MiЯRor: Multimedia Query Processing in Extensible Databases

Arjen P. de Vries
Centre for Telematics and Information Technology
University of Twente, The Netherlands
arjen@cs.utwente.nl

Abstract

The miЯRor project investigates the implications of multimedia information retrieval on database design. We assume a modern extensible database system with extensions for feature based search techniques. The multimedia query processor has to bridge the gap between the user's high level information need and the search techniques available in the database. We therefore propose an iterative query process using relevance feedback. The query processor identifies which of the available representations are most promising for answering the query. In addition, it combines evidence from different sources. Our multimedia retrieval model is a generalization of a well-known text retrieval model. We discuss our prototype implementation of this model, based on Bayesian reasoning over a concept space of automatically generated clusters. The experimentation platform uses structural object-orientation to model the data and its meta-data flexibly, without compromising efficiency and scalability. We illustrate our approach with some first experiments with text and music retrieval.

Keywords: Multimedia Information Retrieval, Digital Libraries, Multimedia Query Processing, Inference Network Retrieval Model

1 Introduction

Large archives of digitized multimedia data are set up today, and more and more digitized data will become available online. Digitized multimedia data cannot be searched directly on its binary content. Content-based access to multimedia data therefore requires meta-data about the objects. Meta-data may be manually added descriptions, but can also consist of automatically extracted **features**. Such features are low-level representations of multimedia data, like color distribution and texture [10].

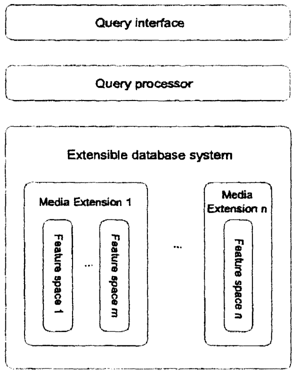

Figure 1: Multimedia database architecture

Traditional database technology, mainly developed for administrative applications, has severe shortcomings with respect to the support of multimedia digital libraries. The access to the digitized multimedia objects, the extraction of meta-data from these objects, and the management of the objects and the meta-data, all have characteristics very different from administrative applications. In the miЯRor project, we study these different requirements on database support, with the purpose to design multimedia database management systems accordingly.

The miЯRor database architecture is especially targeted to support application development in the multimedia digital library environment. It consists of three layers, corresponding to the three light gray boxes in figure 1. At the bottom, we assume an extensible database system, with extension modules (also known as 'data blades' or 'data cartridges') that provide abstract data types (ADTs) encapsulating feature spaces and their distance measures. Our research concentrates on the query processor in the middle box. At the top, we assume a user interface that supports the interaction between the user and the multimedia database.

In [7] and [5], we describe our view on the bottom layer. We introduce an open distributed architecture for the management of multimedia data and its associated meta-data. Using this architecture, many independent parties can easily cooperate in the construction of a digital library. The extraction of meta-data from the objects in the library is a transparent process and takes place automatically when new data becomes available. A very important aspect of the architecture is modular extensibility. New data formats and new meta-data extraction software can be easily 'plugged in'.

Users typically do not know how to express their information needs in database queries, making the support of multimedia retrieval a tough problem. As we argued in more detail in [8], textual queries cannot capture the full semantics of multimedia data. Content-based retrieval techniques may provide the 'missing' semantics. Querying multimedia data using feature models is performed using example objects; a distance measure between the feature representations of two multimedia objects expresses the similarity between those objects. However, the gap between the meta-data used in the content-based retrieval techniques and the concepts in the users' minds is too big. We term this the **query formulation problem**.

The query processor in the middle layer bridges this gap between user and extensible database system. In the remainder of the paper we focus on its design and implementation. We start with an informal example in section 2, illustrating the query formulation problem in multimedia databases. Next, we introduce in section 3 our approach to multimedia query processing. We discuss the design and implementation of our prototype multimedia database management system in section 4. We are especially concerned with the issues of efficiency and scalability of the architecture. In section 5, we demonstrate the functionality supported in our system with some (small-scale) experiments in text and music retrieval.

2 THE PROBLEM OF QUERY FORMULATION

Imagine a journalist writing an article on *the effects of the recent economical crisis in Asia.* Part of the journalist's task is to illustrate the article with photos that hopefully attract readers and increase the sales of the magazine or news paper. A study of journalists at work, reported in [15], made clear that for such 'feature articles', journalists have more freedom than with normal news items. For example, the function of the photo may also be to evoke associations. Also, there is more time to find a 'good' photo.

A journalist usually considers more than one concept for a single illustration task. For the economical crisis example, a possible concept could be a very crowded stock market. Another illustration idea is a photo demonstrating that normal people do not have much money left to spend, for example by showing an empty shopping street in otherwise crowded Hong Kong. In both cases, a photo expressing despair or panic is probably preferred over photos without explicit emotions. Furthermore, constraints like overall page layout may affect the choices made while performing the illustration task.

Assume now that the journalist has access to a video archive of news bulletins originating from various broadcasters. In the archive, the time, date, and source are maintained for each news bulletin. The video data itself is modelled with a sequence of key-frames, and a text version of the audio track. The content of the key-frames is indexed using color and texture features. For comparison, a news archive storing similar meta-data is described in [13].

Searching for 'stock market' in the subtitles may be rather succesful as an initial query. The precision of the results is probably high, meaning that most key-frames with matching subtitles really show stock market scenes. However, the recall may be low: many scenes at stock markets may not have been labelled with an explicit annotation mentioning 'stock market'. Note that this problem will be much worse for the second illustration idea, using 'Hong Kong shopping street' as a text query.

Emotional aspects of the images searched are especially hard to capture in a textual query. Searching for 'despair' in subtitles will probably not retrieve many useful results. These aspects of the illustration task may be captured more easily in terms of feature representations of the images. However, the journalist cannot possibly be expected to express a high level concept like 'despair' in a combination of color and texture features. Conversely, the internal representation of the video with its meta-data should be completely invisible to the users.

3 Multimedia query processing

3.1 Desiderata

The query formulation problem leads to a different view on query processing than common in the database community. Instead of a one step process with a single query, and the database simply retrieving its matching objects, the interaction between a multimedia database and the user should be a *dialogue*. The query processor should iteratively interpret the user's judgements on the results of the previous step, and adapt the initial query such that it will better reflect the observed but unknown information need. It derives database queries against the meta-data, using information from the interaction with the user.

An iterative approach to query processing is already common in information retrieval (IR) systems [24]. We therefore base the miЯRor query processor on the theory and techniques developed in the IR research field [6]. However, a multimedia database management system differs significantly from a special purpose text retrieval system. The management of multimedia data requires extensible systems [7, 5]. IR systems are not designed for extensibility. The implementations assume detailed knowledge about the structure of the indexed documents and the meta-data that models the content. In an extensible system however, we do not know beforehand what representations of the multimedia objects will be available as meta-data at run-time.

A somewhat related difference between IR and multimedia databases is the number of sources of evidence used in the retrieval process. In IR, only a small number of different sources is considered, e.g. abstract, full text, citations, and maybe hypertext links. On the other hand, the combination of evidence from *many* different sources is crucial for multimedia retrieval. Experiments with the Foureyes learning agent for the Photobook image retrieval system demonstrated the advantages of a collection of data-dependent and task-dependent feature spaces over a universal similarity measure defined on a generic feature space [16, 17]. Different feature spaces capture different aspects of the data. Typically, a feature space performs only well at a small set of tasks, on only a subset of the data. Rather than a carefully selected 'society' of models as envisioned in Foureyes, 'anarchy' seems however a more appropriate metaphore in our context; indeed, in miЯRor the collection of feature spaces changes dynamically as new meta-data extraction software is added or removed.

3.2 Retrieval model

Figure 2 proposes the design of the miЯRor query processor. An IR system is described by its **retrieval model**, which defines the document representation, the query formulation, and the ranking function [26]. These three aspects are reflected in the design of our multimedia query processor, in subsequently the **concept layer** (document representation), the **evidential reasoning layer** (ranking function), and the **relevance feedback layer** (query formulation).

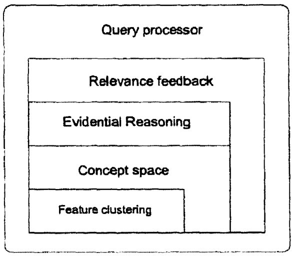

Figure 2: The multimedia query processor

3.2.1 Concept layer

The concept layer defines the basic units representing the content of the multimedia objects. IR literature usually refers to these units as the **indexing features**; to avoid confusion with the features used in content-based multimedia retrieval, we prefer to call these **concepts**. The concepts are input to the evidential reasoning layer, which selects the objects in the database that best match the user's query.

Most IR systems use the words occuring in the document as concepts. In text documents, words naturally refer to classes of objects in the real world. For example, the word 'street' occuring in an English text is the same, whether that particular street is located in Cambridge or Oxford. Sometimes, words occuring in the text are first clustered, using **stemming** algorithms and **thesauri**. This may alleviate the problems with ambiguity in natural language.

In multimedia retrieval, the content representation of objects is a (usually unique) point in multi-dimensional feature space. Therefore, an important task of the concept layer is **feature clustering**. The feature representation of a street in Cambridge will be different from the representation of a similar street in Oxford. To complicate matters, the representation of one and the same street in two different images will usually be different as well. Hence, before we can develop a theory for multimedia retrieval similar to the retrieval models in IR, we have to cluster these points, based on their relative positions in feature space.

The concept layer uses unsupervised clustering algorithms to identify clusters in feature space. Of course, we realize that not no algorithm will automatically cluster all streets in a single concept. Nor do we expect to construct concepts that only occur in a subset of the streets but in no other classes of objects. However, the assumption underlying the content-based retrieval techniques is that proximity between points in feature space corresponds to some sort of similarity in the real world. Thus, the proximity of the clusters' feature points is likely to reveal an implicit underlying concept that captures some of the semantics of the objects.

3.2.2 Evidential reasoning layer

The responsibility of the evidential reasoning layer is to identify the multimedia objects in the database that may fulfill the user's information need as expressed in the query. The evidence is based on the presence or absence of concepts, very similar to traditional IR. The evidential reasoning process combines the evidence from different sources into a single judgement. It should take into account the structural composition of objects from their component objects. We discuss the evidential reasoning layer in more detail in section 3.3.

3.2.3 Relevance feedback layer

The relevance feedback layer has two tasks. First, it is responsible for query (re-)formulation. It controls the dialogue between the user and database, analyzing the user's feedback information and changing the query such that it (hopefully) better reflects the user's information need. We term this online processing **query-space modification**. Second, the relevance feedback layer maintains a history for offline processing, logging the interaction between users and database. Supervised clustering techniques may use these logs to improve the initial clustering constructed in the concept layer. Also, statistical tests may identify dependencies between feature spaces. We refer to this task as **object-space modification**. Although we regard both types of feedback as important, we currently focus on query-space modification.

3.3 Reasoning layer

The 'probability ranking principle' states that an object ranking is optimal when the objects are ranked by their probability of relevance to the user [24, p. 113]. Many competing IR theories can be used to estimate these probabilities. We base our theory for multimedia retrieval on the inference network retrieval model, introduced by Turtle and Croft [22, 23]. It has been shown that this probabilistic model can also express other common retrieval models, such as the Boolean and the vector space model. The model is based on the theory of Bayesian belief networks. A Bayesian belief network is a graph representation of probabilistic knowledge. In a belief network, nodes represent random variables, and arcs reflect relationships between the linked variables. The direction of an arc between parent node and child node represents causality. The strength of this causal influence is expressed by a conditional probability. A belief network encodes a joint probability distribution. The advantage of the network representation of this distribution is that inference procedures exist to compute the value of any conditional probability in the network given the available evidence, without having to derive a closed form formula for the complete distribution. The reader is referred to [19] for more details.

Turtle and Croft claim advantages of their model over different retrieval models because of its theoretical foundation in Bayesian belief networks. Unfortunately, due to the simplifications made to the inference procedure and the network structure (trading mathematical correctness for efficiency), it is hardly possible to take advantage of theoretical developments in the more general theory of Bayesian networks. Nevertheless, we take this model as a starting point for the development of a theory for multimedia information retrieval [6]. It is very suited for our purpose, since it has been introduced in IR to combine evidence from different sources more easily. Also, it has a modular structure that reflects the architecture's extensibility. Most importantly, its implementation in the InQuery retrieval system has been very succesful in many IR evaluation experiments.

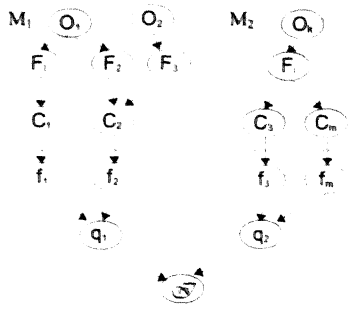

Figure 3: A multimedia retrieval model based on Bayesian inference networks

3.3.1 The network structure

Using figure 3, we first explain the general idea behind miЯRor's version of the inference network retrieval model. Each base type, e.g. image or audio, has its own media extension $\mathcal{M}_i$. A media extension, depicted as a dark gray box in the figure, manages a collection of content representations $\mathcal{F}_j$, shown as light gray boxes. The nodes in the network represent binary random variables. The top part of the network is called the **object network** and is static for a given data collection. The bottom part, the **query network**, is dynamically created by the relevance feedback layer, based on interaction with the user.

At the roots of the network, we find the object nodes O_i. For now, we will ignore the internal structure of the multimedia objects; all objects are considered **atomic**. In section 5.2, we discuss the retrieval of compound documents to illustrate a possible approach to modelling structured objects. The objects O_i are connected to their meta-data representations of content F_j. The concept nodes C_p represent the concepts identified by clustering in the concept layer. The model allows concept clusters to overlap. Thus, a single representation node may be connected to several concept nodes. Node $\mathcal{I}$ in the query network represents the user's information need. The information need is expressed by the example objects provided by the user in the interaction process. The query nodes q_i model these example objects. The meta-data extracted from these objects is represented by the f_j nodes. These nodes are connected to their corresponding concept nodes in the static object network. In the dialogue between database and user, the relevance feedback layer adapts the structure of the query network by adding or removing nodes.

Let us take a closer look at the example instantiation of the network model given in figure 3. Assume that $\mathcal{M}_1$ is an image media extension. It manages feature spaces $\mathcal{F}_1$ for color and $\mathcal{F}_2$ for texture. Image object O_1 has a color feature F_1 and a texture feature F_2. Color feature F_1 has been clustered into concept C_1, and texture features F_2 and F_3 into concept C_2. Color representation f_1 and texture representation f_2, extracted from the example image q_1, are part of the same clusters in feature space, hence also connected to C_1 and C_2 respectively.

3.3.2 Ranking objects

The inference network is used to compute $\Pr(\mathcal{I}|O_i)$, which corresponds to the chance that the information need as expressed in the query network is fulfilled when presenting this object to the user. The random variables associated to the objects and their meta-data represent observations. In the ranking process, each object O_i is considered in isolation: its node is set to true, and all other nodes to false. This evidence is propagated through the network until it reaches $\mathcal{I}$, when we have computed the desired $\Pr(\mathcal{I}|O_i)$.

The joint probability distribution encoded in the object network is independent of the query. In our current model, observing O_i always implies observing its meta-data F_j. We assume the feature spaces independent and equally important. In later revisions of the retrieval model, we may use the conditional probability distribution $\Pr(F_j|O_i)$ to represent knowledge about how reliably each feature space describes an object. $\Pr(C_p|F_j)$ expresses the belief that concept C_p is observed when we observe feature F_j. This probability should be estimated in the feature clustering process. Similarly, $\Pr(f_j|C_p)$, specified at the arcs connecting the object network with the query network, describes our belief that feature f_j in query space is described by the concept C_p in object space.

Instead of first computing these probabilities independently, and then propagating the belief to the nodes f_j in the query network, the implementation of the inference network retrieval model computes $\Pr(f_j|O_i)$ directly. In InQuery, this probability is estimated using term frequency *tf*, inverse document frequency *idf*, and default belief α:

$$\Pr(f_j|O_i) = \alpha + (1 - \alpha) \cdot tf \cdot idf \qquad (1)$$

In a multimedia feature space, we have to define a procedure to estimate this probability using the relative position of that point in a cluster and the distribution of other points in the cluster. An unsupervised clustering algorithm like AutoClass provides such an estimate [4]. As an alternative, we plan to investigate the cluster-based probability model that has been proposed in [20].

3.3.3 Propagation of evidence

To explain the propagation of evidence from the f_i through the query network to $\mathcal{I}$, we introduce a formal description of the inference network adapted from [21]. Let x_i be a node in a Bayesian network G, and Γ_{x_i} be the set of parents of this node. Since G is a Bayesian belief network, the influence of Γ_{x_i} on x_i is specified by the conditional probability distribution $\Pr(x_i|\Gamma_{x_i})$. Let the cardinality of Γ_{x_i} be n, and the random variables be binary like in our retrieval model. Then we have to specify 2^n different probabilities to describe this conditional distribution. Obviously, this is problematic for the computational tractability of the inference. Therefore, we have to find an approximation of the real probability table (also known as link matrix).

Note that, for a node x_i, the influence of Γ_{x_i} on x_i can be specified by *any* function $F(x_i, \Gamma_{x_i})$ that satisfies:

$$\sum_{y \in \mathrm{Y}} F(y) = 1 \qquad (2)$$

$$0 \leq \mathrm{F}(y) \leq 1 \qquad (3)$$

where Y is defined as $x_i \times \Gamma_{x_i}$. In the general theory of belief networks, functions approximating $\Pr(x_i|\Gamma_{x_i})$ have been used to model **causal independence** efficiently: the case when multiple causes contribute independently to a common effect. A famous example is the 'noisy-or' model [19]. In his thesis, Turtle gives closed-form expressions for a limited subclass of functions $F(x_i, \Gamma_{x_i})$, that are useful in IR and can be evaluated in $\mathcal{O}(n)$. Greiff gives a larger class of functions, described by so-called PIC-matrices, for which the evaluation depends on the number of parents that are true but not on their ordering [12]. He first provides an evaluation procedure in $\mathcal{O}(n^2)$, and then gives an algorithm in $\mathcal{O}(n)$ for a subclass of these PIC-matrices. Functions in these classes are 'sum', probabilistic versions of logical operators 'and' and 'or', as well as variations of these usually referred to as **'pnorm'-operators**. These functions are all part of InQuery's language to describe the structure of the query network.

Of course, an approximation of $\Pr(x_i|\Gamma_{x_i})$ with a different function $F(x_i, \Gamma_{x_i})$ is only semantically valid if this function behaves similar to the true probability distribution. The succes of the retrieval system InQuery, that is based on the inference network retrieval model, is often given as 'proof' that these functions really model the true probabilistic dependencies between for example the concepts and the document's relevance. We do not agree with this line of reasoning. The experiments with InQuery demonstrate *only* that the computed value for $\Pr(\mathcal{I}|\mathrm{O}_i)$ may be interpreted as a good approximation of the probability of relevance of the O_i. The distribution captured by the complete network apparently reflects its desired interpretation in the real world. However, we should not deduce that the probability estimates for the nodes x_i and their parents also have an interpretation regardless of the choice of $F(x_i, \Gamma_{x_i})$. This observation is confirmed by the difficulties with chosing an optimal value for default belief α (cf. equation 1) in the experiments with 'pnorm'-operators reported in [12]. Despite of these limitations, the inference network retrieval model is a very powerful model because of its ability to flexibly model varying approaches to the combination of evidence from different representations. Also, the original Bayesian belief network underlying the retrieval model, without its approximations used to achieve tractability, can still be used as a reference when we want to understand why some operator combined with some formula estimating the concept probabilities does or does not work well.

4 Design and implementation

The implementation of miЯRor's multimedia query processor requires the integration of IR and databases. Integration of IR and databases has historically led to impractically slow systems; the efficient execution of IR techniques required special purpose software systems. We believe that IR and databases *can* be integrated in a single system, but only if this integration is complete, and neither a layer on top of, nor a black box inside a database system. Therefore, our prototype implementation is based on **structural object-orientation**. A detailed discussion of the benefits of structural object-orientation for IR processing in a database system can be found in [9].

Figure 4 shows the design of our research prototype. The design is focused on the development of a system that will scale up to very large

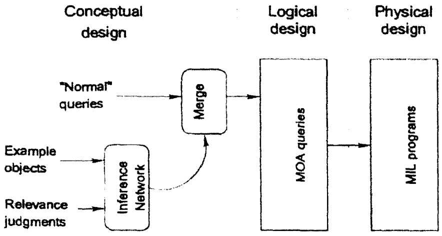

Figure 4: Design

data collections. Its main characteristic is the strict separation between the logical and physical databases. This separation provides data independence, and allows for algebraic query optimization in the translation from expressions at the logical level to queries executed in the physical database. Also, parallellization of the physical algebra is orthogonal to the logical algebra, such that we can transparently distribute the data over different database servers by changing only the mapping between the two views. In this paper, we only discuss query processing at the logical level. The interested reader is referred to [9] for a discussion of the implementation in the physical database.

MOA is an object algebra for the logical level, being developed by our research group. It provides an extensible nested object data model and an algebra on this model. The prototype implementation does not yet provide a query language at the conceptual level; queries can only be specified using MOA expressions. The MOA Tools translate the query expressions specified in MOA into efficient MIL programs[1] that are executed in the **Monet** database system [1]. Monet is an extensible parallel database kernel that is intended to serve as a backend in various application domains [2]; e.g., image retrieval is supported by an extension module defining the 'Acoi' algebra [18]. Monet has also been used succesfully for geographic information systems as well as commercial data mining applications.

MOA's data model is based on **base types** and **structuring primitives**. Base types are ADT-style types. They are inherited from the physical database schema, including common types such as `int` and `str`, but also large object types like `Image`. A structuring primitive combines known types to create a **structured type**. Common examples in object-oriented data models are bag, set, and tuple. To demonstrate the specification of multimedia data collections in MOA, we give in example 1 the definition of a structured data type for the video archive mentioned in section 2.

Example 1

```
BAG<
  TUPLE<
    time:        Atomic<Time>,
    date:        Atomic<Date>,
    keyframes:   LIST<
                   Atomic<Image>
                 >,
    audiotrack:  Atomic<Audio>,
    transcript:  Atomic<Text>
  >
>;
```

In the implementation of the query processor, we perform the evidential reasoning process as database queries. For this purpose, we extend MOA with structures for components of the inference network. Operations on these structures model the propagation of beliefs within a component. The resulting language allows us to specify many different network topologies, by simply choosing varying operators to combine different sources of evidence. The relevance feedback layer can thus adapt the network structure by simply generating different MOA expressions.

For the integration of content-based querying in MOA, we first define a structure that encapsulates the object network. The `CONTREP` structure is defined as the content representation of object O_i in feature space $\mathcal{F}$. If an object has meta-data representations in several feature spaces, then each combination of object and feature space is modelled in a distinct instantiation of this structure. The concept layer constructs a `CONTREP` from the output of the feature clustering process. Recall that the $\Pr(\mathrm{f}_j|\mathrm{O}_i)$ are estimated directly from the statistical distribution of occurrences of C_p in O_i and in the collection. Therefore, we can sufficiently describe the object network for O_i by the C_p present in the object. Thus, a `CONTREP` stores the connections from node O_i to its associated nodes C_p in $\mathcal{F}$. In the current prototype, the clustering of a set of features is performed outside the database, and the `CONTREP` structures are bulk-loaded from files describing the identified concepts.

We also extend MOA with two other structures, that allow us to specify the propagation of evidence through the query network. The `INFNET` structure models a node x_i with its parents Γ_{x_i}.

[1] MIL stands for Monet Interface Language

It can be constructed from a set of probabilities, in which each value corresponds to the belief in a node of Γ_{x_i}. The structure defines operators for the class of functions $F(x_i, \Gamma_{x_i})$ that is expressed by PIC-matrices [12]. DOCNET is a specialization of INFNET that is optimized for the assignment of default beliefs α to nodes that do not occur in the content representation of an object.

The three structure extensions interact as follows in the computation of $\Pr(\mathcal{I}|O_i)$. The relevance feedback layer constructs a query network, based on the example objects provided by the user. In the first step of belief computation, CONTREP's operation getBL connects the query network to the object network. Its operands are the f_j nodes of the same feature space as the CONTREP, and a structure representing global statistics of the feature space. This operation computes estimates of $\Pr(f_j|O_i)$, returning a DOCNET structure capturing the instantiation of the nodes at the top level of the query network. A belief operator $F(q_i, \Gamma_{q_i})$ then computes an estimate of $\Pr(q_i|\Gamma_{q_i})$. Next, we repeat constructing an INFNET from these estimated probabilities, and computing the belief in the nodes at the next level of the query network, until we reach node $\mathcal{I}$. We then have computed $\Pr(\mathcal{I}|O_i)$ using the joint probability distribution described by the inference network.

Example 2

```
BAG<
  TUPLE<
    time: Atomic<Time>,
    date: Atomic<Date>,
    keyframes:
      LIST<
        TUPLE<
          keyframe: Atomic<Image>,
          color:    CONTREP,
          texture:  CONTREP
        >
      >,
    audiotrack: Atomic<Audio>,
    transcript:
      TUPLE<
        transcript: Atomic<Text>,
        content:    CONTREP
      >
  >
>;
```

In combination with standard MOA structures like bag and tuple, we can now define and manipulate multimedia data collections and their metadata. For each feature space modelling the content of a multimedia object, we define a CONTREP structure. Since this structure is an orthogonal extension of MOA, we can also query the collection on the combination of content with conventional attributes. For example, we can easily restrict the query results of a content query to a ranking of only last week's news bulletins. Example 2 extends the type definition for the video archive example with its content representations. Of course, the content representations may be hidden from end users, such that they only see the definition of example 1.

5 Examples

5.1 Text retrieval

We first implemented a simplified version of the original inference network retrieval model, leaving out its proximity operators. Assume now that docs is a bag of content representations of text documents, query is a collection of query terms, and stats provides collection statistics such as *idf*. The MOA expression in example 3 computes $\Pr(\mathcal{I}|O_i)$ as described in the previous section. A map on a bag performs an operation on all elements of the bag. In the specification of the operation to be performed, the bag's element is referred to as THIS. Since the getBL constructs a DOCNET, the inner map converts the bag of document representations in a bag of DOCNET structures. The outer map uses the 'sum' belief operator to compute the probability of relevance for each document.

Example 3

```
map[sum(THIS)](
  map[getBL( THIS,
    query, stats ) ]( docs ));
```

5.2 Compound documents

In the discussion of our retrieval model so far, the objects O_i have been assumed atomic. We will now rank compound documents on logical units like sections or chapters, rather than on their full content. In example 4, we model the content of a news document as a bag of items. The topology of the inference network specified by this particular query is taken from [3]. These experiments suggested that the best results are achieved when a document is ranked by the contribution of its best section. Note the use of the INFNET constructor, to express the belief propagation through an extra layer of nodes in the query network.

Example 4

- *data definition for compound documents:*

```
BAG<
  TUPLE<
    Category : str,
    Content  : BAG< CONTREP >
  >
>;
```

- *ranking news documents by their best items:*

```
map[max( INFNET<THIS> ) ](
  map[ map[ sum(getBL( THIS,
         query, stats ))](
    THIS.Content ) ]( docs )));
```

5.3 Music retrieval

We conclude the paper with a small scale multimedia retrieval experiment using our experimentation platform. The results should not be given more status than just 'proof of concept'. Although the experimental evaluation has not been very thorough, the results are encouraging. Indeed, it seems possible to interactively retrieve groups of similar songs, in particular for well defined categories.

In multimedia retrieval, emotional and aesthetic values play an important role in the user's evaluation process [5]. Because subjective judgments seem especially important when we compare music fragments, we decided to try out the multimedia query processor on a content representation of music objects. Note that we assume the similarity between two fragments to be defined by the overall 'sound' of the music. The extraction of meta-data is based on [25]. We augmented the feature vectors with a simple rhythm indicator based on peaks in the autocorrelation function of the lowest parts of the frequency domain.

Data set **Symbol-1**, created in cooperation with the Dutch company 'Symbol Automatisering', consists of 287 songs. Domain experts of Symbol Automatisering have manually classified these songs into six main categories: rock, house, alternative, easy listening, dance, and classical. We sampled between one and two minutes of each song, that we segmented into fragments of 5 seconds each. The result is a data collection of 3363 fragments for which we computed the feature vectors. Feature clustering with Autoclass identified 53 different clusters; we assigned to each feature vector the concept node according to the cluster with the highest probability. We then modeled a song as a collection of these concepts. We treated this representation of songs as if they were text documents in which the concepts are the words. Thus, we simply used equation 1 to estimate $\Pr(f_j|O_i)$. In future experiments, we plan to evaluate the representation of songs in more detail, e.g. using the $\Pr(f_j|C_p)$ estimated by Autoclass, and using all concepts detected in the fragment.

We performed the following experiment with music retrieval from this collection. Simulating online relevance feedback, we constructed a query network of the concepts that occurred most frequently in half of the songs belonging to a category. We then tried to retrieve other songs of the same category. Of the top 20 songs for the query based on 'rock', 15 had also been classified manually as rock. Of the other 5 songs, only 2 clearly do not belong in the rock category. With the 'classical' and 'house' songs, we found hardly any misses. Results for the category 'alternative' were however hardly better than chance, but maybe this is partly because the category is not well defined.

6 Conclusion and future research

We developed a multimedia query processor that supports the end users of a multimedia database with query formulation. The architecture is extensible with new algorithms for meta-data extraction, and the query processor is designed to use the available representations transparently. The integration of the content-based query processing in MOA also allows the user to query both the logical and the content structure of multimedia objects. The main contribution of our work is the design for scalability.

Improving the basic functionality of the prototype is a topic high on our research agenda. From a technical viewpoint, we should implement clustering in our architecture. Also, we want to experiment with multiple representations in the database. The foundation of the model in the theory of probabilistic networks provides a strong theoretical framework [19, 11, 14]. Within this framework, there is a lot of scope for experiments and we would like to investigate its use to model and learn dependencies between representations.

An important but open research issue is the development of an evaluation methodology for multimedia retrieval. The inherent subjectivity in multimedia searching makes it impossible to develop a test suite that is not related to a real user task. We believe the music domain provides a context well suited to evaluate how the query process adapts to subjectivity of the users. However, content modeling of music is not easy and the success criteria are vaguely defined. To evaluate the effect of multiple representations and their interdependencies in retrieval, retrieval from publishers' photo and video archives may provide a better context. However, the challenge in this domain is to construct a test suite with realistic user tasks and clearly defined success factors, without making the evaluation process too expensive (amount of data) and elaborate (user studies).

ACKNOWLEDGEMENTS

Many thanks go to Annita Wilschut, who encouraged me to prototype my view on a multimedia DBMS using structural object-orientation. Dick Theissens of Symbol Automatisering generously provided data for the music retrieval experiment, and Harold Oortwijn implemented the feature extractor. I would also like to thank Henk Ernst Blok for proofreading draft versions of this paper and his useful feedback.

REFERENCES

[1] P. Boncz, A.N. Wilschut, and M.L. Kersten. Flattening an object algebra to provide performance. In *Fourteenth International Conference on Data Engineering*, pages 568–577, Orlando, Florida, February 1998.

[2] P.A. Boncz and M.L. Kersten. Monet: An impressionist sketch of an advanced database system. In *BIWIT'95: Basque international workshop on information technology*, July 1995.

[3] J.P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 1994.

[4] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1995.

[5] A.P. de Vries and H.M. Blanken. Database technology and the management of multimedia data in Mirror. In *Multimedia Storage and Archiving Systems III*, volume 3527 of *Proceedings of SPIE*, Boston MA, November 1998.

[6] A.P. de Vries and H.M. Blanken. The relationship between IR and multimedia databases. In *IRSG'98*, Autrans, France, March 1998.

[7] A.P. de Vries, B. Eberman, and D.E. Kovalcin. The design and implementation of an infrastructure for multimedia digital libraries. In *Proceedings of the 1998 International Database Engineering & Applications Symposium*, pages 103–110, Cardiff, UK, July 1998.

[8] A.P. de Vries, G.C. van der Veer, and H.M. Blanken. Let's talk about it: Dialogues with multimedia databases. Database support for human activity. *Displays*, 18(4):215–220, 1998.

[9] A.P. de Vries and A.N. Wilschut. On the integration of IR and databases. In *IFIP WG 2.6 Working Conference on Database Semantics - Semantic Issues in Multimedia (DS-8)*, Rotorua, New Zealand, January 1999. Accepted as short paper.

[10] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[11] R.M. Fung and B.A. Del Favero. Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(3):43–48, March 1995.

[12] W. Greiff, W.B. Croft, and H. Turtle. PIC matrices: A computationally tractable class of probabilistic query operators. Technical Report IR-132, The Center for Intelligent Information Retrieval, 1998. submitted to ACM TOIS.

[13] A.G. Hauptmann and M.J. Witbrock. *Intelligent multimedia information retrieval*,

chapter Informedia: news-on-demand multimedia information acquisition and retrieval, pages 215–239. AAAI Press/MIT Press, 1997.

14] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced technology division, March 1995. Revised edition November 1996.

[15] M. Markkula and E. Sormunen. Searching for photos - journalists' practices in pictorial IR. In *The challenge of image retrieval*, Newcastle upon Tyne, UK, 1998. University of Northumbria.

[16] T. Minka. An image database browser that learns from user interaction. Master's thesis, MIT, 1996. Also appeared as MIT Media Laboratory technical report 365.

[17] T.P. Minka and R.W. Picard. Interactive learning using a "society of models". Technical Report TR-349, MIT Media Laboratory Perceptual Computing Section, 1997. Submitted to Special Issue of Pattern Recognition on Image Databases: Classification and Retrieval.

[18] N. Nes and M. Kersten. The Acoi algebra: A query algebra for image retrieval systems. In *Advances in Databases. 16th British National Conference on Databases, BNCOD 16*, pages 77–88, Cardiff, Wales, UK, July 1998.

[19] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference.* Morgan Kaufmann, California, 1988.

[20] K. Popat and R.W. Picard. Cluster-based probability model and its application to image and texture processing. *IEEE Transactions on Image Processing*, 6(2):268–284, February 1997.

[21] B.A.N. Ribeiro and R. Muntz. A belief network model for IR. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval (SIGIR '96)*, pages 253–260, Zürich, Switzerland, August 1996.

[22] H.R. Turtle. *Inference networks for document retrieval.* PhD thesis, Univeristy of Massachusetts, 1991.

[23] H.R. Turtle and W.B. Croft. A comparison of text retrieval models. *The computer journal*, 35(3):279–290, 1992.

[24] C.J. van Rijsbergen. *Information retrieval.* Butterworths, London, 2nd edition, 1979.

[25] E. Wold, Th. Blum, D. Keisler, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3), 1996.

[26] S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, January 1995.