-frames are handled in the same manner as still images and for that reason they result in scenes and objects. Further processing of a shot also results in motion objects.

QBIC distinguishes two main steps: (1) database population and (2) database query. They have led, next to a set of supporting utilities, to four primary parts: object identification, shot extraction, feature calculation and query interface.

In earlier described models, the first step was split into two separate steps, called database population and feature calculation. Due to a change of view on the matter, they now look at the object identification, shot extraction, feature extraction and feature storage as one step. They all contribute to the population of the database.

In the first step, database population, still images and video are imported into the system. From each still image, a thumbnail image of a common size of 100x100 is generated. If desired, available textual information is added. Object identification is an optional but important part in the process of identifying interesting areas in an image, and results in outlined objects.

The object identification program used to be a manual or semi-automatic method. More ideally they wanted to do this automatically. Unfortunately, the automatic methods for identifying and outlining objects were not sufficiently robust to be implemented in the QBIC product.

More recently, methods of fully automatic and unsupervised segmentation have been successfully used. The method works with a foreground/background model to identify images in a restricted class of images. The images that can be used successfully only have a small number of foreground objects against a generally separable background. It also makes use of the knowledge that objects tend to be in the centre of the picture.

The manual or semi-automatic method is called interactive outlining or shrink-wrapping. Interactively, a user provides some information. This is used by the image analysis methods to compute an object outline. The interactive outlining methods provided are the snakes method and an enhanced flood-fill technique. Internally, the outlined objects result in binary masks for each object.

The snakes method helps the user track object edges. The user draws a coarse initial outline around an object and the tool automatically aligns it with nearby image edges. This is also called shrink-wrapping. The user drawn curve or perimeter lies like a rubber band along object boundaries.

Flood-fill starts from a single object pixel that is selected by the user. The technique repeatedly adds adjacent pixels whose values are within some given threshold of the original pixel. A dynamic threshold can be calculated automatically by clicking on background and object points. For reasonably uniform objects which are sufficiently distinct from the background, it allows for fast object identification. For other images the threshold needs to be manually adjusted.

Feature calculation is the second part of the first step. The computation of image features and attributes is done by a set of batch feature calculation programs. They generate the colour, texture, shape, location and sketch features as image representatives. It is a compute-intensive, but one-time, operation.

Specific for scenes, positional colour/texture and sketch properties are supported. For objects, QBIC supports location and shape features. Colour and texture features are supported for both. According to the model, user-defined properties are also possible to define. Extensibility and flexibility are still key challenges to make QBIC more useful and competitive.

The first part of the database population step for video data is shot extraction. It consists of three major components:

1. shot detection
2. representative frame creation for each shot
3. derivation of a layered representation of coherently moving structures/objects.

Compared to the earlier reviewed report of Smoliar and Zhang [Smoliar], the definition of a shot seems extended. A shot, in the optics of QBIC, is a set of contiguous frames that are grouped together based on coherence. The coherence can be that they:

- depict the same scene

- signify a single camera operation
- contain a distinct event or an action like a significant presence and persistence of an object
- are chosen as a single indexable entity by the user

With QBIC, the concept of a shot is broader than an uninterrupted run of the camera.

They identify two classes for the detection of scene cuts as have been proposed in literature. One class is based on global representations like colour and intensity histograms without any spatial information. The other class is based on measuring differences between spatially registered features.

For QBIC, a method for detection has been developed, that claims to combine the strengths of both classes. They use a robust normalised correlation measure that allows for small motions and they combine it with a histogram distance measure. Results based on a few videos containing 2000 to 5000 frames showed no misses and only a few false cuts.

For the detection of shots caused by the changes of camera operation, an algorithm is developed that computes the dominant global view transformation. The related transformations that result from the algorithm can be used for several purposes like camera operation detection, shot boundary detection based on the camera operation and for creating synthetic *r*-frames. Shot boundaries can also be defined on the basis of events. As with Virage, events form a fundamental concept in the area of video information.

Each shot is represented by the generation of an *r*-frame. The *r*-frame can be any particular frame in a shot. Because no single frame may be representative for the entire shot, they use a synthesized *r*-frame. It is created by seamless warping and combining background parts of all the frames in a shot as a mosaic.

First all background captured in the whole shot is used. It results from the dominant global view transformation algorithm. Any foreground object can be superimposed on this background. The *r*-frames are used for object identification and as representative for the video in the query process.

Different layers within the video information are used to identify significant objects. The objects can be used for feature computation and querying. They call it the layered representation of video. The algorithm devises a shot in a number of layers based on the time-varying nature of the video data. A shot from a flower garden with a flower bed, tree and background can result in three layers that are made visual by different shades of gray.

The second and also last step in the QBIC architecture is performing queries. The query routine handles the creation, navigation and refinement of the query by the user. For each full-scene image, identified image objects, *r*-frame and identified video object, a set of features has been computed to allow content-based queries.

Every feature has its own matching method and similarity function. A control window allows the user to specify which properties are to be used and their relative weights. An associated ’picker’ is provided for each property. A multi-object query is specified by specifying several properties and combining them in a query.

Average colour is used in queries to find images and objects that are similar to a given colour. The average colour is calculated by adding up the red, green and blue components of each pixel. The Mathematical Transform to Munsell (MTM) coordinates in colour space, a 3D vector, is used as metric. The numeric colour values are distributed in a perceptually uniform way. This means that a distance in the colour space results in a colour shift which is expected by the human eye. The picker for specifying this feature consists of three sliders: red, green and blue (RGB 0-255).

Colour distribution is used to find images and objects with a similar distribution of colours. A *k*-element histogram is used (user definable, *k* mostly is 64 or 256). The computation of this feature requires several steps; at the end the histogram is normalised. With the picker, up to five different colours can be specified, and for each specified colour the amount can be specified by adjusting the sliders to the contributing percentage required.

Texture queries can be used by selecting a texture from a sampler. This picker works with a set of pre-stored example images. As underlying features, a 3D vector of the mathematical representations for coarseness (average size), contrast (light and dark) and notion of direction (ordered or random edges) is used.

Shape queries for objects can be specified by drawing a shape in a blackboard drawing area or by picking sample shapes from a palette. Colours can be used in the drawing area. Features that are used to characterise and match shapes are area (size), circularity, eccentricity, major-axis direction (orientation), features derived from objects moments and a set of tangent angles around the object perimeter. It results in a 20-dimensional moment-based vector.

Sketch queries are similar to shape queries. Where shape queries are used for objects, sketching is used for full-scene queries. The query specification is a freehand drawing of the dominant lines and edges in an image. The sketch feature is an automatically extracted reduced resolution ’edge-map’. A template matching technique is used for matching.

Query by example is also a possibility with retrieved images from earlier queries. A thumbnail image can in that manner be used to initiate a query of the form ’Find other images like this one’. The selected images are used as a basis for refining and expanding the search.

Little information was given about querying video data. Of course the video’s *r*-frames can be used as were they still images. The model further indicates that the query interface can also be based on object and camera motion. Except for the example ’Find all shots panning from left to right’, no information is given about how queries of this type are specified.

The kind of attributes used by the queries are typically *k*-element feature vectors. The dimensionality of *k* is often large and leads for indexing methods to ’exponential explosions’. Two different techniques are used for fast searching. One is filtering, which works in two stages. The other technique is multidimensional indexing.

In the first stage of filtering, a computational fast filter is applied to all data. The details about the operation of this fast filter aren’t described. Items that pass through this filter are operated in the second stage, which computes the true similarity metric. This technique is used for high dimensional feature vectors like the 256-dimensional colour histogram.

For features that have a lower dimensionality, such as the aforementioned average colour and texture, multi-key or multidimensional indexing methods can be used. Higher dimensional features, like the 20-dimensional shape feature vector, can be reduced to a feature space of two or three dimensions by a transformation.

With the multidimensional indexing methods, images are returned that are the closest in the feature space. From previous work, they have learned that R^*-trees as underlying indexing method have been the most successful when the dimensionality is not too high.

All queries are approximate queries. The results of these similarity queries are ranked based on the similarity function that is associated with the selected property. Similarity is defined as a distance metric in high-dimensional feature spaces. The results of the query are displayed in order, from the best match to the *n*th best match (*n* can be set by the user).

An image is returned as a thumbnail. A video is returned as *r*-frame thumbnail. The thumbnails function as active menu buttons. By clicking on them, a list of options are displayed. Some of the options are: ’find images like this one’, display the full scene image and so on.

With QBIC, also ’false’ retrievals will occur, because they have e.g. a colour and texture distribution as requested. This is not considered a problem. The human visual system is excellent in quickly focusing on items of interest and discarding the unwanted patterns. There should, however, not be too many of them.

4.10.2 QBIC Applications and Web Demonstration

QBIC has been used in several applications. One of them is the application for retrieving art images of the Davis Department of Art and Art History at the University of California. It is discussed in detail by Holt and Hardwick [Holt] and it is also referenced by Petkovic *et al.* and Treat *et al.* [Petkovic, Treat]. They wanted to determine how effective retrieving art images based on what they look like would be, rather than relying on text indexing. They expect it to be useful for finding art-work where images cannot be accurately described in words.

Their experiment was hindered by the available hardware. Only approximately 1 gigabyte of storage capacity was available. They had to compromise between, obtaining good colour and image resolution on one hand, and achieving maximum storage capacity on the other hand. For this reason they only used images at 8-bits of colour to keep the required storage space per image low.

They found the classification of images by manually outlining and identifying the objects in each scene quite labour intensive. This is however no more time consuming than the use of conventional text systems. An average of 3 or 4 objects were outlined for each image.

Images were selected based on the presence of clearly readable elements, and an additional set was selected that contained obvious visual relationships to the issues of race, class and gender. Also images that contained specific visual elements like fish, horses, hands, reclining female nudes, phallic symbols, drapery, textures, skeletons, text as a component of art-work and perspective were chosen based on the reference questions.

The preliminary results were quite variable. The search based on shapes was problematic. A horse can have several shapes. By looking for a horse shape not all horses will be retrieved. The human eye knows if it’s a horse, although electronically the shapes are different. A text query for this purpose is in their opinion more suitable.

Finding images based on racial characteristics, like skin colour, proved to be more and more accurate. Especially when the search was expanded by using QBE methodologies. Similar results were obtained with class and gender searches. They expect to obtain the most interesting results in combination with text based searches.

The most accurate searches turned up the areas of colour and texture. Simple shapes, like circles and other geometrically shapes are also reasonably accurate. With additional storage capacity (4 gb) they want to use images at 24 bits of colour. They are also refining their methods of image preparation and objects outlining, and of combining content-based searches with text.

Petkovic *et al.* also described other applications that make use of the QBIC technology. The applications are in the areas of stock-photography for remote printing, textile industry and environmental design. Unfortunately their story is too global to get a realistic idea about how QBIC contributes to the results and therefore they won’t be used in my thesis.

The QBIC technology is also incorporated in IBM’s Ultimedia Manager. The brochure mainly contains a commercial overview of the Ultimedia Manager product [Brochure]. The report deals with the combination of traditional database search and content-based search [Treat]. The latter is the QBIC portion of the Ultimedia Manager. The same steps and search methods are treated.

The IBM site on the World Wide Web also provides information about QBIC. A demo version of the QBIC search engine can be run from here. The demo changes on a regular basis. In the period May till June 1996, the QBIC demo suffered from similar performance problem as the Virage demo. In May 1997, I tried it again. The performance was then very well.

I will describe the version of May 1997. It didn’t have a shape or sketch search option. This is very unfortunate, because beforehand this method seemed most promising to me for finding airplane-pictures. With the QBIC demo I tried several queries to observe their effect.

The QBIC demo starts with a screen with 8 thumbnail images. This number can be changed. Between 4 to 8 columns and 1 to 4 rows of images can be displayed. This results in a maximum of 32 images that can be shown on a screen. According to the information the demo catalog contains 1900 images. A maximum of 50 images will be shown as result of any query.

Three different types of similarity measures are supported: colour percentages, colour layout and texture. They are also called methods. It is possible to search based on an exemplar image by selecting one of the shown images, or by customised search. A new set of random images can also be selected.

Several search approaches are distinguished:

- Image similarity select a method, click on an image and leave the keyword field blank.
- Text only type in keywords in entry field and press enter
- URL only type in URL or file name of an image and press enter
- Text & similarity enter keywords, select a method and click on an image
- Customised custom query generation

Text only search was the first approach I used to find airplane-pictures. None of the words airplane, aircraft, flying-machine, airliner, airship, bus, etc. did return any airplane-picture. The word ’bus’ did return some images, but they all had to do with business situations.

By supplying the keyword ’air’, I finally found some airplane-pictures. A set of 24 images satisfying the 'air' keyword was found with the following depictions:

14	airplanes (of which 3 fighter jets)
6	balloons
2	helicopters
1	cockpit of an aircraft
1	person on an air-bed on a white beach

After selecting the 'air'-pictures I tried to influence the order of the images by clicking on the image with the for me most characteristic airplane. I did this in combination with each of the different similarity search methods. This didn't positively improve the order of the images within the set.

After emptying the keyword entry field, I tried the different methods again. The image similarity methods were executed one by one in combination with the same airplane-picture. The texture, colour percentages and colour layout methods all returned a set of 50 images.

With the texture method one other airplane-picture and a picture of a hang-glider or delta wing were returned. The other 47 images seemed to be a random set. With both colour methods 2 pictures of hang-gliders and one with a helicopter were returned. A lot of boats and other objects with a mainly blue surrounding (water and sky) were also returned.

I also tried the customised search approach. This is only available for colour percentages and colour layout. It is not available for texture. For colour layout a colour picker of 16 rows by 16 columns is shown. From this picker a total of 256 different colours can be chosen.

The drawing area is 9 rows by 12 columns. Each little box can be filled with one of the 256 colours by clicking in it after selecting the desired colour. Two fill methods can be used to fill a group of boxes with a few clicks. These are rectangular and block fill.

I painted the whole area blue and made the 4 middle boxes of the middle row light gray. A set of 50 images was returned:

1	airplane
1	helicopter
1	hang-glider
1	bird

and a lot of bluish pictures.

Colour percentages shows the same colour picker. Five different colours can be chosen and for each colour a percentage can be specified. By default the colours are blue, black, red, yellow, and white. The total percentage of all colours can exceed the 100 percent.

A query for 90% blue and 10% white resulted in one hang-glider and one helicopter. They were part of a set of 50 returned images. A lot of bluish pictures were also returned in this set.

Customised search in combination with the keyword ’air’ gives the same 24 hits as before. Only the order is different. The more bluish pictures come first and the more reddish at the end.

I also used the texture similarity search on simple patterns. With a horizontal-striped sand-picture as example, a lot images with a horizontal, vertical, or diagonal striped pattern were retrieved. With a dotted pattern (aspirins), a lot of pictures with people and images with round objects (balloons) were retrieved. Some of the images were retrieved for mysterious reasons.

The interface of the demo is intuitive and simple. However, I am an experienced user and also a computer programmer. I can not compare myself with a novice. How QBIC is experienced by less experienced users is not known by me. The performance of this version was a great relief as compared with a year before.

5 Analysis and Discussion

In the preceding chapter, the research and developments in the area of multimedia databases are reviewed. I selected the respective articles and reports mostly by the different areas they covered. The areas are metadata, modelling, approximate matching, pattern-recognition on shape and texture and content-based retrieval within a specific application area. The chapter was completed with one article about video indexing and retrieval, and with two available operational products.

After the completion of the preceding chapter, it struck me that there are also many similarities between the different sections. Issues that, at first, occurred to me as differences also seem to be resemblances or are based on similar starting-points. There are of course differences between the followed approaches and points of view of the researchers.

In this chapter, I will discuss the striking aspects of all sections. I will also discuss the differences between the sections. Differences can be a dissentient opinion, a different objective, or can be the result of diversity in observations. The overall similarities will also be discussed. Together, this will give an overview of the preceding chapter.

The analysis and discussion is ordered by the different aspects of content-based retrieval on multimedia databases. The different aspects have been distinguished based on their repeating occurrence in the sections of the preceding chapter.

5.1 Metadata

Metadata is one of the building blocks of a multimedia database. As the term metadata indicates, it is information about the data. Within the metadata several abstraction levels can occur.

The section about the work of Kashyap *et al.* is completely dedicated to metadata. The distinction of the different types of metadata in this section is noticeable. Metadata in their view is a broad concept. It both represents information that is independent of the contents and information that depends on the features and properties of an image. The metadata is the middle level of their three layer architecture. Within the metadata layer, no hierarchical distinction is made between different sorts of metadata.

Kashyap’s metadata should be able to model the semantics. In this respect, the term *’semantic gap’* or *’semantic bottleneck’* first occurs here. It is also mentioned by Gudivada *et al.* With this notion they mean the incapability of the state of the art of information technology to represent the semantics of multimedia information. A gap remains between the users conceptualisation and the actual specification.

The *’glue’* or media instance in the section of Marcus and Subrahmanian resembles Kashyap’s metadata. This concept is used as representative of the physical information. It can be used as general purpose access structures and for the definition of indexing structures.

In the works of Gudivada and his co-writers a similar concept can be recognised. It consists of image and image-object logical attributes and structures. The semantic and meta attributes are also part of it. The different forms of metadata, like image and image-object, occur on different hierarchical levels.

The feature vector that is developed by Jagadish for the 2D-rectilinear shapes can be seen as a representative of the physical information and therefore as a form of metadata as well. The same is the case with the feature vector developed by Ma and Manjunath, which is a representative for the textural features. A similar line of reasoning can also be applied on the work of Huijsmans and Lew. Their similarity measure is designed for a specific application area and based on the gray-values of the B/W studio portraits.

Sistla and Prasad also use metadata. It is used for the computation of similarity in their research after approximate matching. The metadata consist of features like colour, size and state of motion, and relationships like spatial associations among objects and actions. How this metadata is obtained is not described, except that it is done manually. It is probably obtained in the same manner as the query is specified.

Zhang and Smoliar distinguish shots, frames and key frames within the video information. For the representation of the key frames from a video several image features are used, like colour, texture, shape and sketch (edges). Also model-based classification methods are possible.

The VIMSYS model in the work of Virage distinguishes primitives, features and events besides the physical visual information. The primitives are generic properties, like colour, composition, texture and structure (shape). They are obtained by several computational processes and are called objects.

The events are not yet implemented in the product. They should consist of the time-dependent aspects of the visual information, like object motion, object discontinuities, scene breaks, cuts, dissolves, fades and wipes. For the future domain-dependent features should also be recognised.

How these representatives are implemented is not described. All these primitives, features and events together form the metadata. User-defined metadata can also be used. Virage’s datablade concept makes it possible to effectively incorporate newly defined primitives in the system. A computation function for the vector data, a distance function and debugging supportdshould be provided for the new primitive. In the architecture the possibility of keywords is also briefly mentioned.

For QBIC the metadata consists of several visual features. Textual information and thumbnail images are also used. QBIC distinguishes images, image objects, video, shots, video *r*-frames and motion objects. For the images, *r*-frames and objects, several features like colour, texture, shape, location and sketch are computed by a set of feature calculation programs. These are mainly generic properties that are not specific for a certain domain.

Shots in a video can be detected by an algorithm. Objects in an image or an *r*-frame can be identified. This usually happens in an interactive manner. Fully automatic and unsupervised segmentation is also possible for images with objects that are sufficiently separable from their background.

As with Virage all the objects, features and other information that is not the original image or video can be seen as metadata. User-defined properties are also possible. However, it isn’t clear how they should be implemented.

With QBIC, Virage and the work of Smoliar and Zhang not all metadata occurs on the same level of abstraction. This can be gathered from the aforementioned different types of metadata, e.g. object and shape.

As can be concluded from this section, various types of metadata can be distinguished. Metadata can be textual but can also be a feature vector for a visual property. It can either be dependent or independent from the contents. And a distinction can be made between generic and domain-dependent features. Further, user-defined features also occur. The types of metadata occur on different levels of abstraction.

As follows from the reviewed work, metadata is used for efficient querying and retrieval of multimedia information. Content-based retrieval cannot really be performed without it.

5.2 Data-driven versus Model-driven Approach

In the process of obtaining the metadata, a distinction is made between a bottom up or data-driven and a top down or model-driven approach. Kashyap *et al.* consider the domain-dependent, content-descriptive and media-independent metadata to support the semantic correlation the best. They follow the model-driven approach.

Gudivada *et al.* also follow the model-driven approach. Although some attributes can be derived without domain expertise, most of the representatives are derived with considerable human involvement. The elicitation of the domain-dependent semantic attributes can be supported by the Personal Construct Theory. It is based on the assumption that people use constructs in the process of decision making. It is a valuable technique when domain-dependent attributes are to be used.

Huijsmans and Lew more or less agree with them. Their conviction is that efficient retrieval can only be done for specific application areas. Researchers that have more faith in the model-driven approach consider the state of the art in information technology as insufficient for the data-driven approach to support the semantic correlation. Many research challenges are to be met to overcome this.

Smoliar and Zhang have the same conviction in relation to the content of video. They consider the automatic extraction of semantic information as beyond our grasp. They also consider the manual classification as problematic, because subjectivity will be introduced in this process. However, some form of content-parsing, based on domain knowledge, is possible. It relies on the classification of each shot according to the model.

Smoliar and Zhang also support the data-driven approach. Model-based parsing functions very well when certain shots need to be identified in a video. For the representation and classification of the content of single frames, data-driven techniques are used. They look on them as features of a more general model.

Marcus and Subrahmanian, on the other hand, think that an approach independent from a specific application domain is required for the large-scale development of multimedia systems. Nevertheless, their work is fully theoretical and they haven't bothered themselves with the question of how the information should be stored physically or can be represented conceptually.

Sistla and Prasad use manually created metadata for the computation of similarity in their work related to approximate matching. The metadata should be generated a-priori with the use of image processing algorithms, manually or with a combination of both. They expect that computer vision and pattern recognition techniques will help in automatically identifying some of the objects and the relationships.

Jagadish explicitly uses the data-driven approach within his shape-recognition research. With the data-driven approach promoted by him, an index structure is constructed over the data. It allows for the retrieval of similar shapes in less than linear time. While for the model-driven approach, a given shape needs to be compared individually with each shape in the database.

The approach of Ma and Manjunath is mainly data-driven. Although domain-specific knowledge may be necessary in some cases of their method, for textures randomly selected locations suffice.

The VIR technology of Virage also seems to be data-driven on the first impression. Virage mainly uses generic properties for the features. The contents of an image are extracted and calculated on image-insertion time.

By carefully comparing their approach with the definitions of data-driven and model-driven, it seems it also has model-driven aspects. When a query is performed, they compare the query image or features with the features of all the target images in the database. The comparison part functions in such a way that it computes the similarity distance between a pair of primitive vectors.

QBIC mainly has a data-driven approach. Generic properties are used to capture the visual information in an image or video. Object and shots can be detected automatically, like the expectation of Sistla and Prasad. In most cases, however, an interactive method is used for the identification of objects in a picture or video frame. All processing of this kind and feature calculation happens on image- or video-insertion time.

As is also expressed by Smoliar and Zhang, the interactive identification of objects implies a certain level of subjectivity. One user might decide to identify the objects *A*, *B* and *C*, while another will choose *A*, *B* and *D*.

Each feature is represented by an appropriate feature vector. Depending on the dimensionality of the vector, two techniques can be used for searching. One is multidimensional indexing and the other is filtering. The latter is used for high dimensional feature vectors, because the current indexing techniques can’t handle them.

As can be concluded from this section, in most cases the identification of objects in an image is an interactive process. It is performed on image-insertion time. When objects are identified by users, this is considered a domain-dependent approach. On the other hand, when indexing is applied, like by the work of Jagadish, this is considered a data-driven approach.

The data-driven and model-driven approaches can be used in combination. This is not a contradiction. One approach, or a combination of approaches are used for the generation of metadata, and another or again a combination can be used for the retrieval. Considering the state of the art of information technology, a partially model-driven approach is needed for an effective retrieval based on the content.

For the generation of semantic metadata, the involvement of users cannot be omitted. This cannot be avoided, although subjectivity will be introduced by it.

5.3 Model and Architecture

For visual information, such as images and video, an architecture or model of several layers is repeatedly distinguished. The layers are needed to distinguish the physical information from the metadata.

A three-layer architecture is described in the work of Kashyap *et al.* The three layers from top to bottom are: ontology, metadata and (raw) data. The AIR architecture that is developed by Gudivada and his co-writers also consists of three layers. It results from their observations of the AIR data model that is part of their extensive framework for an image retrieval model. The three layers of the AIR architecture are the semantic, logical and physical level representation. In both of their researches, the physical data is part of the architecture.

By studying the work of Marcus and Subrahmanian, it is possible to distinguish three layers as well. The lowest is of course the physical data. Above that is the *'glue'* or media instance. This can be seen as logical representation. The frames are based on the media instances and can therefore be seen as a layer of information on top of them.

No specific layered architecture is distinguished by Sistla and Prasad. They consider the well-known ER diagrams as appropriate for modelling images. The ER diagrams only model the objects, the object characteristics and the relationships among the objects. The physical data isn't modelled by it.

The feature vectors from Jagadish, Ma and Manjunath and Huijsmans and Lew have formerly been interpreted as metadata. For that reason they can also be considered as a layer on top of the original raw data.

Zhang and Smoliar also do not describe a model or architecture in their work. The basic concepts of their work are shots, frames and key frames. These are parts of a video, single images in a video or shot, and respectively representatives for a whole shot. Within the video's key frames, several generic image features are recognised. The video basic concepts and the generic features together can be seen as a basis for a model.

In the work of Gupta, a model is clearly defined. The VIMSYS model consists of 4 layers of information abstraction. The first level contains the original image and is called the Image Representation Layer. The subsequent other layers capture the content of the image or video. They are the Image Object Layer, the Domain Object Layer and the Domain Event Layer. They are ordered from low to high levelled information. The levels respectively contain the primitives, user-features and user-events (video).

Virage distinguishes both a model and an architecture. The model is central to the architecture. Within the architecture other aspects, like keywords associated with images, play a role. The architecture consists of an engine and a number of functional parts. Depending on the type of engine, these are for analysis, comparison, scores, weights and registering. All layers of the VIMSYS model are not yet fully implemented in the architecture.

An architecture is clearly defined for the QBIC technology. It is also referred to as the model. They have more or less combined the visualization of the model and architecture in one picture. In earlier work, only image data was modelled. In more recent work video information is also modelled.

Two main datatypes are recognised for images, namely scenes and objects. For video this has been extended with shots. The image datatypes are also relevant for video information. The still image and video images are also part of the model. There are two different types of objects: image objects, which are subsets of an image, and motion objects. The latter are specific for video. Generic and user-defined features are determined for the various datatypes.

QBIC’s architecture consist of two parts. The first part, database population, deals, as the name indicates, with the population of the database. The other part, query, deals with the retrieval of information from the database. The last cannot function without the first.

The database population part comprises of the object identification, shot extraction and feature extraction components. The query part consists of the query interface, the match engine, the filtering/indexing part and a component for result ranking.

The metadata or representatives are sometimes separated from the original raw data and put into a separate database. The metadata can also be part of the same database. Another approach is that for each type of multimedia information a separate database is used.

As can be seen, a model is often recognised in the reviewed works. It can also be put together based on the information given, if no model is explicitly described. A model should at least contain the physical information and a meta level of information on top of it. As has been mentioned before, several levels of abstraction are recognised within the metadata.

Datatypes or aspects that have been modelled in the previously reviewed work are:

- scene or frame
- state (video frame)
- object or entity
- event (video)
- motion object
- relationship or association
- feature, attribute or primitive

The architectures are based on the underlying model. The architecture describes the functional parts it comprises of. Often features or other modelled components are generated automatically. The architecture should contain the functional parts for this.

5.4 Indexing and Matching

The size of the database causes the necessity of indexing structures. These structures are based on a data-driven approach. Based on the data, a feature vector needs to be generated, possibly with some human involvement. Although it is possible to develop a feature vector for a specific domain, this approach is still called data-driven.

The notion of a feature vector is of great importance in the area of content-based retrieval. It always points to a specific point of the feature space. This way indexing techniques can be used. In a query all images with feature vectors that point in approximately the same part of the feature space, compared to the feature vector of the query image, are to be retrieved.

Indexing is not part of the work of Kashyap *et al.* and Gudivada *et al.* Marcus and Subrahmanian, on the other hand, deal with indexing in their theoretical work. Their indexing structures allow for relaxation, thus more than exact match is supported. They have shown in their work that the defined indexing structures are relatively efficient. What this will mean for a practical situation is not known.

Jagadish does address the question of how to construct an indexing structure. According to their point of view, such a structure enables efficient retrieval by similarity. For the construction, a data-driven approach needs to be used. Jagadish’s approach is applied in a specific, however unrealistic, application area of 2D-rectilinear shapes.

Jagadish elaborately describes which components the necessary feature vector that represents such shapes is composed of. By omitting certain factors of the feature vector, several kinds of queries can be performed. This approach also allows for the retrieval of shapes that are not identical. Approximate matching is also possible by allowing error margins on the factors of the feature vector.

Sistla and Prasad use metadata for the computation of similarity in their research after approximate matching. They use a function that computes the similarity between the query image and each image in the database: the greater the similarity, the higher the similarity value. Images can be ranked based on this similarity value. They haven’t applied any indexing techniques yet. The kind of similarity they use doesn’t make their approach easily adaptable for indexing.

In the work of Ma and Manjunath, the extraction of image features for texture-based pattern recognition are treated. The steps for transforming an image into a feature vector are described. The current developments in computer vision can handle the detection of low level features and give a basis for the recognition of textures. Currently, indexing is not implemented in their work. Instead, they use sequential search to find similar textures.

Indexing is treated by Huijsmans and Lew in their report about content-based image retrieval. They also use a measure of difference for the ranking of the results. The kind of indexing they describe is not what I thought was meant with this term. The indexing is applied on the precalculated ranking results of all images against every other image.

Smoliar and Zhang distinguish both filtering and multidimensional indexing as retrieval techniques. Although several approaches of multidimensional indexing are described, it can only effectively be used on feature vectors with a sufficient low dimensionality. With a higher dimensionality, the technique is not better than sequential scanning. With higher dimensions, filtering can be used. It acts as a preprocessing step by reducing the search space.

Although the notion of a primitive vector seems to point in that direction, no real indexing is supported by Virage. The mentioning of a multidimensional search space does suggest a form of indexing, but this is not the case.

With Virage, each image item in the database needs to be compared with the query item for a query. This results in one or more similarity or distance metrics. These are combined in a composite metric. The results are ranked based on this value.

With this approach, it is not only items that are identical and have a distance of zero are retrieved. Approximate or relaxed matching is thus supported. Virage considers this to be an important concept of CBR and appropriate for this type of information.

The relaxation can be controlled by applying weighing factors for the different measures. Also, a threshold value for the distance can be used to discard the items that differ too much as not qualifying. This is similar to the error values of Jagadish. Virage's approach suggests a form of filtering.

QBIC sees the need for techniques that speed up queries. Although sequential scanning of features followed by a straightforward similarity computation can be adequate for a small database, this can be too slow for a growing database. QBIC therefore supports both filtering and multidimensional indexing for fast searching.

Filtering works in two stages: a filter step, and the computation of the true similarity. Unfortunately, the details of the fast filter operation are not described. With indexing, the value of the distance metric in feature space determines the similarity. The number of best matching images can be specified by the user. False retrievals can be discarded manually from the returned set of images.

6 Conclusion and Proposed Model

In this chapter I will verify the objectives of this thesis. I have described them in the second chapter. My principle objective was to find out if an answer to the question *'Give me all pictures that represent airplanes?'* can be given. As can be concluded from the preceding chapters, a straightforward answer to this question is not simple to give.

In this chapter, I will also propose how a multimedia system should look, based on my findings. This will result in a model. For the model, I will take the state of the art of information technology into account. I will also try to describe how in my opinion the ideal multimedia system should look, regardless of the technical possibilities.

Although a multimedia system consists of more than just images, I will treat it as if multimedia information only consists of this type of data. Video information will also be mentioned when relevant. This choice is based on the available information about images and the lack of information about the other types.

6.1 Conclusion

There are several possible strategies that can be followed for solving the problem of the 'airplane'-query. The possible ways have been distinguished and are described as five retrieval classes for image databases [Gudivada96]. They can also be summarised as three different search methods: 1. by browsing; 2. with keywords and 3. on the basis of the content. I will deal with these different approaches individually.

- *Browsing*

Retrieval by browsing is the simplest method. As with all other methods, the system should at least have provisions for the storage and retrieval of images. This can be done in the form of the BLOB. Nowadays, the BLOB has almost become a standard feature in most DBMS.

Browsing can be used on the original images and on the so called thumbnails. The disadvantage of the use of the complete images is that only a few of them can be viewed on a single screen. Further, the formats of the individual images can vary a great deal. The details of the scene are kept when the full images are used. The generation of a thumbnail image from the original image could be a database feature, but can also be done externally.

The selection of the pictures with airplanes will be the full responsibility of the user. Which images need to be selected can be a problem. Not all airplanes will look the same. As well as modern passenger airplanes, there are all kinds of airplanes. Also pictures containing objects with depictions of airplanes in them can be considered to be airplane-pictures.

If I were the person searching for pictures with airplanes, I would probably have a good idea of what kind of airplane-pictures I am looking for. If I asked someone else to select the airplane-pictures for me, this would probably result in a different set of pictures.

Browsing through large databases and images with small details is a labour intensive process and can lead to an unworkable situation. In modern times this is no longer acceptable, because the salary costs are often the highest costs involved. Browsing as a search method cannot be considered to be a real option.

- Keywords

Another of the possible approaches is by the use of keywords. This method is also called retrieval by objective attributes. As stated before, many disadvantages are associated with this method. The main disadvantages are the occurrence of synonyms, defining uniform textual descriptions and the possibility for approximate matching.

The occurrence of synonyms is a reality. Even for the word *'airplane'*, there are many words that have the same or a similar meaning. Examples of these synonyms are flying-machine, plane, aircraft, airliner, airship and bus. Some of these words also have another meaning than airplane, like for plane and bus. There are also airplanes for specific purposes, like bomber, glider and sailplane, and airplanes of a specific make, like Boeing, Concorde and Fokker.

When allowing users to enter all kinds of textual information, the occurrence of synonyms will increase, and seriously hinder the chance of an effective search. The generation of a keyword interface for an application area should be a one-time operation. This depends on the stability of the application area and the model.

When the model is subject to continuous changes, it may need to be adjusted as well. This can also imply that the interface is subject to change as well. More importantly, keywords that have been determined and stored previously need to be changed as well. With a large database, this will be a time-consuming operation. Therefore, when the keyword interface is specified, only the stable aspects should be modelled. In a continuously and rapidly changing environment the use of this method is discouraged.

The disadvantage of defining uniform textual descriptions can more or less be solved. Although no commonly accepted vocabulary for describing features like colour, texture and shape is at our disposal, for a specific application area this can be defined. The Personal Construct Theory can be a helpful mechanism in this sense. It will also solve the occurrence of synonyms.

Colours can be defined and specified by creating colour-cards for the necessary colour ranges. Texture-cards with samples can be used to distinguish ’dotted’ from ’spotted’, ’ribbed’ from ’striped’ and possibly also ’wavy’ from ’curly’. Behind these colour and texture-cards, there will be unique keywords that form a code or description for each of the colours or patterns.

Even with the use of these cards, there will be colours and textures that are borderline cases. Even experienced users of these systems will mistake one for the other. The description of shapes is a much greater problem. Based on keywords, the description of complex shapes will be an impossible task.

The mentioned solutions will result in an interface with a limited vocabulary only. With this interface, only predefined keywords, colours and textures can be selected to search with. Disadvantages remain with this solution. This mechanism is not feasible for general applications and for unanticipated searches. Searching without the use of such an interface still involves the earlier mentioned problems.

The disadvantage of exact matching only cannot be circumvented with this solution. Finding representations that look like an airplane cannot be performed with this method. To make this possible, such a system could be extended with a form of associative search. This type of search will be by the keywords associated with the selected keywords.

For an airplane, these associated keywords can be airport, flying, propeller, sky, wing, etc. A good name for such an approach in the line of the retrieval classes of Gudivada, will be Retrieval *by Associated Keywords*. This type is merely a variation of retrieval by objective attributes.

Of course, a system based on textual keywords has advantages as well. For the retrieval of pictures that represent airplanes, the presence of a uniformly defined keyword as *'airplane'* will be the shortest cut in getting all of them.

The main advantage of the use of keywords is that traditional databases are very well suited for this kind of data. All search data is alphanumeric. Also, most DBMS have standard provisions for storing raw data as BLOBs. For specific application areas, the use of keywords will be a valuable method, whether it is an extension on other methods or not.

- Content

The last of the approaches I will deal with is retrieval on the basis of content. My view on content-based retrieval is one single approach, although Gudivada has distinguished three different retrieval classes within it. This approach has had the most attention in the preceding chapters. This is because my interest has been focused on it. It is also the area where most of the new developments will occur in the future.

For content-based retrieval, several approaches have been discussed. Some of the reviewed systems have been created for specific application areas and function reasonably good for that. Other systems, like Virage and QBIC, are not developed for a specific application. They should perform reasonably well for all kinds of applications and also for general applications. None of these systems function well enough to be suitable for all possible application areas.

As many authors have mentioned before, the state of the art of information technology has not yet reached the level of maturity that means effective content-based retrieval can be achieved. More especially, giving a semantic meaning to objects, that can be distinguished in an image or for the image as a whole, is an almost impossible task. This remains a very difficult task to achieve.

The systems that have been reviewed are in general good at distinguishing the low level features, like colour and texture. The capturing of low level features is the main strength of such a system. This is also a weak point of multimedia systems based on keywords.

Other features, like shape, are harder to recognise. A shape is represented by the outlined contour of an object in an image. The outlining often requires considerable human involvement and is therefore an expensive and subjective task. A lot can be concluded from a specific shape, although a certain shape is mostly insufficiently discriminating for what an image or object comprises of in essence and meaning. Even in combination with the colour and texture information of the object and the surrounding area, this often cannot be achieved either.

For the pictures that represent airplanes, a content-based retrieval query will possibly result in silver "shapes with wings" in a gray and blue surrounding. Of course, not all airplanes are silver and an airplane is also an airplane when it is stationed on the ground.

Even when it is turned into a restaurant or has its wings removed, it will remain an airplane for most observers. To give this meaning to it by a computer program is still an impossible task. A kite in the sky with the shape of an airplane will be considered an airplane for some people. Other persons will only see a kite in it.

With content-based retrieval, how the query is specified will largely determine what kind of airplane - and other pictures - will be retrieved. The query for a silver airplane-shaped object in a half blue and half gray surrounding will also return a lot of seagulls, swans and other birds. Getting all airplane-pictures on the basis of a single content-based query is an impossible task.

A possible extension on shape queries is by the distinction of the objects within objects. They will provide information that can have a surplus value for the identification. Next to giving a general impression of the depiction, the outlined objects and sub-objects should emphasise the domain specific visual aspects. These sub-objects are merely objects that have a specific spatial relation with another object.

With the additional information of the sub-objects, the discriminating ability will grow. Even with this extension, getting all airplane-pictures on the basis of a single content-based query remains an impossible task.

Because systems based on content-based retrieval are still in their infancy, for the future, a lot can be expected of them before these systems will reach a mature stage. With all the attention that currently is given to multimedia and more specifically internet, many accomplishments will be made in the near future.

A general system for recognising all sorts of animals, buildings, airplanes and other objects cannot be created based on the current maturity level of the technology. A system for recognising airplanes only is a more likely possibility. Such a system will work as a classification or identification key, like an identification list in a flora for plants. Only specific airplanes will be retrieved based on the supplied information.

Even with a specific system that works with domain dependent features, recognising an airplane in a picture may prove to be difficult. Pictures might be of a bad quality, for example by over- or under-exposure, insufficient contrast or coarseness of the film. Further, the scanning conditions might influence to what extent an object is classifiable.

Even when the quality of an image is in optima forma, the staging can be the obstacle for identification. The distance between the camera and the object can be wrong, because the airplane is a spot in the sky or only a small part of the airplane is in the picture. Also, the angle under which the object has been photographed may not show a sufficient number of identification attributes.

6.2 Model

As I stated before, I will limit the proposed model to images only. Many researchers agree in their work on what should be modelled to describe an image. After studying their work and considering my own ideas, I can only agree with them, that an image can be described by its objects, features and the relationships among them. With video other aspects, like a shot and an event, also may play a role besides the modelled parts of the image.

This model leaves much flexibility to its implementation in a multimedia system. The model is both suited for general systems and for specific applications as well. Which objects and features are to be used and how these are represented, will determine a lot of the possibilities of the multimedia system.

When considering the state of the art of information technology, the possibilities of existing systems like Virage and QBIC and the idea behind my principal objective, a multimedia system can be described that combines the advantages of other system and that will also support the retrieval of airplane-pictures.

Both QBIC and Virage can be seen as basic multimedia systems. The word multimedia is overly descriptive for systems that only support visual information. For the description of the proposed system, I will take QBIC as a starting-point. Only the aspects that need to be different and the additional functional parts need to be described. Further, functional parts that are alike and that I consider to be important will be emphasised.

As has been concluded from the previous section, a multimedia system cannot be based on content characteristics alone. To get all airplane-pictures from a multimedia database, a textual keyword containing *'airplane'* should be present with each airplane-picture. With this all of them can easily be retrieved. QBIC does already integrate such textual information with the other features. Although this is not the way I would like it to be, the maturity of the technology doesn't allow for a fully content-based retrieval.

As described in the previous section, finding pictures with depictions that are somehow related to airplanes can be made possible by extending the system with the so called retrieval by associated keywords. For the current airplane-query this type isn't relevant. But expanding the retrieved set of images from 'airplanes' to 'airports' is possible by allowing one association step in the query. A general mechanism for associated textual search will make a multimedia system more powerful.

An interface that gives the opportunity for defining textual keywords as an extension will give extensive possibilities for adding specific information about the image. At least a general interface should be provided. The general interface should support the specification, storage and query of general properties of images, like format, resolution and date. These properties can also be part of the catalog.

A reserved area for free form information should also be provided. This is also the case with QBIC. In this area, specific information about the image and its objects and features can be entered. For airplane-pictures, this information can be *'airplane, flying, passenger, silver-blue, KLM'*.

An interface that is specific for the application area is also recommended, because this can capture the model-dependent attributes. Such a user definable interface for textual descriptions of multimedia items is a welcome addition, if not a mandatory one. Possibly, a system based on the Personal Construct Theory should be available to generate a user defined textual interface.

The support for this kind of interface is not yet an integral part of any multimedia system. Of course, it is possible to build it with the help of a separate product. By having this support as a functional part of the multimedia system, the user defined interface can be automatically linked with the other generic feature 'pickers'.

For the retrieval of certain airplane-pictures, the content-based approach can play a role. As mentioned earlier, the choice of objects and features leaves a lot of freedom for the possibilities of a multimedia system. The choice of the combination of these largely determines the possibilities for the identification of an image.

Pictures of airplanes with a specific colour, staging or perspective are much easier to select with a content-based query. By combining a shape with wings, drawn under a certain angle, specifying its colour to be silver and blue and specifying the colour of the surrounding area to be blue with white and gray spots, only specific airplane-pictures will be retrieved. Some false retrievals might also occur with this query. They can easily be dropped from the set. These kind of features and objects are also supported by QBIC.

As described in the previous section, details of an object that emphasise the domain specific visual aspects can be outlined as well. QBIC does already support this, because these sub-objects are merely objects that have a specific spatial relation with another object. This approach enlarges the discriminating ability but is also a labour-intensive task.

All information about the images, whether they are keywords, objects or features should have a representation that can be indexed or should at least allow for fast filtering. For the textual information the traditional indexing techniques still apply. The representations of the objects and features should preferably be real feature vectors of a low dimensionality.

The ideal multimedia system should be able to handle the combination of those features by which the meaning of the images or its objects can be captured and understood. This is what the fifth retrieval class, described by Gudivada, contemplates.

The current generic features that can be determined for images, do not cover a sufficient amount of the image characteristics. Therefore, a full interpretation of the image depiction cannot be given. Additional generic, or even domain specific, features need to be developed to achieve this.

Another approach that might improve the understanding of the depiction, is to add knowledge about the visual world or a specific application area. The multimedia systems of the future should have some sort of knowledge base to be able to successfully interpret and recognise the depiction of an image.

Having a multimedia system that is effective is one thing. In order to have it widely accepted, the interface should be intuitive and the performance should be acceptable. As mentioned in chapter 3, in which several concepts were discussed and defined uniformly, a natural interface has been widely seen as an integral part of a multimedia system.

Of course, the performance of a multimedia system is part of such a human-like interaction. The responses of a multimedia system should resemble the natural human conversation, in character as well as in time. With these aspects in mind, future accomplishments are eagerly anticipated.

Literature

[Blanken] Henk Blanken, Peter Apers
Speciale database-toepassingen, ontwikkeling en toekomst.
Informatie, Kluwer Deventer, januari 1996, jaargang 38, p18-24.

[Brochure] Ultimedia Manager 1.1 and Client Search - Find images by color, shape, texture, and related business data.
Brochure from International Business Machines Corporation, 1994, 2 pages.

[Cheyney] Matthew Cheyney, Peter Gloor, Donald B. Johnson, Fillia Makedon, James Matthews, Panagiotis Takis Metaxas
Toward Multimedia.
Communications of the ACM, January 1996, Vol. 39, No. 1, p50-59.

[Colton94] Malcolm Colton
Illustra, Relational Databases and Spatial Data.
Illustra White Paper, November 1994.

[Colton95] Malcolm Colton
Multimedia Asset Management White Paper.
Illustra White Paper, 1995.

[Faloutsos] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, R. Barber
Efficient and Effective Querying by Image Content.
IBM Research Report: Computer Science, RJ 9453 (83074), August 3, 1993.

[Flickner] Myron Flickner, Harpreet Sawhney, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, Peter Yanker
Query by Image and Video Content: The QBIC System.
Computer, volume 28, Number 9, September 1995, p23-32.

[Gudivada94] Venkat N. Gudivada, Vijay V. Raghavan
Picture Retrieval Systems: A Unified Perspective and Research Issues.
Technical Report TR-19943, Ohio University, Department of Computer Science, Athens OH, 1994.

[Gudivada96] Venkat N. Gudivada, Vijay V. Raghavan, Kanonluk Vanapipat
A Unified Approach to Data Modelling and Retrieval for a Class of Image Database Applications.
In: Multimedia Database Systems - Issues and Research Directions, Springer Berlin Heidelberg, 1996, p37-78.

[Gupta] Dr. Amarnath Gupta
Visual Information Retrieval Technology A VIRAGE Perspective.
White Paper, Revision 3, 1995.

[Gurchom] Manfred van Gurchom, Erwin van Rijssen
De waarde van multimedia-toepassing.
Informatie, Kluwer Deventer, juli/augustus 1996, jaargang 37, nr 7/8, p452-460.

[Holt] Bonnie Holt, Laura Hardwick
Retrieving art images by image content: the UC Davis QBIC project.
In: Aslib Proceedings, vol.10, n.10, October 1994, p243-248.

[Hoogeveen] Martijn Hoogeveen
Een introductie in multimedia: De evolutie van stand-alone naar genetwerkte publieke multimedia-systemen.
Informatie, Kluwer Deventer, juli/augustus 1996, jaargang 37, nr 7/8, p438-442.

[Huijsmans] D.P. Huijsmans, M.S. Lew
Efficient Contend-based Image Retrieval in digital Picture Collections using projections: (Near)-Copy location.
Proceedings 13th International Conference on Pattern Recognition, Wenen 1996, Volume III, p104-108.

[Jagadish] H.V. Jagadish
Indexing for Retrieval by Similarity.
In: Multimedia Database Systems - Issues and Research Directions, Springer Berlin Heidelberg, 1996, p165-184.

[Jansen] René M. Jansen, Ester Koster
Multi-mania en de Information Hype-way.
Informatie, Kluwer Deventer, juli/augustus 1996, jaargang 37, nr 7/8, p480-489.

[Kashyap] Vipul Kashyap, Kshitij Shah, Amit Sheth
Metadata for Building the MultiMedia Patch Quilt.
In: Multimedia Database Systems - Issues and Research Directions, Springer Berlin Heidelberg, 1996, p297-319.

[Kay] M.H. Kay, Y. Izumida
Object-databases binnen multimedia systemen.
Informatie, Kluwer Deventer, februari 1996, jaargang 38, p32-36.

[Ma] W.Y. Ma, B.S. Manjunath
Texture-Based Pattern Retrieval from Image Databases.
In: Multimedia Tools and Applications, Kluwer Academic Publishers, 1996, Volume 2, p35-51.

[Marcus] Sherry Marcus and V.S. Subrahmanian
Towards a Theorie of Multimedia Database Systems.
In: Multimedia Database Systems - Issues and Research Directions, Springer Berlin Heidelberg, 1996, p1-35.

[Niblack] Wayne Niblack, Ron Barber, Will Equitz, Myron Flickner, Eduardo Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos
The QBIC Project: Querying Images by Content Using Color, Texture and Shape.
IBM Research Report: Computer Science, RJ 9203 (81511), February 1, 1993.

[Noordzij] Arie Noordzij
Internet en multimedia: een terreinverkenning.
Informatie, Kluwer Deventer, juli/augustus 1996, jaargang 37, nr 7/8, p443-451.

[Petkovic] Dragutin Petkovic, Wayne Niblack, Myron Flickner, David Steele, Denis Lee, John Yin, James Hafner, Frank Tung, Harold Treat, Richard Dow, May Gee, Mimi Vo, Peter Vo, Bonnie Holt, Janet Hethorn, Kenneth Weiss Peter Elliott, Colin Bird
Recent Applications of IBM’s Query by Image Content (QBIC).
IBM Research Report: Computer Science, RJ 10006 (89095), January 23, 1996.

[Silberschatz] Avi Silberschatz, Mike Stonebraker, Jeff Ullman (editors)
Database Research: Achievements and Opportunities Into the 21st Century.
Report of an NSF Workshop on the Future of Database Systems Research, May 26-27, 1995.

[Sistla] A. Prasad Sistla, Clement Yu
Retrieval of Pictures Using Approximate Matching.
In: Multimedia Database Systems - Issues and Research Directions, Springer Berlin Heidelberg, 1996, p101-112.

[Smoliar] Stephen W. Smoliar, HongJiang Zhang
Video Indexing and Retrieval.
In: Multimedia Systems and Techniques, Kluwer Academic Publishers, 1996, p293-322.

[Suijker] John Suijker
Inside the IBM Digital Library.
Master’s Thesis, 16 march 1996.

[Treat] Harold Treat, Ed Ort, Jean Ho, Mimi Vo, Jing-Song Jang, Laura Hall, Frank Tung, Dragutin Petkovic
Searching Images Using Ultimedia Manager.
Report IBM Santa Teresa Lab. (ca 1994)

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AIR	Adaptive Image Retrieval
B/W	Black and White
BLOB	Binary Large OBject
C/S	Client/Server
CBR	Content-Based Retrieval
CBT	Computer Based Training
dpi	dots per inch
GIS	Geographical Information System
GUI	Graphical User Interface
H/V	Horizontal and Vertical
IDM	Image Data Model
ILR	Image Logical Representation
IRM	Image Retrieval Model
MBR	Minimum Bounding Rectangle
MTM	Mathematical Transform to Munsell
OLR	Image Object Logical Representation
OODB	Object Oriented DataBases
PCT	Personal Construct Theory
QBE	Query By Example
QBPE	Query By Pictorial Example
QBIC	Query by Image Content
RAD	Rapid Application Development
RBR	Retrieval by BRowsing
r-frame	Representative frame
ROA	Retrieval by Objective Attributes
RSA	Retrieval by Semantic Attributes
RSC	Retrieval by Spatial Constraints
RSS	Retrieval by Shape Similarity
VCLI	Virage Command Line Interface
VIMSYS	Visual Information Management System
VIR	Visual Information Retrieval
VSPDB	Visual Search Photo DataBase

EPILOGUE

I always wished to have some sort of creative or artistic ability. I very much admire people who can write a book, compose music or create an art object. I especially admire them when these art forms somehow give rise to emotions. Although this graduation paper is of course not meant to be literature, I tried to make it as readable as possible.

Originally, I was brought up to be a botanical analyst. My choice for this study was based on my interest in Biology, my grades for Chemistry and just not knowing what else to do. My ideas about the kind of jobs I could fill seemed appealing. Unfortunately, during the last year of my study the labour-market for this line of work collapsed and my chances for a job became poor.

Only 5 months after I got my diploma, I made the final choice to be trained as a computer programmer. This choice was merely based on chances for getting a job. My analytical abilities and my experience with writing small Pascal programs during my study also played a role. After a five-months training I entered the information technology line of business as a programmer.

After having followed several courses, which weren’t always very satisfactory, I sought for other ways of learning more about computer science. That is why I started evening classes at the University of Leiden. During this study I learned a lot and it also gave me great pleasure and satisfaction doing it.

A study of Computer Science normally has a minimum duration of 4 years. Doing it in the evenings, in tandem with a job and other obligations, it will take 6 years. In the last year, the study will be completed with a project on a specific subject, resulting in a Master Thesis Paper. The project for completing the study can be done at the University, externally within a research institute or at your (or another) company.

In my case it wasn’t practical or possible to do the latter. The offer from the University to do some literature research in the area of multimedia was received by me with open arms. Because of my interest in databases, this area will have the focus.

At first, I started writing this paper in Dutch. I ran, however, into problems with all the English terms. Although some terms have been generally adopted, other words can’t be simply translated into Dutch. Their meaning wouldn’t be the same, they wouldn’t be clear or they would lose their strength. The decision was then made to write it in English.

Another problem that occurred during the creation of this Thesis is the occurrence of abbreviations. Within the area of Computer Science it is quite common to use them. When I want to use a shortened form of a term, I first will use the full term and I will put the abbreviation behind it between brackets. Generally adopted abbreviations won’t be explained in this manner. When the common use is doubted, it will be listed.

At the end of this epilogue, I want to thank all the people who helped me with the coming about of this thesis. I won’t name them, but I’ll hope their contribution is clear to them. I’ll make two exceptions, namely, for Ida Sprinkhuizen-Kuyper, who coached me through the whole process, and Jan de Rooij, my partner in life. He supported me during all of the years this study lasted.