GMD Report
43

GMD –
Forschungszentrum
Informationstechnik
GmbH

Silvia Hollfelder, André Everts,
Ulrich Thiel

Designing for Semantic Access: A Video Browsing System

November 1998

GMD –
Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

Anschrift der Verfasser/Address of the authors:
Silvia Hollfelder
André Everts
Dr. Ulrich Thiel
Institut für Integrierte Publikations- und Informationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Dolivostraße 15
D-64293 Darmstadt

E-mail: {hollfeld, everts, thiel}@darmstadt.gmd.de
http://www.darmstadt.gmd.de/~{hollfeld, everts, thiel}

ISSN 1435-2702

Abstract*

Digital Library applications based on huge amounts of digital video data require efficient browsing and searching mechanisms for the extraction of relevant information. To avoid information overload, the browsing system needs to preselect shots of interest from the database in a user-adequate manner. Additionally, it should support continuous presentation of time-dependent media. In this paper, a browsing system architecture is proposed that offers conceptual, content-based access to videos, and content-based buffering and admission control. It consists of the following components. The core of our architecture is a retrieval engine which calculates relevance values for the results of a conceptual query by feature aggregation on video shot granularity. An admission control mechanism supports highly interactive browsing scenarios and uses application semantics for the admission of new client sessions. We extended a commercial DBMS system for continuous media transport and presentation. An intelligent client buffer strategy uses the relevance values from retrieval for browsing support. By this approach a higher system throughput is achieved.

Keywords: browsing architecture, semantic browsing, video retrieval, content-based search, MM-DBMS

Zusammenfassung

Anwendungen in digitalen Bibliotheken basieren auf enormen Mengen an Videodaten und benötigen zum Extrahieren der relevanten Informationen effiziente Browsing- und Suchmechanismen. Um eine Informationsüberflutung zu vermeiden, muß das Browsingsystem Funktionen anbieten, die eine Vorauswahl an Szenen aus der Datenbank in einer für den Nutzer adäquaten Weise ermöglichen. Zusätzlich muß das System die Präsentation zeitkontinuierlicher Medien unterstützen. In diesem Artikel wird eine Systemarchitektur für Videobrowsinganwendungen vorgestellt, die sowohl konzeptuellen, inhaltsbasierten Zugriff auf Videos, als auch inhaltsbasiertes Puffern beim Client und Zulassungskontrolle ermöglicht. Sie besteht aus den folgenden Komponenten: der Kern der Architektur ist eine Retrieval-Engine, die die Relevanzwerte der Ergebnisse einer konzeptuellen Anfrage durch Aggregation von Kennwerten einzelner Videoszenen errechnet. Der Zulassungskontrollmechanismus unterstützt stark interaktive Browsingszenarien und benutzt hierfür Applikationssemantik. Ein kommerzielles DBMS System wurde um Transport- und Präsentationskomponenten für multimediale Daten erweitert. Eine intelligente Clientpufferstrategie setzt die Relevanzwerte des Retrievalvorganges zur Unterstützung von Browsingsitzungen ein. Dadurch wird der Durchsatz des Systems erhöht.

Keywords: Browsing Systemarchitektur, Semantisches Browsen, Video Retrieval, Inhaltsbasierte Suche, MM-DBMS

*This work was funded by the ESPRIT joint project (Long Term Research No. 9141) **HERMES** (Foundations of High Performance Multimedia Information Management Systems).

1 Introduction

Large collections of multimedia data play a major role in new types of networked information systems such as digital libraries (DLs), virtual museums, and others. A major problem is that users have to browse through large amounts of data to find relevant, i.e. informative, parts of documents.

A common feature of these emerging applications is the need to support users with vague or imprecise information needs. In contrast to conventional video-on-demand systems, the (part of the) video the user wants to see is not easily identified by selecting a video title, e.g., from a list of movies. Instead, the user usually may only be able to characterize a sought-for video by providing conceptual terms which describe the video's properties. Therefore, the typical usage will consist in submitting a broad query first, and then browsing the retrieval result. Browsing is even more important for video than for text-based media because pictorial data in general have even wider ranges of possible semantic interpretations than texts. Hence, the user must be able to inspect retrieval results in an efficient way.

A browsing system has to provide support for this kind of information-intensive work. Content-based retrieval, based on low-level image features like colour distribution, textures, and contrast was used in early approaches (e.g., QBIC [FSN$^+$97]), usually in a query-by-example mode. Since query specification based on these basic image features is not acceptable for a user, more sophisticated access modes have to be provided. For example, similarity search based on feature combinations, or pattern recognition methods, are more reliable and user-friendly. But these approaches require that the user has concrete ideas of his/her information need or can select or provide a sample image, or is willing or able to draw sketches, etc. Even if users accept this clumsy way of accessing the database, the poor retrieval quality yields unsatisfying result sets. Techniques like clustering and feedback can improve this situation, but users browsing through video databases will not accept long periods of query refinement (so-called users in the loop) due to the long presentation time of time-dependent data (such as videos). Hence, users need efficient ways to access video collections based on task-related, conceptual, and content-based criteria. On the other hand, up to now, semantic access to video data has had to cope with severe problems: manual indexing of data is costly and a not very reliable practice, since humans may have different interpretations of a given picture; and speech analysis applied to sound tracks [PSAP98] cannot be used for all applications, e.g., due to lack of analyzable speech or noise where the dialog often refers to topics which have no relation to the shown scene. In short, when we want to retrieve pictoral information, the sound track may be not useful at all in many cases. In this paper we describe how semantic access to the visual information conveyed in videos can be achieved by rule-based interpretation of feature aggregations, which thus can be automatically associated with semantic terms.

Browsing applications not only have an impact on the required data access functionality, but also on the presentation and interaction capabilities. The main goal of users in video browsing applications is to find relevant information quickly, since they will not have time to go through the whole collection and view videos as a whole. Usually, they only inspect shots of interest in sequence, called ”sneak previews”. Depending on the application, a user may pursue different goals in a browsing session. We distinguish between *explorative browsing* and *serendipity browsing* [CW88]. Within explorative browsing a problem-solving user is assumed who has a specific subject in mind about which he wants to collect information. Serendipity browsing occurs when a user is scanning through data not necessarily with concrete goals in mind, and suddenly finds interesting, but unexpected items. Furthermore, the accuracy of inspecting found data varies: some users may aim at getting an overview of all hits in the hit list, others may intend to extract detailed information from the result set. This results in different browsing behaviors. For presentation purposes the system has to support these various browsing behaviors by providing a continuous playback of videos or video chunks and full VCR functionalities such as ”fast forward” and ”rewind”. In addition to these general presentation requirements, browsing support for content-based video browsing needs to provide for quick switching in terms of short delays between the playout of shots from several videos within a browsing session [HS98, Red97].

To support a large number of users, the system has to optimize its throughput. The drawback of most contemporary approaches to high-performance multimedia (MM) systems is that knowledge about contents of MM data such as relevance to given queries is not considered for performance improvement. Our browsing system architecture addresses this problem by a client buffering strategy that employs information about relevant video shots with regard to a previous query as well as presentation requirements in general [THE98]. Thus, the amount of irrelevant data that are sent to users is minimized. The session-based admission control mechanism takes application-specific browsing features into account by estimating the required resources for a browsing session based on the hit list of a previous query.

The paper is structured as follows: We first discuss related work on browsing systems in Section 2. In Section 3 we describe our overall system architecture with the focus on content-based buffering (Section 3.4), resource prediction for browsing scenarios and admission control (Section 3.5), and multimedia retrieval techniques to achieve content-based access by feature extraction, feature aggregation, and classification of videos based on shot granularity in Section 4.1. An example of a query based on our running protoype is given in Section 4.3.

2 Related Work

Early work on system architectures for browsing applications was proposed by [LAC$^+$95]. They developed the Virtual Video Browser (VVB) [Web] for the location, identification, and delivery of digital audio and video in a distributed system. The VVB allows the interactive browsing and content-based querying of a video database, and facilitates the subsequent playout of selected titles. The VVB is designed for a distributed client-server architecture. Several application domains including multimedia-based home entertainment, catalogue shopping, and distance learning are being considered. In the playout phase, a connection is established between the client workstation and a video server for the delivery of video data.

The WebClip system is a working prototype for editing and browsing compressed video over the World Wide Web [SC96]. WebClip is a Web application based on the CVEPS technology (Compressed Video Editing, Parsing, and Search). It supports content-based access to videos using a hierarchical scene-based video browser, the CVEPS client. A video pump located on the CVEPS server is responsible for real-time delivery of continuous media during interactive video viewing over the network.

VideoQ [CCM$^+$97] allows video-based searching on a set of visual features and spatio-temporal relationships. The user can search the videos by sketch or text, or browse the video shots. The video shots are cataloged into a subject taxonomy, which the user can navigate. Each video shot is also manually annotated so the user can perform simple keyword searches.

Other interface designs rely on direct manipulation techniques the user can apply to visual representations of the feature space: While [IKN$^+$98] prefer sliders to adjust feature values in the query, [CBG98] allow the users to employ the functions of a drawing tool to modify sketches and sample images. The modified images can then be used as a starting point for similarity searches.

Besides providing complementary access methods, advanced interface designs try to reduce the user's workload by presenting preferably the interesting or informative parts of video documents. The browsing system proposed by [FBGW98] enables the calculation of so-called confidence scores by means of audio and video analysis. These confidence scores represent the degree of interest of media data, similar to our relevance values as described later. Confidence scores are used to assist the user by its browsing activities in terms of visualization of the score values, specification of rate control corresponding to the confidence scores, and supporting indexing points for random access. In contrast to our approach, the relevance values are considered only at the user interface level.

So far, virtually no method has been published which utilizes information about the relative relevance of fragments of videos for enhancements of system components to increase throughput.

While these approaches do not consider conceptual queries - unless they are supported by manual indexing or sound track based categorization -, other authors report on experimental systems in which the feature representations of pictures and videos are associated with conceptual ”interpretations”. Here, we can distinguish between genre-dependent and more generalistic approaches: The first type of system exploits the knowledge gained from detailed analyses of pictorial material belonging to a specific domain (e.g., ”street landscape images” [SK98], or commercials [CCBP98]), which allow complex constraints to be placed on feature values associated with semantic interpretations (e.g., ”openness”, ”depth” for street scenes, stylistic attributes ("utopic") in the case of commercials). Other advanced MM-Systems, e.g., [WKSS96], [KSN97], employ sophisticated means to identify the contents of videos, e.g., by face detection, keyword detection, and combinations thereof. Usually these approaches are based on predefined selections of features which can be compared with newly entered items in order to detect relevant information.

Only a few steps towards more generality have been reported in the past: In a pioneering experiment, Rowe and Frew [RF97] used neural networks which were trained to associate patterns in feature vectors with semantic interpretations, and recently Grosky [GT98] reported on encouraging results of MM mining, an approach to apply data mining techniques to relate feature data and metadata, e.g., annotation. The mined rules can be employed for retrieval purposes, as can the rules obtained by Quantile-analysis by [ME97], [THE98]. The approach presented here is a more powerful alternative to the latter method.

3 A System Architecture supporting Semantic Video Browsing

The design of a system for semantic video browsing differs from classical MM-DBMS applications such as medical image systems, and video-on-demand (VoD). The main reasons for a modified architecture are directly implied by the task to be accomplished. First, we have to handle vague conceptual queries, and second, we must cope with complex interactive behavior.

Our browsing system is based on a client/server architecture where the server is responsible for the storage of continuous and discrete data. The client is responsible for requesting data units from the server (client-pull), for presenting them and for handling interactions with the user. The client-pull architecture is best suited for browsing scenarios with non-deterministic user interactions. In case of user interactions, the client only has to change its data request behavior, for example, it will request larger blocks of a video or send more frequent requests [RVT96, HS98].

The system architecture as displayed in figure 1 consists of the following components: The Multimedia Database Management System (MM-DBMS) is responsible for the storage and retrieval of metadata and media objects. The Continuous Long Field DataBlade (CLF DataBlade) offers continuous media support. The admission control (AC) mechanism manages the limited resources on the server. Our retrieval engine (RE) makes content-based access to videos possible. The user interface enables query formulating and presentation of results. In the following, the components are described in more detail.

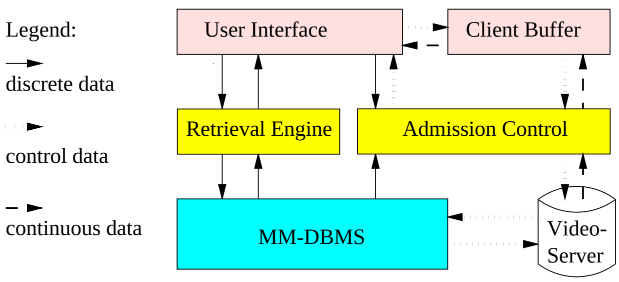

Figure 1: The Browsing System Architecture

3.1 MM-DBMS

Our browsing prototype is implemented on top of the object-relational DBMS Informix Dynamic Server (IDS). The IDS enables the integration of so-called DataBlades which provide flexible extensions of new datatypes and their corresponding functions. We use the Video Foundation DataBlade [INF97] that builds a system architecture for video data. It stores the discrete data such as text and images, and the metadata of the videos. Metadata are either manually edited, e.g., author, date, keywords, or automatically extracted image features. For performance reasons, these features are extracted a priori since feature extraction is a time consuming task relating to single frames or shots of a video. Furthermore, the Video Foundation DataBlade controls access to the external storage managers and devices that store the raw continuous data.

3.2 Continuous Long Fields (CLF) DataBlade

Since the Video Foundation DataBlade does not efficiently support continuous presentation of time-dependent media, we developed the CLF DataBlade. The CLF DataBlade, in turn, consists of two parts: the DBMS functionality extending the object-relational DBMS, and the client buffering module that requests media parts for presentation. In the following, the basic concepts of the CLF DataBlade that have been integrated as multimedia database extensions are briefly outlined. For more detailed information, we refer to [HSHA98].

Continuous Long Field. We provide a data type Continuous Long Field (CLF) as a generic representation for any kind of continuous media, such as audio, video, or animations. This datatype supports operations for editing and presentation, e.g., ”insert”, ”request”, or ”delete”. Structural metadata, such as the format, other recording parameters, or features, are stored with every CLF object [HL97].

Buffer Mechanism. We developed an algorithm called Least/Most Relevant for Presentation (L/MRP) for preloading and replacing of video objects in the client buffer. Its goal is to support a continuous presentation of a media stream with respect to VCR interactions. The client buffer manager prefetches data by means of requesting single units of a continuous medium from the server [MKK95].

Buffer–triggered Adaptation. Our session-based admission control mechanism gives no deterministic service guarantees. In case of high system load, bottlenecks may occur during a browsing session. Therefore, we use a client-based adaptation mechanism to dynamically adapt to short-term bottlenecks. It changes the presentation quality in order to reduce the data volume that needs to be delivered from the server to the client. In this way, intra-media synchronization can be maintained by reducing system load [HKR97].

The MM-DBMS extended by the CLF Datablade provides the basic functionality needed for interaction with MM data. Browsing, however, requires more structured interaction possibilities, based on user queries. Hence, we added a retrieval engine to the basic architecture, enhanced the CLF buffer strategy, and provided an adequate admission control mechanism for browsing sessions and interface components.

3.3 Retrieval Engine for Browsing

Within the retrieval engine we employ multimedia retrieval techniques to achieve content-based access by feature extraction, feature aggregation, and classification at videos based on scene granularity. Content-based access to media data is supported by conceptual queries. For example, when a user is interested in indoor shots he specifies ”artificial light” and ”artefacts”. The queries are mapped by a rule-based engine to requests expressed as constraints on feature values. A query results, in contrast to conventional queries on databases, in relevance weights - ranging between 0 and 1 - of the found stills, scenes, and videos [THE98]. Basic image features are, for example, edge analysis values, grayscale, and entropy. They relate to single frames or scenes of a video. Rules define search criteria on the feature level which can be executed on the metadata [ME97]. The relevance values corresponding to a conceptual query are calculated by means of feature aggregation on video scene granularity. The retrieval engine is described in section 4 in more detail. Since metadata management is relatively important to the browsing application, an MM-DBMS implementation is best suited for providing the needed support [CT95].

3.4 Relevance-triggered extensions of the CLF Buffer Strategy

In this section, we introduce our client buffer management approach enabling content-based browsing for videos. A typical result list to a conceptual query consists of shots from many different videos. The user's goal is to quickly find relevant information in these videos. This means that a user frequently switches between shots from different videos. The client buffer manager has to reduce the start-up latency for these presentations, since a user is not willing to wait long to start the presentation of short video shots. Due to the large size of videos it is neither possible to preload a video as a whole nor to preload all video shots of the hit list into the buffer. We assume a buffer size of 20 to 30 MB, i.e., the capacity to buffer 2 to 3 minutes of MPEG-1 video data.

The buffer manager has not only to support browsing specific requirements, but also to take into account the general presentational aspects of time-dependent media in order to support continuous, jitter, and latency free playout. Especially in distributed environments this is a formidable task.

Although many approaches for server-side buffering of multimedia data aiming to share data among multiple users exist ([NY97], [RZ95], [KRT95], [DS96], [AWY96]), only few mechanisms for client-side buffering have been proposed ([MKK95], [HGP98], [MA97], [ACS98]), all lacking satisfying solutions for the browsing requirements.

For buffering purposes, we segment videos into single parts, called "buffer units" in the following. Each buffer unit represents a logical unit of a video instance, e.g., a Group of Pictures of an MPEG stream [HL97]. These single buffer units are requested by the client buffer manager from the server. The main task for the buffer manager is to decide which buffer units should be held in the buffer and which should be replaced to support interactive browsing applications. Our buffer management algorithm preloads and replaces single objects based on their buffer value assuming that a user selects those shots with the highest value.

The main idea of our approach is to combine both presentational and retrieval aspects for the buffer value. Presentation values consider the current presentation state and likely VCR interactions. For example, the buffer units following the presentation point (that is the current buffer unit to be presented) in presentation direction have the highest value, decreasing with the distance from the presentation point [MKK95]. The *presentation relevance values pv(buffer_unit)* are used by the buffer management strategy, implemented in the CLF DataBlade with

$$pv(buffer\_unit) \in [0,1]\ \forall\ buffer\_units$$

These presentation values – a normalisation guarantees that they range between 0 and 1 – refer to the currently presented frame. We assume that they comply with the following distributions: fast forward and fast rewind are modeled as negative exponential distribution and a standard gaussian distribution is used for playback and working sets. However, this does not consider jumps to shots of other videos, as needed for the browsing application, where switches to other videos occur frequently. Furthermore, it is not considered whether the buffer units have higher or lower relevance to the content of the query. We assume a more or less rational user who prefers to select more relevant hits for presentation. The probability that less relevant scenes are inspected is dependent on the size of the result list, i.e., the larger the result list, the lower the probability that less relevant scenes are selected at all.

For all video shots of the hit list the retrieval relevance value *rv* is given by the retrieval engine. The value rv can be simply mapped to a buffer unit when we assume that a buffer unit of a video has the same granularity as a logical retrieval unit. Note that the retrieval values within a single video vary. Thus, a *retrieval relevance value* of a buffer unit is in the following formulated as

$$rv(buffer\_unit) \in [0,1]\ \forall\ buffer\_units$$

Furthermore, highly relevant scenes may be inspected several times, for example, when a user recognizes that the remaining, not selected scenes with low relevance value do not map well to the goals in mind as compared to the previously inspected scenes. Since the browsing history influences the probabilities that a hit will be selected, the relevance values have to be dynamically adapted to past user behavior.

Therefore, we use the following heuristic: the longer a video scene has not been selected and the more other hits have been selected the more likely it becomes that it will be selected again since the remaining, not selected scenes might have a low relevance to the query. During a browsing session, the retrieval relevance values for every buffer unit belonging to a hit that has previously been viewed is dynamically modified as follows:

$$\textit{rv'(n, buffer\_unit)} = \textit{rv(buffer\_unit)}\frac{2n}{n+|L|},$$

where n is the number of times a hit has not been viewed in the history h and $|L|$ the number of hits corresponding to a previous query. Note, that for $n = 0$ we get $rv' = 0$ and for $n = |L|$ we get $rv' = rv$. The factor is monotonically increasing for $n > 0$. This dynamic adaptation on past user behavior requires bookkeeping of a running session.

In our approach the buffer manager preloads and replaces buffer units based on their buffer value. That means with respect to the browsing activities of the user that a continuous presentation of the most relevant shots with respect to the browsing history is supported and thereby the system performance is improved by reducing server requests. Furthermore, explicit user wishes, represented by current user interactions, are supported by the buffer management strategy, too. The buffer value *bv* is determined by the parameters *pv(buffer unit)* and *rv'(buffer unit)*:

$$\textit{bv(buffer unit)} := \frac{(\alpha * \textit{rv'(n, buffer unit)}) + (\beta * \textit{pv(buffer unit)})}{\alpha + \beta}$$

where α and β are weighting factors. These weighting factors represent the importance of the *pv* and *rv* value and consider Quality of Service (QoS) specifications. In general, a high α value represents high QoS parameters concerning the start-up latency of a video scene request and a high β value represents high QoS parameters concerning the continuous presentation like jitter or skew. Furthermore, the specification of these weighting factors is dependent on the browsing behavior, as described as follows.

In the case of explorative browsing a ”rational” user is assumed, who will browse sequentially through the hit list, starting with the most important ones and finishing after his information needs are fulfilled. This means, assuming a rational user, that the weighting of the α value has to be high in contrast to the β value. In the case of serendipity browsing, assuming a less problem-oriented user, the behavior is not so predictable, since the user grubs through the data. He may start with less relevant hits for presentation and even skip those scenes with high relevance. Furthermore, when a user finds (more or less) random topics of interest, he will inspect them in detail. This unpredictable behavior makes it much harder to decide which buffer units have to be preloaded. For this user type, the weighting of the β value is more important since behavior is not as much relevance driven and it is assumed that the presentation duration of selected scenes may be longer than for explorative browsing. Of course, the concrete values for α and β need to be calibrated given a practical application and a given user population.

After the user sends a query to the system, the buffer manager preloads those video parts which have highest relevance to the query. The buffer space is dynamically divided for all hits. When a user starts to view video shots the presentation process influences the relevance of the buffer units. This means that buffer units preloaded before have to be replaced due to increasing buffer value of video shots which are not in the buffer.

In sum, the buffer managment strategy considers the content-based relevance values, given by the retrieval algorithm, and the current presentation process. Therefore, we increase the system's throughput by preloading those scenes that are assumed to be selected for presentation. An evaluation of this strategy will require empirical experiences on real user behavior which is difficult to simulate. Furthermore, quantitative evaluations with other client buffer strategies are not fair since they do not consider media content at all.

3.5 Admission Control Mechanism for Browsing

The admission control module is located on top of the DBMS. It is responsible for the allocation of limited storage system resources. We assume that the resources of the storage system are given, e.g., as bandwidth value, and neglect network problems [HA98]. Thus, given the delay-sensitivity of multimedia presentations, there is a limited number of browsing sessions that can be admitted for the service. The admission control module has access to metadata stored in the IDS DBMS.

Interactive browsing sessions cause varying resource demands since the user is able to select videos that are encoded in various formats, the user may use VCR functionality, or view various videos in parallel. On the other hand, users will not accept long delays between scene switches. A typical approach that targets quick response to users' requests is proposed by [Red97]. This request-scheduling approach reduces the start-up latency for urgent requests caused by interactions. Urgent requests may cause the re-scheduling of already scheduled non-urgent requests for a later service time. A stream is serviced if a schedule without resource conflicts of all involved disks can be generated[1]. The drawback of the approach is that the bandwidth required for all streams is assumed to be constant.

In our approach, the admission of clients is granted in a session-oriented manner, i.e., the client has to ask for admission for a browsing session instead of asking for admission for each single video shot to be presented during a browsing session. This stands in contrast to most approaches to admission control that are related to the admission of single media streams [ORSN96, NMW97, PC96, ZK97]. The advantage of the session-oriented approach is that the start-up latency for media switches within a session is reduced since the admission is given for the whole session. For this granularity of admission, reservations based on worst case assumptions or even stoachstically modeled resource requirements cause waste of resources.

To the best of our knowledge, the only session-based admission control mechanisms have been proposed by [ZT98] and [LP98]. The first approach is adapted for multimedia presentation plans, i.e., the presentation of multiple time-dependent objects that have to be synchronized in temporal order. The second one grants admission to web sessions that consist of a sequence of web server accesses. Its goal is to achieve a fair warranty of completion for any accepted session, which might be, for example, an e-business application, independent of session length. Neither approach reflects the browsing specific requirements as described in the introduction.

Our admission control framework is developed for highly interactive applications with variable data consumption. Its goal is to achieve high system utilization and high Quality of Service. The framework is divided into two tasks: (1) the admission of new browsing sessions based on heuristic resource prediction (2) the scheduling and adaptation of the single data requests of the admitted sessions. A detailed description of the framework with an application-independent method for resource prediction can be found in [HA98]. For browsing sessions, we make predictions for the resource demands based on the semantics of the request by modeling user behavior.

3.5.1 Modeling of User Behavior as Continuous Time Markov Chains

The browsing session itself can be viewed as a state transition system, where the user switches between states for presenting particular shots and idle states for selecting the next video shot to be presented. For resource control it is important to consider the temporal dimension, i.e., the holding time of a state. A well established model for describing such state transition systems stochastically are Continuous Time Markov Chains (CTMC) [Tij94]. A CTMC is a stochastical process that is specified by a set of states I, by mean holding times $\frac{1}{v_i}, i \in I$, and by transition probabilities $p_{i,j}$, with $i, j \in I, i \neq j$ and $\sum_{j \neq i} p_{i,j} = 1$ for all $i \in I$. When modeling browsing sessions, the presentation states, like a playback or pause, can be modeled as a state of a CTMC. The holding time in a state i represents the time until a user changes presentation mode with mean $\frac{1}{v_i}$ independently of how the system reached state i and how long it took to get there. The transition probabilities denote the probabilities that a user will switch to another presentation state.

In our approach to resource prediction, we assume that the parameters determining a CTMC, i.e., the transition probabilities and the holding times of a state, are related to the retrieval relevance values of a hit. Below, we first specify the transition probabilities for our CTMC. We assume the most simple structure of a CTMC, which is specified by one single idle state is and $|L|$ playback states with $|L|$ as number of hits. Let rv_i be the retrieval relevance value of a state i. The transition probability $p_{is,i}$ is a function of rv_i.

[1]In this approach, the notion admission control is not used since the main focus is on disk scheduling.

Since the following equation holds: $\sum_{j=1}^{|L|} p_{i,j} = 1$ we use the *normalized relevance values* $\overline{rv}_i$ given by

$$\overline{rv}_i = \frac{rv_i}{\sum_{j=1}^{|L|} rv_j}, i = 1, \ldots, |L|$$

as transition probabilities. Then $p_{is,i} = \overline{rv}_i$ and $\sum_{i=1,\ldots,|L|} p_{is,i} = 1$, whereas always $p_{i,is} = 1$. An advanced model might use a weighting function in addition, e.g., to overproportionally increase the probability that videos of higher relevance are viewed.

For the holding times of the states, we assume the following heuristic model: for short scenes, the mean of the exponentially distributed holding time is proportional to the length of the scene. There is a minimum presentation time d_{min} and the mean is limited by a maximal presentation duration $d_{min}+d_{max}$. In addition, we weight the mean by the relevance of the video, i.e. more relevant videos are viewed longer than less relevant ones. This heuristics is reflected in the following formula for the mean holding time:

$$1/v_i = d_{min} + d_{max} \frac{d(scene_i)}{d(scene_i) + d_{max}} rv_i,\ i = 1, \ldots, |L|$$

where $d(scene_i)$ is the playback duration of a scene of the hit list.

In the case of explorative browsing, a rational user who finds a large number of hits typically selects those scenes that have a high relevance with respect to the query. Such a user will spend more time with the more important scenes than with the less important ones. So, the user behavior model is based on information that is extracted from the set of browsing candidates selected by a preceding retrieval request. The hits for a conceptual user query contain shots of videos or whole videos. The hit list of a query features the scenes or videos that are found, their presentation duration, their data rates stored both as metadata in the database, and their relevance value to the query determined by the retrieval engine. Additionally, it is assumed that the data rates of all possible presentation states, e.g., ”fast forward”, are given for all hits.

In earlier work we described the structure of the CTMC used to model the browsing sessions. Furthermore, we discussed different possible browsing behavior models to illustrate how different assumptions on the nature of browsing sessions lead to structurally very different models. For more details, we refer to [AH98].

3.5.2 Admission Control Strategy

From the CTMC the resource predictions can be given in the following way. The probability for each presentation state during a browsing session can be deduced (see [Tij94]). Since for each state the required data rate is given, the expected resources required by a state i, $E(i)$ can be simply determined. The expected amount of resources E_c required for a session of client c is then

$$E_c = \sum_{i \in I} E(i).$$

where I is the set of states of a browsing session.

Let us assume the system has already admitted clients $c_1, \ldots, c_k$ and a new client c_p requests admission. Then the admission control mechanism computes the expected resource demand of the running clients $E_{c_j}, j = 1, \ldots, k$, and the expected resource demand E_{c_p} of the new client.

Now we define the admission criterion. Let s_{max} be the maximum amount of resources available. Then the admission criterion is

$$E_{c_p} + \sum_{j=1,\ldots,k} E_{c_j} < \tau * s_{max},$$

where $\tau \in$ [0,1] is a safety margin to allow small deviations from the expected resource usage.

For a large number of possible clients, such a criterion based on an estimation of the average resource usage appears to be appropriate, since deviations from the average values of single clients can be expected to compensate for statistical reasons. For a small number of clients, other admission criteria based, for example, on maximum expected resource usage or maximum expected deviation, can be considered in addition.

3.5.3 Scheduling of Data Requests

An admitted client session requests data aperiodically. These data requests are scheduled at the server by the policy Earliest Deadline First (EDF). Overloads occur when it is not possible to schedule all requests within their deadline. In this case, the request scheduler has to adapt the data requests to be served to the available resources. From the user's perspective, a longer delay when switching between videos is preferable to a quality degradation in terms of temporal or spatial Quality of Service degradation (e.g., frame skipping or lower resolution).

The problem is that the server needs to know whether a client requests a data unit that belongs to the currently presented scene or to another, highly relevant scene with high retrieval relevance. Therefore, we introduce the following protocol. A client specifies a long deadline for a request when the retrieval relevance value of a data request is high, and for buffer units with high presentation relevance it specifies short deadlines to achieve a continuous, jitter free playout. In case of underutilized periods, requests with larger deadlines are served as well. Since the retrieval relevance values only decrease during a browsing session when a shot has been presented (see section 3.4), it cannot happen that the server delivers a data unit that is not required anymore.

3.6 Browsing User Interface

The user interface enables the following user interactions: a user can specify and elaborate a conceptual query and send it to IDS DBMS. Since the result set of conceptual queries may be very large and the playback of the hits is very time consuming, the user needs some further assistance for the selection of hits. We represent for each result a set of images, representing key scenes of the video hits. Additionally the corresponding retrieval relevance values to the found shots are visualized.

A user can select retrieved clips for presentation. At this time the buffer manager on the client side is initialized. It requests those media parts with the highest relevance from the server. At the server, the first media request of a browsing session is subject to admission control. During presentation the user can exert control through VCR interactions, and can jump interactively to other shots in the hit list.

As many of the advanced algorithms outlined above hinge on the performance of the retrieval engine, we will now have a closer look at this part of our system.

4 The Video Retrieval Engine: Enabling Conceptual Queries

Many approaches to video retrieval propose an image retrieval process based on video stills ([ZLSW95], [CSM$^+$97], [CZKA96]). The authors report that good results in content analysis were found using low level image analysis of video shots. Note that this result refers to retrieval tasks only, mostly based on similarity queries. For browsing purposes, additional functionality is needed, as we will describe in this section.

Most realistic browsing scenarios involve semantic information needs, hence manual indexing prevails in most contemporary applications [LKT$^+$96]. In the context of this paper indexing means specification of content descriptors and not access structures for data. In our approach to automatic conceptual access for browsing, we attempt to employ indexing rules that capture the semantic content to a certain extent [ME97].

Our approach to enabling conceptual queries on videos is divided into the following steps: first, a video is divided into single scenes by a scene detection algorithm. The video stills indexing (analysis) system then employs a number of feature detection algorithms on selected frames. The results of these algorithms - called the feature extraction values - are used to find rules which map the values to conceptual terms. For rule generation we employ an empirical approach in which manually indexed images are used as a training set. Generated rules and extracted feature values are stored in the meta database. Note that the latter feature values are not restricted to the original sample set used for the rule generation process. Instead, feature analysis results obtained from the many times larger video collection are now used as the basis for semantic access. If the user poses a conceptual query, the retrieval engine analyses the query and maps it to a set of rules which are requested from the metadatabase. The rules are interpreted by an appropiate rule interpreter yielding specification of features extraction values to be searched for. If feature values matching the constraints can be found, the associated video parts are retrieved. The result is given back to the user as a ranked list. In the following, these steps are described in more detail.

4.1 The indexing step

In the indexing process, our prototype –called HERMES/AVIA[2]– starts a shot-detection method (see [Ste96]). The system now chooses the frame in the middle of each shot as representative of the video document, calculates the feature-extraction values for these frames and stores this metadata in the database.

As a part of our experiment, a number of feature-extraction and comparison algorithms were implemented [Eve96] using the PBM (portable bit map) collection of image processing software[3]. Since texture-based classifications are very effective (up to 100% correct classifications, if they are applied carefully [PKL93]), the pbm texture module [HSD73] was selected as a promising tool. The problem with the feature-extraction values is the processing time to calculate those values (approx 10 sec for all feature-extraction values per image). To solve this problem, we decided to calculate these values only for some representative frames from a video document.

For the rule generation, we use the following empirical approach: the starting point is a set of images (video stills) manually classified by different indexers and used as a training set. The set of index terms consists of a number of domain-independent categories, e.g., content of an image (animate, artefact), contour (sharp, blurry), source of light (natural, artificial), or type of image (photo, drawing, graphic, cartoon, close-up). The set of video stills were divided in two disjunct sets, a training set and a testing set. We used the training set for the rule generation and the testing set to validate the rules (see section 4.4).

The association between feature values and indexing terms can now be accomplished using algorithmic statistical methods, ranging from exhaustive exploration to complex stochastic computation. Which method is applicable depends on the degree of aggregation applied to the original feature values. We start with n-dimensional vectors $\bar{x}$, containing the results from n different analysis methods, applied to the still. In the next step, the feature-extraction values may be aggregated to dynamically built constraints, e.g., ranges, or linear combinations of the feature values. If we regard ranges of scalar aggregated feature values, we can derive plausible rules to describe the content of pictures on a general level, by analysing the feature-extraction values of the manually classified images by a α-Quantile analysis, as described in [ME97], [THE98]. In a number of cases, however, no useful association pattern could be found due to the fact that the aggregation process was too coarse-grained. In this situation, it is useful to examine the original data, i.e. the feature vectors. In a first exploration study [Eve96] we employed a brute force exhaustive search. The results being promising, we changed to a more efficient method.

In our next experiments, we used the Quadratic Classification Function (QCF) (see [Bor89]) to calculate the probability that an image matches a classification item. The QCF gives a measure for the distance of the feature extraction values of an image and the mean values of a set of manually classified images.

Let $x_i, i = 1 \ldots p$, be feature-extraction values of an image m, and $\bar{x}_j, j = 1 \ldots p$ be the median of all x_i from the images in a training set, manually classified with the same classification item. Then, a distance vector d is defined as

[2] Analysis of Video Information Approach
[3] Future development will take into account the Excalibur-Datablade.

$$d = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_k \\ \vdots \\ d_p \end{pmatrix} := \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \vdots \\ \bar{x}_j \\ \vdots \\ \bar{x}_p \end{pmatrix} - \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_p \end{pmatrix}.$$

This means that an element d_k of the vector d shows the difference between the k-th feature-extraction value of a given image and the median of the corresponding feature-extraction values of the training set.

To calculate the probability that an image with value vector x belongs to a classification item C_j with the vector of median values $\bar{x}$, we need the variance-covariance matrix COV_j for the values of the C_j-classified images. Therefore, we define the matrix D_j as

$$D_j = X_j' \cdot X_j - \overline{X}_j' \cdot \overline{X}_j$$

where x_{im} is the i-th feature-extraction value of image m from the learning set, manually classified with the same classification item C_j and $\overline{x}_{im}$ is the mean from all feature-extraction values i from all images from the learning set, manually classified as C_j. COV_j is defined as

$$COV_j = D_j \cdot \frac{1}{n_j}$$

where n_j is the number of images, classified as C_j. Using COV_j and the distance vector d, we now define

$$\chi_j^2 = d' \cdot COV_j^{-1} \cdot d + \ln |COV_j|.$$

χ_j^2 is a distance measure between the feature-extraction values of the image and the c_j classified images of the training set. The probability $p(c_j|x)$ that an image with the feature-extraction values x belongs to a classification item c_j can be calculated as

$$p(c_j|x) = \frac{e^{-(\chi_j^2/2)}}{\sum_k e^{-(\chi_k^2/2)}}.$$

where $\sum_k e^{-(\chi_k^2/2)}$ is the sum over all classification items of the classification topic C. This probability value can be used to order the images in descending order. In the retrieval process, we can derive the relevance status values for an item by computing this probability.

Thus, a foundation is laid for a conceptual way of accessing MM items necessary for browsing. The probability calculation described above can be costly when done at query time. However, a good approximation can be devised which makes use of the observation that when some feature value vector x_i is within a certain section of the feature space, the corresponding picture will have been classified by some term C_j in many cases. Hence, it seems to be useful to encode the corresponding sections and index terms as part of indexing rules. We then can employ a rule-based approach to the mapping of conceptual query terms on certain constraints on feature representations that are stored in the DBMS. Executing the rules during the retrieval process automatically yields proof trees which can be used as appropriate access structures for conceptual retrieval. Hence, the remaining task is to define the logical retrieval engine that could exploit the rules. We employ an abductive reasoning system which automatically maps the user's query to the feature constraints.

4.2 Abductive Reasoning: Rule-based Query Processing

A user's query is a description of what the user is looking for and needs to be translated into feature constraints which may match database entries directly. To deal with this problem properly, we distinguish between the query as an intensional (or: conceptual) representation of the user's information need, and the extensional model (i.e., instances retrieved from the database) of an inferred query "interpretation." Abduction can informally be explained as a way to derive appropriate premises (i.e., feature vectors) which imply a known consequence, i.e., the query.

In principle, the abduction-based approach can be regarded as a more flexible alternative to restricted Bayesian networks (as applied, e.g., in the INQUERY system, [TC90, TC91] to implement a probabilistic inference engine for the purpose of IR. Abductive reasoning allows us to treat a query more flexibly by exploiting the association rules to "expand" the query. The abductive "expansion" process replaces concepts of the original query, thus generating possible "interpretations" of the user's information need. Since the abduction yields all possible ways to interpret the user's query, we have at hand a means to increase both the recall (by using all appropriate expansions), and the precision (by suppressing inappropriate ones) during the retrieval interaction.

A user request Query is a (usually incomplete) description of a concept the user is looking for. Let Query be an existential quantified sentence combining elements of $\mathcal{T}$. Then we can use the abductive reasoning process for information retrieval as follows: "Definition of the theory: A logical theory $\mathcal{T}$ is defined over a language $\mathcal{L}$ of well formed formulae, built from variables, constants and predicates. The domain of constants is defined by the set of atomic values in the underlying data management system(s). Predicates are defined recursively to be either a direct accessible function, provided by the data repository, or to be any computable combination of basic predicates. [...] During an abductive inference process, eventually the process needs to stop and assume a predicate. Typically, abductive systems define only a subset of the theory to be abducible. We follow this principle by defining: The set $\mathcal{A}$ of abducible sentences is the collection of atomic predicates. Each element of $\mathcal{A}$ corresponds directly to an information item via a computational access method within the database of the system." ([TM96], p. 191)

We define an abductive information retrieval proof to be complete if it finds all sets of possible solutions (i.e. a set of sets of hypotheses). This applies naturally to the use of abduction as a retrieval technique: the inherent ambiguity of queries is reflected in a straightforward manner in the information retrieval process by offering mutual distinct hypotheses to the user. This property of abductive retrieval makes it especially useful for dealing with large result sets. The abduction process yields proof trees for every atomic part of the query. The leaves of these trees are abducible, i.e. they can be executed directly to obtain those parts of the document base which are relevant for the concepts expressed in the leaves. Thus, we can partition the result set according to these concepts, and can define hyperlinks according to the proof structure, e.g. by linking those (basic) concepts to the intermediate nodes in the proof tree, which are visualized as system-generated pages [TM96].

In the following, a concrete dialogue example is discussed which explains the user's option for a conceptual search, and the system's reaction to such a submitted intensional query.

4.3 Conceptual User Queries: the Retrieval Step

To start her exploration of the digital library, a user submits a conceptual query specifying some desired categories, e.g., "natural light", "living objects", and "artefacts" if she is interested in street scenes with people, cars etc. Of course, restrictions on manually edited meta data, e.g., location or time, can also be formulated. A rule-based interpretation maps the conceptual query onto a request expressed in constraints on feature values, based on pre-calculated weights of association rules between concepts and feature patterns. The rules define search criteria on the feature level which can be executed in the meta data DBMS. The weights are used - in combination with the degree of matching - to compute relevance weights - ranging between 0 and 1 - of the found stills, shots, and videos.

If a user requests some video clips, the prototype calculates the required value range by analysing the request, determining the feature constraints from the metadata. At this point, HERMES/AVIA provides a

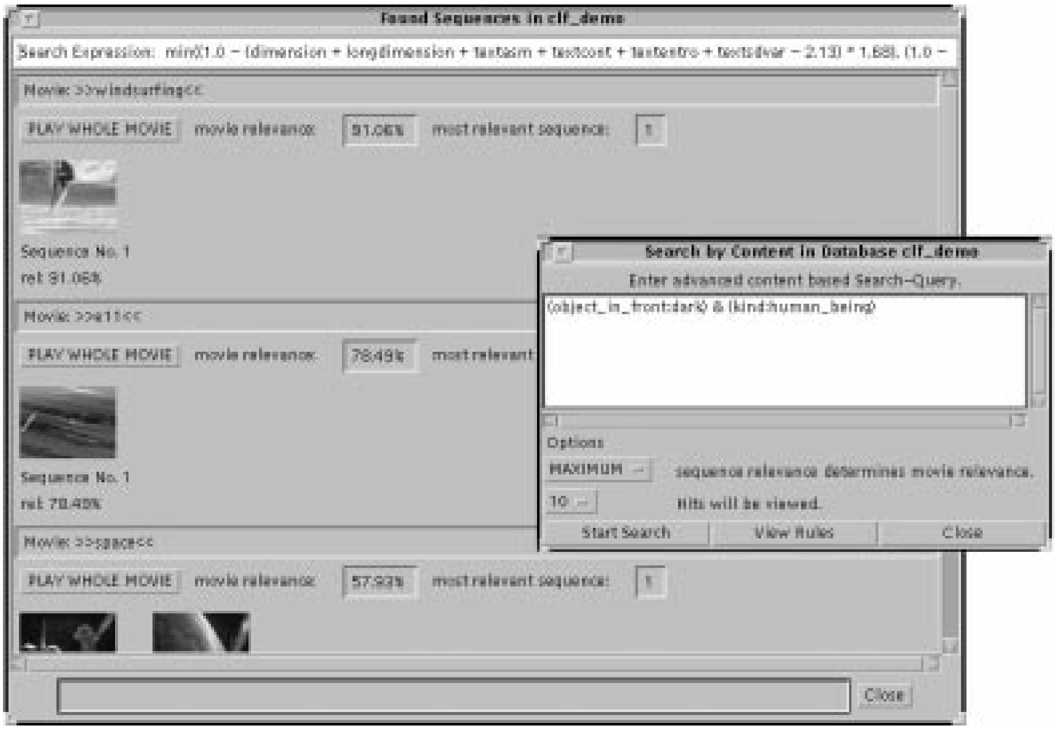

Figure 2: The Query and the Result Window

set of relevance values for each video document in the database.

The user can decide to view the result list sorted by shot or document relevance. As [CWT97] point out, the users prefer to have a selection of keyframes rather than the document title as the result presentation. As a good compromise between bandwith restrictions and user needs, they suggest showing one video still per shot.

If the user wants to rank the results by shot relevance, the documents are sorted by the maximum shot relevance of each document. The ranked result list contains entries providing links to entire videos in combination with a thumbnail-link for each shot in the document. The user can now reformulate her request, if necessary, or view a shot or an entire video document.

4.4 Rule Verification

In our verification, we compared the automatic indexing results of our retrieval engine with the outcome of manual indexing. In the first step, a set of 300 video stills was manually indexed and feature vectors for these stills were computed. Then, the set was divided into two disjunct sets with the same number of elements. The first set was used as a training set to calculate the rules, and the second set as a testing set to estimate the quality of the discovered rules. Table 1 shows the average precision for the QCF and the brute force algorithm for several classification categories.

The brute force algorithm shows a better behavior than the QCF formula. The main advantage of the QCF is that the calculation of the rules is faster than by the brute force algorithm (90 times) with a tolerable loss in accuracy. As far as the results are comparable, both methods perform better than the neural networks used in [RF97], given the fact that Rowe and Frew worked with a rather homogeneous picture collection. For the future we plan to include a relevance feedback mechanism. Thus, the retrieval quality for both approaches can be improved. Furthermore we plan further statistical testing to improve the quality of the QCF.

	QCF	brute force
light	75.83%	84.00%
dimension	73.28%	95.53%
contour of main obj.	54.00%	69.92%
content	44.96%	59.12%

Table 1: Average precision for the QCF formula and the brute force algorithm

5 Conclusion

In this article, we described a browsing system architecture that enables content-based access to videos in browsing applications by feature extraction and rule-based feature aggregation. Using a small set of annotated stills as training set, a statistical analysis associates patterns in feature vectors with semantic index terms. When a query containing semantic constraints is submitted, these association rules are employed to derive constraints on feature vectors, which allow the identification of probably relevant portions of the videos.

The system is built on top of a commercial OR-DBMS. Our extensions provide continuous media playback and resource prediction for highly interactive video browsing applications, both using the retrieval relevance values of a conceptual query result in terms of content-based client caching and admission control.

In this work, we proposed a mechanism supporting interactive multimedia browsing applications by exploiting information about the expected browsing behavior of the user which is estimated on the basis of a rationalistic exploration strategy for retrieval results. The algorithm determining the media objects to be kept in the cache takes into account both the relevance estimation of the retrieval engine and the presentation process, i.e., playout. If the user deviates from the anticipated course of interaction, the system can react in a reasonable way by adjusting its estimates.

Future work will include applications of the browsing system in different contexts, differentiation by user group, users' tasks, and video collections, in order to establish characteristic patterns of usage with appropriate calibrations of system parameters. Since the major information source has been evaluation studies both on admission control and caching of real user behavior, real browsing applications need to be analyzed in detail.

References

[ACS98] Prathima Agrawal, Jyh-Cheng Chen, and Cormac J. Sreenan. Use of statistical methods to reduce delays for media playback buffering. In *IEEE Multimedia Systems*, pages 259–263, June/July 1998.

[AH98] Karl Aberer and Silvia Hollfelder. Resource prediction and admission control for interactive video browsing scenarios using application semantics. GMD Technical Report 40, GMD, Sankt Augustin, Germany, September 1998. to be published in Proc. of Int. Conf. on Data Semantics - 8 (DS-8), Semantic Issues in Multimedia Systems, IFIP TC-2 Working Conference, Rotorua, New Zealand, 5-8 January 1999.

[AWY96] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. On optimal batching policies for video-on-demand storage servers. IBM Research Report RC 20400, T. J. Watson Research Center, March 1996.

[Bor89] Jürgen Bortz. *Statistik für Sozialwissenschaftler (in German)*. Springer, 1989.

[CBG98] C. Colombo, A. Del Bimbo, and I. Genovesi. Interactive image retrieval by color distributions. In *IEEE Multimedia Systems*, pages 255–258, 1998.

[CCBP98] M. Caliani, C. Colombo, A. Del Bimbo, and P. Pala. Computer analysis of tv spots: The semiotics perspective. In *IEEE Multimedia Systems*, pages 170–179, 1998.

[CCM+97] Shih-Fu Chang, William Chen, Horace Meng, Hari Sundaram, and Di Zhong. Videoq: An automated content-based video search system using visual cues. In *ACM Multimedia*, 1997.

[CSM+97] S-F Chang, J.R Smith, H.J. Meng, H. Wang, and D. Zhong. Finding images/video in large archives. *D-Lib Magazine*, February 1997.

[CT95] Stavros Christodoulakis and Peter Triantafillou. Research and development issues for large-scale multimedia information systems. *ACM Computing Surveys*, Dec 1995.

[CW88] J.F. Cove and B.C. Walsh. Online text retrieval via browsing. *Information Processing and Management*, 24(1):31–37, 1988.

[CWT97] M.G. Christel, D.B. Winkler, and C.R. Taylor. Multimedia abstraction for a digital video library. In *ACM Digital Libraries '97*, pages 21–29, Philadelphia, PA, 1997.

[CZKA96] Y.-L. Chang, W. Zeng, I. Kamel, and R. Alonso. Integrated image and speech analysis for content-based video indexing. In *Proceedings of ACM MM*, 1996.

[DS96] Asit Dan and D. Sitaram. Dynamic batching policies for an on-demand video server. *ACM Multimedia Systems*, 4(3):112–121, June 1996.

[Eve96] Andre Everts. PiClasso – picture classification operators. Master's thesis, TU Darmstadt, Department of Computer Science, 1996. in German.

[FBGW98] Jonathan Foote, John Boreczky, Andreas Girgensohn, and Lynn Wilcox. An intelligent media browser using automatic multimodal analysis. In *Proc. of ACM Multimedia*, 1998.

[FSN+97] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J.Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanke. Query by image and video content: The QBIC system. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval*, pages 7–22. AAAI Press/MIT Press, Menlo Park et al, 1997.

[GT98] William I. Grosky and Yi Tao. Multimedia data mining and its implication for query processing. In Roland R. Wagner, editor, *Proc. of the 9th Int. Workshop on Database and Expert Systems DEXA'98*, pages 95–100, August 1998.

[HA98] Silvia Hollfelder and Karl Aberer. An admission control framework for applications with variable consumption rates in client-pull architectures. In Asuman Dogac Sushil Jajodia, M. Tamer Oezsu, editor, *Proceedings of 4th Int. Workshop Multimedia Information Systems MIS'98, Advances in Multimedia Information Systems*, pages 82–97. Springer Lecture Notes in Computer Science, September 1998. also published as GMD Technical Report, Sankt Augustin, Nr. 8, April 1998.

[HGP98] P. Halvorsen, V. Goebel, and T. Plagemann. Q-L/MRP: A buffer management mechanism for QoS support in a Multimedia DBMS. In *Proc. of IEEE Int. Workshop on Multimedia DBMS*, August 1998.

[HKR97] Silvia Hollfelder, Achim Kraiss, and Thomas C. Rakow. A client-controlled adaptation framework for multimedia database systems. In *Proc. of European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97)*, Darmstadt, Germany, September 1997. Springer Lecture Notes in Computer Science.

[HL97] Silvia Hollfelder and Hyo-Jun Lee. Data abstractions for multimedia database systems. GMD Technical Report 1075, GMD, Sankt Augustin, Germany, May 1997.

[HS98] Tobias Helbig and Oliver Schreyer. Protocol for browsing in continuous data for cooperative multi-server and multi-client applications. In Thomas Plagemann and Vera Goebel, editors, *Proc. of 5th Int. Workshop Interactive Distributed Multimedia Systems and Telecommunication Services IDMS'98*, pages 231–236. Springer Lecture Notes in Computer Science, September 1998.

[HSD73] R. M. Haralick, K. Shanmugan, and I. Dinstein. Textual features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[HSHA98] Silvia Hollfelder, Florian Schmidt, Matthias Hemmje, and Karl Aberer. Transparent integration of continuous media support into a Multimedia DBMS. In Tamer Ozsu, Asuman Dogac, and Ozgur Ulusoy, editors, *Proceedings of the 3rd Biennial World Conference on Integrated Design and Process Technology (IDPT), Issues and Applications of Database Technologgy (IADT)*, volume 2, July 1998. Enlarged version published as GMD Technical Report, Nr. 1104, Sankt Augustin, Germany, December. 1997.

[IKN+98] Hiroshi Ishikawa, Kazumi Kubota, Yasuo Noguchi, Koki Kato, Miyuki Ono, Naomi Yoshizawa, and Akiko Kanaya. A document warehouse: A multimedia database approach. In Roland R. Wagner, editor, *Proc. of the 9th Int. Workshop on Database and Expert Systems DEXA'98*, pages 90–94, August 1998.

[INF97] INFORMIX Press. *Video Foundation DataBlade Module User's Guide, Version 1.1*, June 1997. Version 1.1.

[KRT95] M. Kamath, K. Ramamritham, and D. Towsley. Continuous media sharing in multimedia database systems. In *Proc. of 4th Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, pages 79–86, April 1995.

[KSN97] T. Kanade, S. Satoh, and Y. Nakamura. Accessing video contents: Cooperative approach between image and natural language processing. In *International Symposium on Research, Development and Practice in Digital Libraries*, pages 143–150, Tsukuba, Ibaraki, Japan, 1997.

[LAC+95] T. D. C. Little, G. Ahanger, H.-J. Chen, R. J. Folz, J. F. Gibbon, A. Krishnamurthy, P. Lumbda, M. Ramanathan, and Dinesh Venkatesh. Selection and dissemination of digital video via the Virtual Video Browser. *Multimedia Tools and Applications*, 1(2):149–172, 1995.

[LKT+96] Barbara Lutes, Said Kutschekmanesch, Ulrich Thiel, Catherine Berrut, Yves Chiaramella, Franck Fourel, H. Haddad, and Phillipe Mulhem. *Study on Non-Textbased Information Retrieval - State of the Art*. EU, Study ELPUB 106, 1996.

[LP98] Cherkasova Ludmila and Peter Phaal. Session based admission control: A mechanism for improving the performance of an overloaded web server. HP Labs Technical Reports, External HPL-98-119, 980612, Hewlett Packard, June 1998.

[MA97] Defeng Ma and Gustavo Alonso. Distributed client caching for multimedia data. In *Proceedings of Third Int. Workshop on Multimedia Information Systems (MIS)*, pages 115–120, September 1997.

[ME97] Adrian Müller and Andre Everts. Interactive image retrieval by means of abductive inference. In *RIAO 97 Conference Proceedings – Computer-Assisted Information Searching on Internet*, pages 450–466, June 1997.

[MKK95] Frank Moser, Achim Kraiss, and Wolfgang Klas. L/MRP: a buffer management strategy for interactive continuous data flows in a multimedia dbms. In *Proc. Int. Conf. of Very Large Data Bases (VLDB)*, pages 275–286, Sept 1995.

[NMW97] Guido Nerjes, Peter Muth, and Gerhard Weikum. Stochastic performance guarantees for mixed workloads in a multimedia information system. In *Proc. of the IEEE International Workshop on Research Issues in Data Engineering (RIDE'97)*, Birmingham, UK, April 1997.

[NY97] R. Ng and T. Yang. Maximizing buffer and disk utilization for news-on-demand. In *Proc. of the IEEE VLDB Conference*, pages 451–462, Satiago, Chile, 1997.

[ORSN96] Banu Özden, Rajeev Rastogi, Avi Silberschatz, and P. S. Narayanan. The fellini multimedia storage server. In S. M. Chung, editor, *Multimedia Information Storage and Management*. Kluwer Academic Publishers, 1996.

[PC96] Seungyup Paek and Shih-Fu Chang. Video server retrieval scheduling for variable bit rate scalable video. In *Proc. of the IEEE International Conference on Multimedia Computing and Systems*, pages 108–112, 1996.

[PKL93] R. W. Picard, T. Kabir, and F. Liu. Real-time recognition with the entire brodatz texture database. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 1993.

[PSAP98] Dulce Ponceleon, Savitha Srinivasan, Arnon Amir, and Dragutin Petkovic. Key to effective video retrieval: Effective cataloging and browsing. In *Proc. of the ACM Multimedia*, September 1998.

[Red97] Narasimha Reddy. Improving latency in interactive video server. In *Proc. of SPIE Multimedia Computing and Networking Conference*, pages 108–112, Feb 1997.

[RF97] Neil C. Rowe and Brian Frew. Automatic classification of objects in captioned depictive photographs for retrieval. In Mark T. Maybury, editor, *Intelligent Multimedia Retrieval*, pages 65–79. AAAI Press/The MIT Press, 1997.

[RVT96] Siram S. Roa, Harrick M. Vin, and Asis Tarafdar. Comparative evaluation of server-push and client-pull architectures for multimedia servers. In *Proc. of Nossdav 96*, pages 45–48, 1996.

[RZ95] D. Rotem and J.L. Zhao. Buffer management for video database systems. In *Proc. of the 11th Int. Conf. on Data Engineering*, pages 439–448, March 1995.

[SC96] J. R. Smith and S.-F. Chang. Searching for images and videos on the world-wide web. Technical Report 459-96-25, Columbia University, Center for Telecommunications Research, New York, August 1996.

[SK98] Tatsuya Shibata and Toshikazu Kato. Modeling of subjective interpretation for street landscape image. In Erich Schweighofer Gerald Quirchmayr and Trevor J. M. Bench-Capon, editors, *Proc. of the 9th Int. Conference on Database and Expert Systems DEXA'98*, pages 95–100, August 1998.

[Ste96] Arnd Steinmetz. DiVidEd – A Distributed Video Production System. In *VISUAL'96 Information Systems, VISUAL'96 Conference Proceedings*, 1996.

[TC90] Howard Turtle and Bruce Croft. Inference network for document retrieval. In J.-L. Vidick, editor, *Proceedings of the 13th Conference on Research & Development in Information Retrieval*, pages 1–24, 1990.

[TC91] Howard Turtle and Bruce Croft. Efficient probabilistic inference for text retrieval. In *Proceedings of the RIAO'91, Barcelona, Spain*, pages 644–661, April 1991.

[THE98] Ulrich Thiel, Silvia Hollfelder, and Andre Everts. Multimedia management and query processing issues in distributed digital libraries: A HERMES perspective. In Roland R. Wagner, editor, *Proc. of the 9th Int. Workshop on Database and Expert Systems DEXA'98*, pages 84–89, August 1998.

[Tij94] Henk C. Tijms. *Stochastic Models. An Algorithmic Approach.* Wiley series in probability and mathematical statistics. Wiley, 1994.

[TM96] Ulrich Thiel and Adrian Müller. Why was this item retrieved? new ways to explore retrieval results. In M. Agosti and A. Smeaton, editors, *Information Retrieval and Hypertext*, pages 181–201. Boston: Kluwer, 1996.

[Web] The virtual video browser (VVB). http://hulk.bu.edu/projects/vvb_demo.html.

[WKSS96] H.D. Wactlar, T. Kanade, M.A. Smith, and S.M. Stevens. Intelligent access to digital video: The imformedia project. *IEEE Computer*, 29(5):46–52, 1996.

[ZK97] Hui Zhang and Edward W. Knightly. RED-VBR: a renegotiation-based approach to support delay-sensitive VBR video. *Multimedia Systems*, pages 167–176, May 1997.

[ZLSW95] H.J. Zhang, C.Y. Low, S. W. Smoliar, and J.H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. In *Proceedings of ACM MM*, pages 15–24, 1995.

[ZT98] Wei Zhao and Satish K. Tripathi. A resource reservation scheme for synchronized distributed multimedia sessions. *Multimedia Tools and Applications*, 7(1/2):133–146, July 1998.