To appear on Image Communication Journal,
Special Issue on Video on Demand and Interactive TV, 1996.

Development of Columbia's Video on Demand Testbed

Shih-Fu Chang, Alexandros Eleftheriadis, and Dimitris Anastassiou

Department of Electrical Engineering
and Center for Telecommunications Research
Columbia University
New York, NY 10027

Keywords — Video on Demand, Interactive Video, Video Interoperability, Video Servers, MPEG-2 Video over ATM

Abstract

This paper reports on our progress in developing an advanced video-on-demand (VOD) testbed, which will accommodate various multimedia research and applications such as Electronic News on Demand, Columbia's Video Course Network, and Digital Libraries. Two different prototypes have been completed. The first generation of the testbed was based on a constant-bit-rate (CBR) video server utilizing Ethernet delivery. Contents were encoded and stored as MPEG-2 audio/video elementary streams. Software encoders/decoders were used in content generation and playback. The second generation of the testbed was enhanced with the capability of transmitting true MPEG-2 transport streams over the campus ATM network as well as the wide-area NYNET ATM network. A real-time video pump and a distributed application control protocol (MPEG-2's DSM-CC) have been incorporated. Hardware decoders and set-tops are being incorporated to test wide-area video interoperability. Our VOD testbed also provides an advanced platform for implementing proof-of-concept prototypes of related research. Our current research focus covers video transmission with heterogeneous quality-of-service (QoS) provision, video storage architecture design, content-based video indexing and browsing, multi-resolution (MR) video coding, efficient manipulation of compressed video, and advanced user interfaces. An important aim is to enhance interoperability. Accommodation of practical multimedia applications and interoperability testing with external VOD systems are currently being undertaken.

1. Introduction

At Columbia University, we are developing a VOD testbed with advanced features of video storage, coding, manipulation, transmission, and retrieval. The main objective is to use this testbed as a platform for state-of-the-art multimedia research and application development. Among the potential applications are Columbia's Electronic News System, Digital Libraries, Interactive Video Courses on Demand, and other interactive multimedia applications.

Designing a full-function VOD system for general multimedia applications requires extensive interdisciplinary knowledge and skills. Several research groups have reported progress in various aspects. System-level studies are presented in [1]. Multi-resolution representations for image databases were studied in [15]. Innovative methods for indexing/searching images by image contents were proposed in [5, 8, 16]. Dedicated storage architectures for real-time multi-access have been studied in [10, 11, 12, 13, 17]. Systematic approaches to the design of video servers (VS) are being undertaken in [3, 18, 20] as well. In addition, many field trials of VOD services using proprietary high-performance VS technologies have made news headlines recently. Lastly, a major international forum called DAVIC has been established to

come up with timely recommendations for critical protocols and interfaces for achieving *interoperability* between various audio-visual applications [2].

We have developed two different generations of a VOD testbed. The first generation was based on a constant-bit-rate (CBR) video server communicating over Ethernet. Contents were encoded and stored as MPEG-2 audio/video elementary streams. Software encoders/decoders were used in content generation and playback. The second generation of the testbed was enhanced with the capability of transmitting true MPEG-2 transport streams over the campus ATM network as well as the wide-area NYNET ATM network. A real-time video pump and a distributed application control protocol (MPEG-2's DSM-CC), have been incorporated. Hardware decoders and set-tops are being incorporated to test wide-area video interoperability. Starting in the summer of 1995, a series of interoperability experiments will be conducted, both within our lab and with external participants. The results of these interoperability tests will be reported to the VOD research community and international standardization forums such as DAVIC.

In developing the VOD testbed, we also look beyond today's technology and standards. We are investigating critical research areas and use the testbed as a prototyping platform. Some current research endeavors related to the VOD system are:

- optimization techniques for video storage architecture design, allowing multi-user real-time access with heterogeneous QoS (e.g., utilization efficiency, interactivity latency, and buffer size requirements);
- multi-resolution image/video coding and dynamic rate shaping for provision of multiple QoS;
- efficient interactive manipulations of compressed bitstreams;
- innovative methods for video feature extraction and indexing, allowing advanced mechanisms of video search and browsing;
- interactive navigation tools and user interfaces allowing effective search and browsing of visual data;
- Quality of Service (QoS) negotiation and guarantees in heterogeneous network environments;
- synchronization between different types of media (video, audio, captions, etc.).

All the above issues are being addressed in our VOD testbed. Figure 1 illustrates the overall spectrum of various cross-disciplinary issues. There are great synergies among various subareas. One of our primary goals is to explore maximum synergies by taking a systematic approach and pursuing joint optimization of various cross-disciplinary issues.

This paper provides a general overview of the testbed and describes in further detail the major research areas listed above. Section 2 describes the system architecture and major components. Research issues and progress in various technical areas are discussed in Section 3, while some concluding remarks are given in Section 4.

2. Testbed Architecture and Components

In order to investigate complete end-to-end system solutions, our VOD testbed consists of all critical components required in video/multimedia on demand applications — *content, server, network, client, and user control*. The envisioned applications, as discussed briefly above, include Interactive News on Demand and Video Courses on Demand. The former stresses the importance of real-time interactivity and multi-user access efficiency. The research issues driven by these applications are discussed later on in Section 3. In the following subsections, we describe each of the individual testbed components. A simplified system architecture diagram is shown in Figure 2.

2.1 Visual Content Coding

All visual material will be stored on the video server in a compressed form (e.g., MPEG-2 transport streams and MPEG-1 system streams). In order to reduce the time spent in content preparation, it is

desirable to have fast encoding, packetization, and multiplexing tools. For live video production, real-time encoding facilities are required. At this stage, MPEG-2 hardware encoders and transport stream multiplexers are available at high costs and with restricted functionalities. Therefore, we have followed a software-based approach and try to enhance the speed performance by utilizing the real-time multiprocessor capabilities of an SGI Onyx graphics supercomputer. Our software encoder and transport stream multiplexer currently run at a speeds lower than real time. The rapid advancement of today's computing technology, however, will hopefully resolve this problem soon. We should note that, for stored-video applications such as video on demand, an off-line non-real-time video coding facility is generally sufficient. For MPEG-1 video compression, a real-time hardware-based system hosted on a PC is utilized, generating MPEG-1 system streams (including both video and audio).

We have developed flexible software facilities for digitizing and encoding video from various sources (e.g., live camera, VCR, and LD player) and various domains (e.g., recorded lecture video, general movies, and test video from the public domain). Columbia's MPEG-2 software encoder (a full implementation, including all scalability profiles) is utilized to compress the video sequences, generate the video elementary streams, packetize the elementary streams, and multiplex multiple packetized elementary streams (PES) to MPEG-2 transport streams. Timing information is captured by time stamps at different layers of the compressed bit streams, such as decoding time stamps, presentation time stamps, and program clock references, in order to maintain synchronization at different levels. Multimedia data such as text, graphics, and audio may be multiplexed into the transport streams as well, as described in the MPEG-2 system standard.

The bit rate allocation of each video stream depends on the type of the video source. In addition to the constant bit rate MPEG-2 main profile video streams, we also use hybrid scalable MPEG-2 coding to generate scalable video streams. We combine the spatial scalability and SNR scalability modes to produce three different layers of video. The base layer has a small spatial size and is suitable for video browsing and preview functionalities. The first enhancement layer increases the spatial size and keeps the video signal quality (i.e., SNR values) at a consistent level. The second enhancement layer improves the signal quality as well as the spatial resolution. The experimental bit rate allocations selected are shown in Table 1. Based on our preliminary and subjective evaluations, the base and first enhancement layers provide a subjective quality comparable to VHS video quality, while the highest layer provides a subjective quality comparable to that of a LaserDisc. The decision of whether to use scalable video coding or not should be based on the application types, system capacity, and encoder/decoder capabilities. There are currently no commercial scalable MPEG-2 hardware decoders available; scalability, however, is desirable in heterogeneous environments including different networks (wired and wireless), different client processing/display capabilities, and different user preferences. The use of scalable video coding also has significant implications on the design of the video server, as will be discussed in more detail in Section 3.1.

2.2 Video Server

Optimization of the overall VOD system performance requires a balanced system approach in exploring all the critical design factors for the video server. Fundamentally, it's a real-time data pumping problem — how to store massive video streams in a hierarchical storage unit (including memory, disk, tape, and tertiary storage), move them through the I/O interface and memory, and then pump them to the network interface. Careful data layout within the storage hierarchy, efficient real-time scheduling, and admission control mechanisms are all required in optimizing the system performance. We are investigating all these research issues in designing our video server.

Our server platform currently includes an SGI Onyx multiprocessor graphics supercomputer as a super-server (with 6 CPUs and 1GB of memory), and clusters of workstations as distributed servers. The Onyx super-server is equipped with the high-end computing power and 3D-graphics capabilities that are

needed in many interactive multimedia applications and real-time video manipulations. Dedicated disk array secondary storage is connected to the server, while local storage systems are available on distributed workstations. The server's communication interface is enhanced by the connection to an ATM LAN, which in turn is connected to an external ATM WAN.

Software support on the video server includes a video pump for real-time CBR video stream retrieval, and a high-level control/management entity. The former is responsible for retrieving the video stream from the storage unit to the network and guarantee real-time performance. Our goal is to design a generic object-based video pump in which different network interfaces (e.g., TCP/UDP/IP and ATM) and video types (e.g., MPEG-1 and MPEG-2) can be accommodated. We take advantage of the multiprocessor architecture and the real-time process scheduling control of the Onyx machine to achieve real-time guarantees at a certain temporal granularity (e.g., 50 μs). For typical MPEG-2 video rates and ATM Service Data Unit (SDU) sizes, the temporal resolution at this level is sufficient. Currently, we have designed one type of video pump software for MPEG-2 streams that uses the ATM AAL5 protocol. The packet mapping process (e.g., the 5/8 mapping specified by the ATM Forum) is also included in the real-time scheduling loop to guarantee the real-time performance of data delivery from the server to the network.

The high-level control and management entity is basically concerned with content management, database support, directory information, and interactive control of MPEG compressed streams. We have implemented the MPEG-2 DSM-CC user-to-user control protocol [34] to provide a subset of the above functions. In order to ensure maximum interoperability and take advantage of recent advances in distributed computing technology, the remote procedure call facility used in our DSM-CC implementation is based on OMG CORBA (Common Object Request Broker Architecture [35]). By adopting a standard-conforming interface specification mechanism, we hope to maintain maximum interoperability across different network and computing platforms.

Video data are striped across the disk array connected to the video server. The specific layout schemes have a significant effect on the overall system performance, including parameters such as utilization, access latency, and buffer size requirement. The are discussed in more detail in Section 3.1.

2.3 Network

The network infrastructure of our VOD testbed includes ATM and Internet as the campus core network. The campus-wide ATM network will be expanded to connect several other geographical distributed campuses (e.g., medical and geoscience) in the near future. This ATM network is also connected to the NYNET wide area ATM network, which in turn provides high-speed connections to external VOD testbeds. Different local access networks (e.g., Hybrid Fiber Coax—HFC) will be provided by external VOD testbeds. Users on the campus will access the video servers directly through the ATM network or via Ethernet. The same ATM core network will be used to connect to wireless access networks in the next stage of the testbed's development.

A suite of client-server communication protocols and interfaces has been developed. Downstream channels towards the clients use the TCP/IP protocol stack for transmitting delay-insensitive data and UDP/IP for isochronous video streams over Internet. ATM/AAL protocols are used for delivering real-time video over the ATM network. As discussed above, for MPEG-2 video over ATM, we have implemented an adaptation program for mapping MPEG-2 transport packets to AAL5-type ATM cells. The reverse channel uses TCP/IP for delivering back to the server control data, such as browsing, retrieval, and query commands. For users on the ATM network, IP packets can be overlaid on the ATM cells as well. So far, there are limited signaling functions available from the existing ATM switches (e.g., SPANS on the FORE Systems ATM switches). We anticipate incorporating more advanced signaling software in the near future.

2.4 Client

The client-side platforms may be composed of workstations, PC, or stand-alone set-tops. Many hardware MPEG-2 transport-level decoders will soon be available, while MPEG-1 decoders are already available for PCs and consumer electronics products. ATM adaptors on workstations and PC's are also on the brink of wide proliferation. Due to the highly asymmetric complexity in MPEG coding and multiplexing, the client side capability will not be a bottleneck in the end-to-end real-time data pumping chain. There are still some design issues, however, such as the trade-off between decoder complexity and scalable video coding, and the trade-off between the client-side buffer space and stream playback interactivity. These issues need to be investigated from a global, system-level perspective.

We have implemented software-based MPEG-2 decoder and playback routines with VCR interactive control functions on several workstation platforms. Hardware decoders are available for attachment to these general-purpose platforms; we are currently using hardware-based MPEG-1 decoders on PCs, and also testing a PC-based MPEG-2 transport stream decoder implementation. Dedicated hardware set-tops with specific network interface modules are currently provided by external partners. In order to maintain maximal flexibility and extensibility, we focus on the general-purpose computing platforms at this stage. The requirement of real-time operation systems on the client side may not be necessary. As mentioned above, using the hardware transport stream decoders will reduce the processing load of the client computer and therefore may avoid the need of real-time operation systems.

Multimedia synchronization between different media streams (e.g., video and audio) is important on the client side, especially when different media streams are transported separately. Usually, due to the high bit rate and high traffic burstiness of the video data, video will suffer more severe delay and delay jitter. Audio-video synchronization may be accomplished by two different approaches. The default mode is to playback audio in real time while decoding video (by software) with a best effort approach, in other words skipping video frames (e.g., B frames of MPEG) whenever video playback falls behind. The other approach tries to play back every video frame as fast as possible, while making the best effort in keeping the audio track synchronized with video. Typically, this requires intelligent algorithms to adjust the sampling rate of the audio signal without losing its pitch information [21].

In order for users to browse through the video title collections more efficiently, a scene-based video browsing graphical interface is also provided on the client side. Representative thumbnail images of prominent scene shots in a video title can be downloaded for pre-view and quick browsing before users engage in reading the entire stream. Other advanced client-side user interface issues are discussed in Section 3.5.

2.5 User Control

The scenario of an actual VOD application session typically consists of the user issuing various control commands from time to time. At the beginning, users will attach to a service gateway (e.g., a Level 1 gateway) to find the right destination server containing the desired programs and applications. Through user-network and user-user interfaces, users may connect to a specific server, list/open/retrieve one or multiple multimedia streams, and then issue various interactive commands to control the playback of the requested streams. In our testbed, the service gateway is emulated by a simple process running locally or on a remote machine. User-to-user application control is provided by the DSM-CC user-to-user primitives. User-to-network control primitives are limited to the preliminary functions provided by the existing system, and will be enhanced with either the DSM-CC protocol or other alternatives.

2.6 Current Status and Video Interoperability Tests

At this stage, we have developed two generations of the VOD testbed. The first generation of Columbia's VOD testbed has been completed and functioning since the summer of 1994. It's a client-server platform based on a CBR video server and TCP/UDP/IP protocols over Ethernet. The video servers achieve CBR video delivery by using receiver feedback flow control. The performance, therefore, is limited by the client side software decoder. Video and audio contents are stored as MPEG-2 video/audio elementary streams with scalable coding options. The client-side user interface supports database query, scene-based video browsing, interactive playback, and QoS selection panels. Figure 3 shows a snapshot of the client-side user interface.

Many advanced components mentioned above have been incorporated into the second generation of our VOD testbed, which is expected to be fully functioning in the summer of 1995. MPEG-2 audio and video streams are multiplexed into MPEG-2 transport streams (TS). Real-time video pumps without dependence on client feedback are used to deliver the real-time MPEG-2 TS, mapped to ATM AAL5-type cells. CORBA-based DSM-CC user-to-user application control protocols have been completed on client-server platforms as well.

Currently, we are conducting a series of video interoperability tests both individually within our testbed and together with external VOD testbeds. The MPEG-2 audio/video multiplexed transport streams will be sent over the wide-area ATM experimental network to the remote clients, which may be equipped with dedicated set-tops or PC's with hardware decoders. We hope to test the interoperability at various levels and various interfaces in these in-action wide-area ATM/MPEG-2 video experiments. The same testbed will be used to support several different research activities, which are described in the following sections.

3. Research Issues

In addition to accommodating development of advanced multimedia applications, Columbia's VOD testbed also serves as an experimental environment for implementing proof-of-concept prototypes for advanced engineering research. The availability of an end-to-end comprehensive testbed is actually very critical to many research projects which have cross-disciplinary nature. For example, optimization of the video server storage unit cannot be isolated from research on video transmission over networks. There are strong interactions between the server scheduling/buffering mechanisms and network transport mechanisms. These interactions are best understood in an actual experimental testbed covering end-to-end components. We discuss several major research areas highly related to the VOD systems in the following sections.

3.1 Optimization of the Video Storage System

Retrieval of video sequences from the video server has strict constraints on delay and delay jitter. Storage systems (e.g., disks, tapes, and tertiary units) usually have physical performance limits, such as disk head seeking time, and I/O read/write throughput limit. Traditional file system layout strategies (e.g., the various UNIX file systems) are not suitable for real-time video retrieval. Also, in advanced VOD applications, different users may require different QoS (e.g. different rates, resolutions, display sizes). Users may request interactive functions (e.g., VCR playback control) as well. Furthermore, each video stream may be encoded with scalable video coding techniques, which may generate CBR or VBR substreams at the different layers. All these requirements make the video server design a very challenging research issue. Most existing commercial VOD trials provide video servers with fixed video QoS and only limited interactive functionalities.

A conceptual model for mapping multi-resolution (MR) video data to the distributed storage space is shown in Figure 4. Essentially, it is an optimization problem in mapping a multi-dimensional data space

to a multi-dimensional storage space. The constraints here are the users' heterogeneous QoS requirements and the physical performance limitations of the storage systems. The performance factors are multi-fold — the number of simultaneous video streams supported, the hardware cost (e.g., the buffer size), the interactivity functions (e.g., VCR control functions), and the access latency (e.g., the response time for pause and resume). There are interesting trade-off relationships among these performance factors. The design space is also complicated, including data layout schemes, real-time scheduling policies, and admission control mechanisms. In order to find the optimal operating point in this multi-dimensional design space, we are currently using the following techniques to investigate optimal video storage architectures.

- **Constrained data placement and grouping** — In our previous work on single-disk partitioning [13], we have proposed to partition the disk space into P partitions and place video elements in a particular fashion so that every n-*th* block is stored in the (n mod P)-*th* partition. By doing so, the maximum disk seek delay between two consecutive reads is reduced by a factor of P. In addition, the interframe distance is carefully designed to be constant. Therefore, the buffer size requirement on the user side is reduced from the double buffers to the single buffer. Figure 5(a) illustrates this partitioning technique for a single disk. This technique can also be applied to individual disks in the multi-disk case.
- **Disk head control optimization** — Typical disk head control techniques, such as SCAN and CSCAN (Circular SCAN), can be used to increase the efficiency of the disk head movement. Instead of requiring a maximum disk seek delay time, these techniques reduce the average seek delay by picking up all requested video elements in one disk seeking direction. We have combined the CSCAN technique with our previous disk partitioning technique to enhance the disk utilization efficiency.
- **Interleaving, scattering, and grouping** — By interleaving independent rate components of the same video across parallel disks, we may increase the maximum throughput serving the high-end users who need to retrieve all data elements of the same visual item. For example, different rate components (L, M, and H) are interleaved across three disks in Figure 4. In addition, scattering the same rate component to different disks can increase the level of parallelism. For example, the low (L) rate components of the same video sequence are scattered across three different disks in Figure 4. This will allow multiple low-end users to access the same video simultaneously. Finally, by determining the optimal group size of each rate component, the disk head movement efficiency can be increased as well. Figure 5(b) shows one possible mapping which interleaves different frames of the same video into different disks. The mapping of Figure 5(c) further interleaves different rate components of the same frames to different disks [7].
- **Load balancing** — In practical video coding schemes, the bitrate of different components are usually non-uniform. For example, the energy and bitrates for different wavelet subbands are not uniform. In typical coding standards such as MPEG-1 and MPEG-2, bitrates of different frames are also quite different. We need to take this into account so that loads are evenly distributed among different disks. The ideal case is that all disks are utilized 100% during each cycle.
- **Statistical Multiplexing** — If the video sequences are variable bit rate (VBR) streams, the requested data amount during each fixed time period is variable. Reserving peak-rate bandwidth for each VBR stream is not cost-effective. Reserving constant average bandwidth for each stream and using the output buffer before the network interface to avoid data starvation will create a long start-up latency. One alternative is to use statistical multiplexing and to tolerate a certain amount of data loss at the video server. This corresponds to the constant-time-length model described in [12], in which a fixed number of frames of data (but variable size) are read from the disk. Hence, a number of trade-offs are simultaneously at play here: the more output buffer is provided, the less statistical multiplexing gain is required and the lower data loss rate is. Also, the better VBR traffic description functions we can get, the higher system utilization we can expect from the admission control mechanism. All these complicated design issues become even more challenging when we introduce scalable video coding tech-

niques. For example, in the scalable MPEG-2 coding method mentioned in Section 2.1, each stream actually consists of multiple layers, each of which by itself is a VBR stream. Currently, we are investigating the optimal scheduling policy and admission control mechanisms taking into account of the impact of the scalable coding technique.

- **Fault-tolerant design** — Unlike numerical or control data, visual data can tolerate loss to some extent, depending on the acceptable subjective quality. For example, some users may be willing to accept loss of few non-critical frames if the server is congested. We must take into account this fault-tolerant model in order to optimize the overall system performance.

One of our objectives in designing the video server is to provide heterogeneous QoS. MPEG-2's scalable coding features provides heterogeneous QoS of both signal quality and spatial resolution. Different temporal resolution can be easily achieved by skipping less significant frames (e.g., B and P frames) in MPEG-2 compressed bitstreams. In [7], we proposed a flexible platform to provide heterogeneous QoS of interactivity latency. By using variable segmentation levels within each disk retrieval block and careful data striping strategies on disk arrays, we can adjust the overall interactivity latency based on the actual application requirements. Furthermore, we derive a new technique which segments different video layers at different levels and use the progressive display approach to reduce the interactive response latency. Figure 6 shows the scenario for the progressive display, in which the small-size base layer is displayed immediately following the request, the intermediate layer with the spatial resolution enhancement is displayed a few cycles later, while finally the full-resolution layer with signal quality enhancement is displayed. This technique can be shown to provide a nice compromise between several different performance factors, such as utilization efficiency and interactivity latency [7].

3.2 Scalable Video Coding and Dynamic Rate Shaping

Multi-resolution video coding is adopted in order to accommodate heterogeneous QoS requirements (e.g., different channel bandwidth or different display resolutions). As mentioned above, MPEG-2 scalability can provide multiple QoS for up to 3 layers [24] (by combining different scalability options such as spatial, temporal, and SNR), which may be sufficient for some applications. In a heterogeneous environment, however, that requires the support of different types of network links, user terminals, and user preferences, a large number of different bit rate levels are needed. In addition, network capabilities may be time varying (this is the case, for example, in best-effort networks); this variation may follow a prescribed statistical behavior or it may be completely random and unpredictable. A very simple example of heterogeneity is accessing a stored video signal compressed at 4 Mbps through a 2 Mbps channel. Due to the potentially large number of different bandwidths, a multi-resolution coding approach with as many layers would be inefficient.

One promising technique being developed at Columbia University for the creation of an arbitrary number of bit rates from pre-encoded video bitstreams is "Dynamic Rate Shaping" (DRS) [36, 37]. Figure 7 depicts the differences between DRS and the more well-known rate control applied at the encoder. Typically, rate control algorithms are used to optimize the rate-distortion performance given the constraints on rate and latency (B and D in Figure 7). In contrast to rate control, DRS is a process typically residing between the encoder and the network transport interface; it manipulates the encoded bitstream to obtain a new bitstream that conforms to the desired bit rate requirements. The objective is to minimize the conversion distortion. DRS can be implemented either in the server or in some third-party locations which conceivably can retrieve compressed video from various distributed servers and add a specific service value (here rate conversion). A key difference between DRS and rate control is that the former requires no interaction with the encoder, and hence is applicable even in stored-video applications. In essence, DRS frees the system designer from the limitations of purely CBR or VBR video, and provides a totally new viewpoint in terms of how video could be handled in multimedia networks.

We have developed a family of DRS algorithms for motion-compensated block-based transform coders (including MPEG-1, MPEG-2, and H.261), and studied optimality conditions and implementation complexity considerations. The basic mechanisms with which rate manipulation can be achieved in such coding schemes are: 1) re-quantization of DCT coefficients, and 2) selective transmission of DCT coefficients (similar to zonal sampling [25]). The former approach leads to schemes similar (but not identical) to re-coding, while the latter results in schemes that form natural generalizations of the data partitioning scalability mode of MPEG-2 [14, 24]. We call this latter approach Constrained DRS (CDRS), since an additional structural constraint is imposed on the DRS algorithm [36]. By allowing transmission of an arbitrary subset of the DCT coefficients (not necessarily a contiguous one), we obtain the so-called Unconstrained DRS scheme [37]. Experimental results have shown that unconstrained DRS performs only marginally better than the constrained approach, at a significant cost in complexity.

A fundamental issue in DRS is the recursive nature of the encoding process that precedes it; since DRS modifies the prediction error communicated to the decoder, care must be exercised for the control of error accumulation phenomena. Our algorithms operate in an operational rate-distortion framework, and their objective is to minimize the conversion error while staying within the prescribed rate constraints. Fast algorithms (both recursive and memoryless) based on Lagrangian optimization have been developed with very good results. It has been shown that, across a wide range of rate conversion ratios, DRS outperforms re-coding. A critical feature of DRS that we have discovered is that the memoryless algorithms (i.e., those that do not take into account the accumulated error through motion compensation) perform almost identically (within few tenths of a dB) to the optimal ones. In addition, this property holds across the range of rate conversion rations. This allows the use of relatively simple algorithms with minimal, if any, additional loss of quality. Since the memoryless DRS algorithms have complexity less than that of a decoder, they are amenable to real-time software implementation on general purpose computers. Such a real-time implementation is currently underway and will be integrated to the Video Pump's output module. Note that, to accommodate time varying channels, indication of the bandwidth availability must be propagated to higher layers from the transport interface.

3.3 Efficient Manipulation of Compressed Video

Rate conversion is one of many functions that may be needed to be applied on the retrieved compressed video before it is transmitted from the VOD servers to end users. Among others are format conversion (video transcoding), image enhancement, zooming, geometrical transformation (such as scaling, rotation, shifting, shearing), and multi-object compositing. The stored image formats may not be desirable for transmission or display. Users may want to see the image objects from different view angles and at different scales. Some image processing functions (such as enhancement, and contour extraction) may be requested when images are displayed. Users may want to subscribe to multiple video streams and composite them into a single displayable stream.

We are exploring innovative algorithms to achieve these functions directly in the compressed domain in order to minimize the incurred computational cost. This basically reflects one of our key design principles mentioned earlier — *exploring maximum synergies among various areas in the VOD systems (in this case, compression and manipulation).*

Figure 8 shows two different approaches to performing the required manipulations on retrieved compressed images. The traditional approach converts the compressed images back to the uncompressed domain, performs the desirable manipulations, and then re-compresses them. The alternative, as we proposed, is to design equivalent manipulation algorithms in the compressed domain. The benefits of manipulating images in the compressed domain are twofold. First, the data rate in the compressed domain is much less than that in the uncompressed domain and thus the required computational cost can be reduced. Sec-

ond, the decoding and encoding processes for converting the images back and forth between the compressed format and uncompressed format can be avoided, as shown in Figure 8.

In particular, in our prior work we have proposed a set of algorithms for manipulating images in any orthogonal transform domain [22,26], including DCT (Discrete Cosine Transform), DFT (Discrete Fourier Transform), and orthogonal DWT (Discrete Wavelet Transform). Functions like geometrical transformations, resolution conversion, image filtering, image multiplication/convolution, and multiple image compositing can be implemented directly in the compressed domain.

Taking compositing as an example, given the transform coefficients (e.g., DCT coefficients) of the input images, we can directly calculate the transform coefficients of the output images directly in the transform domain. If the transform coefficients of the input images have been quantized (as in typical practical coding systems), a great number of small coefficients may be truncated to zeros. Therefore, the computational complexity associated with the compressed-domain operations will be greatly reduced. An example scenario shown in Figure 9 has three input images, each of which needs to be scaled down with different ratios and translated to arbitrary positions. Using the typical DCT with quantization (without motion compensation), input images have a great percentage (sometimes up to 90%) of transform coefficients truncated to zero. The net computations can be reduced by about 65% if we use the proposed transform compressed-domain approach, compared to the traditional approach. In general, the actual computational speedup by using the compressed-domain approach depends on the compression rate of the input video. Some manipulation functions (such as block-based operations) benefit more from the transform-domain approach than others due to their compatibility with the block structure of typical transform algorithms.

One specific manipulation function, *linear correlation* is worth further discussion. Linear correlation is typically used as one form of implementation of *image template matching*. In addition to the low-level signal features (such as texture, shape, color) which will be described in the next section, direct image pattern matching may provide useful means of indexing and access to images and video in the context of VOD, especially when specific image keys are given. As a special case of linear filtering, correlation (and thus image pattern matching) can be performed directly in the compressed domain. Compared to the traditional image pattern matching in the uncompressed domain, this compressed-domain matching may greatly reduce the computational complexity.

For video compression standards based on motion compensation and block-based transform coding (such as MPEG and H.261), we have derived one transform-domain conversion technique to convert the motion compensated video back to the transform domain, in which the transform-domain manipulation algorithms can be applied. The computational complexity of the transform-domain approach depends on the motion vector distribution of the input video, as well as their compression rate.

3.4 Fast Video Browsing and Content-Based Visual Query

Another important research focus of our VOD project is to explore new ways of image/video indexing and query by visual contents. Most existing approaches to image indexing and retrieval use the textual keyword, which naturally lends itself to the usage of conventional textual-based query [27,28]. Search and retrieval are performed on the keyword records and the associated images are retrieved after the matches are found. Some image databases provide enhancement by supporting query by pictorial examples of pre-determined visual objects, such as mechanic design diagrams, electronic schema, and office designs. In some cases, semantic-level descriptions (such as objects in the picture, relationships among objects, and actions associated with objects) are provided by users and used as index of the visual data.

All the above approaches rely on some form of manual input from users. It will become difficult to use this manual approach to indexing huge amounts of visual data in a video server. Also, it is difficult to obtain consistent and complete subjective descriptions of visual data. To overcome this problem, we are

investigating innovative approaches to automatic indexing and searching of visual data by generic signal features such as object shape, texture, color, motion, video scene, etc. We consider this content-based approach complimentary with existing user-assisted keyword-based and semantic-level approaches. Only by providing a rich multiplicity of interfaces towards indexing and retrieving visual data can users efficiently search through thousands to millions of pictures in the server.

Figure 10 shows the role of the content-based indexing/query engine in the VOD server. Users interact with the server through the Query/Presentation interface, which produces the textual or visual keys to the Query Engine to search for the intended objects. The visual features and textual indexes are stored in the Meta-data Index File which is kept separate from the actual compressed image/video data. Logical links and pointers are created to bind related objects. The query results will be forwarded to the Retrieval Scheduler which in turn schedules the actual retrieval of the returned data, through the management of the Storage Controller.

In retrieving the low-level signal features for image content indexing, we again apply the principle of maximizing synergies between different subareas of VOD designs (in this case, compression and feature extraction). In particular, we extract signal features directly from the compressed domain if possible, without the need to perform independent image processing algorithms for feature extraction. Currently, we have developed a set of algorithms to extract low-level signal features (e.g., texture, edge, video scene cuts, and camera operations) allowing indexing and accessing video by signal features [8, 9, 33]. Consider that massive video/image data have been or will be stored in various compressed forms. Compressed-domain feature extraction provides a cost-effective approach to exploring this new direction of indexing visual materials.

To apply the above feature-based indexing and query technique to video, we take a divide-and-conquer approach which segments the entire video sequence into individual scene cuts. We assume each video scene contains consistent contents during the entire scene period. Therefore, we can index each individual scene with its representative features, such as object shape, texture, motion, and relations among objects. As mentioned above, we derive new techniques for performing video scene segmentation directly in the compressed domain, with minimal decoding of the compressed bitstreams (i.e., MPEG-2 encoded bitstreams) [9]. Our approach is to use the distribution of motion vectors and DCT coefficients of the motion compensated residual errors in MPEG-2 compressed bitstreams as cues in detecting dissimilarity between image frames. Compared to the traditional approach, the detection is performed with only a partial decoding of the compressed bit stream. A full decode of the compressed bitstream is not necessary and therefore computation time can be saved. We have tested our algorithms on synthetic sequences and real sequences. We have been able to detect most scene changes successfully, including the *dissolve* technique (i.e., fade in/fade out) which is often used in older movies. Figure 11 shows the graphical user interface for scene-based video indexing and browsing.

We are currently extending our techniques in two directions. First, we explore the optical flow techniques to detect important camera operations such as zooming, panning, and tilting. By extracting and compensating these global motions, we are developing techniques for extracting video objects directly from the compressed streams. These techniques are very useful for scene-based video indexing and video object manipulation. Secondly, we have developed techniques for users to randomly cut and paste the MPEG-2 compressed streams in the compressed domain. Imagine that in some applications users may request more complicated functions than simply passively viewing the video display with or without VCR control functions. Users may want to manipulate multiple compressed streams and produce new video streams. Editing compressed video streams in the compressed domain is more challenging than that in the uncompressed pixel domain. Efficient techniques need to be derived to handle rate control issues and maintain the integrity of the compressed streams.

The scene-based video indexing technique can be used to construct a hierarchical representation for general video data. Once scene cut and camera operation detection is complete, different scenes at different times in the same video can be further classified into different groups. For example, a video sequence may be segmented to multiple groups based on motion, scene length, physical objects in the scenes (e.g., scenes with human figures), and other semantic-level criteria. This semantics-based segmentation of video data will provide an effective indexing scheme and allow users to search large video servers more efficiently. As the video servers store more and more video programs, these types of video indexing and access mechanisms will become more critical.

3.5 Interactive User Interface

In a video server containing huge image/video collections, an effective interactive user interface must provide the capability for users to browse through retrieved images and video sequences at different scales (spatially or temporally). This multi-scale capability can be achieved by using multi-resolution/scalable video coding algorithms as mentioned earlier. Through the interactive tools, users can also manipulate the retrieved images to meet their specific needs. Envisioned image manipulations include geometrical transformation (zooming, rotation, etc.), image quality enhancement (for rare preserved document images or medical images), halftoning (for converting continuous-tone images to bi-level ones for display or printing purposes), color space conversion (to accommodate displays with different color depth), and compositing of multiple image objects, among others.

For video, a *scene browser* is useful in summarizing the contents of the retrieved video sequence, as shown in Figure 11. For example, users can quickly browse through the representative image frames of different scenes contained in each video sequence.

We are also working on interactive tools for users to select arbitrary image segments from retrieved displayed images, specify and extract interesting features (e.g. texture, color, shape), composite features from multiple image segments (e.g. color of object A combined with texture of object B), to reformulate a new image query. Users should also be allowed to combine visual features with text keywords, which can be provided by the user's input or from explanatory text documents associated with retrieved images. For example, a user might be interested in all text and image materials related to a specific topic. Therefore, keywords describing this topic should spawn searches through the text and image sections of the archive. Through a common interface to both text and image type searches, the user should be allowed the freedom to choose the data-type domains to be searched. The retrieval mechanism must then integrate data from different domains, into a single set of query results.

Figure 4 shows a suite of graphic user interface in our VOD testbed. Through the *QoS setup panel*, users can retrieve the same video at different spatial resolutions based on our MPEG-2 spatial scalability codec implementation. Through the *interactive video playback interface*, users can execute VCR control functions during a video playback session. Traditional *bibliographic search interface* is also provided, in conjunction with the *texture-based image query interface*.

4. Conclusions

With the campus-wide multimedia applications and video interoperability tests as the initial driving forces, we are building a VOD testbed with advanced capabilities of audiovisual representation/storage/retrieval/transmission. The testbed serves as an advanced prototyping platform for both engineering research and practical applications. It also facilitates the interaction between engineering researchers and application practitioners.

Our VOD testbed has evolved in two stages. The first generation has all critical components for compressed video/audio material production and CBR video delivery over Ethernet. The second genera-

tion is enhanced with true MPEG-2 transport stream support, transmission over ATM, and use of hardware decoders/set-tops. A series of video interoperability test is currently underway to test transmission of MPEG-2 audio/video over a wide-area ATM network. We are testing the interoperability and validating the protocols/interfaces across various system components from end to end in a VOD application. The first series of tests are scheduled in the summer of 1995.

On the research side, our focus at this point covers innovative video server design, content-based video access, heterogeneous QoS provision through multi-resolution coding and dynamic rate shaping algorithms, innovative compressed-domain video manipulation, and packet video transmission over ATM networks.

Many practical applications are being developed in this VOD testbed, including Columbia's Interactive Electronic News Experiment. Through the close interaction of research undertaking and application development, we expect that this testbed development effort will help to achieve significant technological advancements in the general areas of video on demand and future interactive video.

5. Acknowledgments

Many people have been actively involved in Columbia's VOD testbed development. Jianhao Meng implements the MPEG-2 software viewer and designs Columbia's Video Parsing/Editing System (CPVE). John Smith is working on Content-Based Visual Query (CBVQ) and is the author of the VOD browsing graphical user interface and bibliographic meta-data database. Seungyup Paek is responsible for the CBR video server in the first generation testbed. Hari Kalva implements the DSM-CC software using the CORBA environment. Javier Zamora is working on packet video over ATM networks and is the author of the MPEG-2/ATM AAL5 adaptation program. Steve Jacobs is responsible for the real-time Video Pump software. Xi Chen implements Columbia's MPEG-2 audio/video elementary stream packetizer and transport stream multiplexer. Sassan Pejhan and Kand Ly have also contributed to the development of our first generation prototype.

6. References

[1] *Special Issue on Video on Demand*, IEEE Communications Magazine, May 1994, Vol. 32, No. 5.

[2] The Digital Audio-Visual Council (DAVIC) Opening Forum, San Jose, CA, June 1-3, 1994.

[3] S.-F. Chang, D. Anastassiou, and C. Judice, "Proposal for a Reference Testbed for DAVIC Interoperability Demonstration," contribution to Digital Audio/Visual International Council (DAVIC), August, 1994.

[4] Dimitris Anastassiou, "Current Status of the MPEG-4 Standardization Effort," Invited Paper, SPIE Visual Comm. and Image Proc., Chicago, Sept. 1994

[5] Wayne Niblack, et al., "The OBIC Project: Querying Images by Content Using Color, Texture, and Shape," IBM RJ 9203 (81511), Feb, 1993.

[6] S. F. Chang, D. Anastassiou, A. Eleftheriadis, J. Meng, S. Paek, S. Pejhan, and J. R. Smith, "Development of Advanced Image/Video Servers in a Video on Demand Testbed", IEEE Visual Signal Processing & Communications Workshop, Sept. 1994, New Brunswick, NJ.

[7] S. Paek, P. Bocheck, and S.-F. Chang, "Scalable MPEG2 Video Servers with Heterogeneous QoS on Parallel Disk Arrays," Proc. of IEEE Workshop on Network and Operating System Support for Digital Audio and Video, '95, Durham, New Hampshire, April 1995.

[8] J.R. Smith and S.-F. Chang, "Quad-Tree Segmentation for Texture-Based Image Query," Proc. of ACM 2nd Multimedia Conference, San Francisco, Oct. 1994.

[9] J. Meng, Y. Juan and S.-F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," SPIE Electronic Imaging, Digital Image Compression, Feb. 1994, San Jose.

[10] H. M. Vin, P. V. Rangan, "Designing a Multi-User HDTV Storage Server", IEEE Journal on Selected Areas in Communications vol. 11, No. 1, January 1993

[11] K. Keeton and R. Katz, “The Evaluation of Video Layout Strategies on a High-Bandwidth File Server,” Intern. Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, England, UK, Nov. 1993.

[12] E. Chang and A. Zakhor, “Scalable Video Data Placement on Parallel Disk Arrays,” SPIE Symposium on Imaging Technology, San Jose, 1994.

[13] P. Bocheck, H. Meadows, and S.-F. Chang, “A Disk Partitioning Technique for Reducing Multimedia Access Delay,” Proceedings of Intern. Conference on Distributed Multimedia Systems and Applications, Honolulu, Aug., 1994.

[14] A. Eleftheriadis, and D. Anastassiou, “Optimal Data Partitioning of MPEG-2 Coded Video, “to appear on IEEE 1st Intern. Conf. on Image Processing, 1994.

[15] J. White and A. Klinger, “Image Coding in Visual Databases”, in Visual Database Systems, II, E. Knuth and L.M. Wegner (Editors), Elsevier Science Publishers B.V., 1992.

[16] A. Nagasaka and Y. Tanaka, “Automatic Video Indexing and Full-Video Search for Object Appearances” In E. Knuth and L. M. Wegner, editors, *Video Database Systems, II*, Elsevier Science Publishers B.V., North-Holland, 1992, pp. 113 - 127.

[17] Ming-Syan Chen, Dilip D. Kandlur, and Philip S. Yu, “Support for Fully Interactive Playout in A Disk-Array-Based Video Server”, to appear on ACM 2nd Multimedia Conference, San Francisco, Oct. 1994.

[18] C. Federighi and L.A. Rowe, “The Design and Implementation of the UCB Distributed Video On Demand System,” Proc. of IS&T/SPIE 1994 Int’l Symp. on Elec. Imaging: Science and Technology, San Jose, CA, February 1994

[19] S.W. Smoliar and H. Zhang, “Content-Based Video Indexing and Retrieval,” IEEE Multimedia Magazine, Vol.1, No.2, Summer 1994.

[20] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, W.W. Reeve, D.H. Schelleng, and D. Venkatesh, “A Digital On-Demand Video Service Supporting Content-Based Queries,” ACM 1st Multimedia Conference, Anaheim, CA, Aug. 1993.

[21] S. E. Youngberg, “Rate/pitch modification of speech using the constant Q transform,” IEEE, ICASSP ‘79, Washington, DC, April 1979.

[22] S.-F. Chang and D.G. Messerschmitt, “Manipulation and Compositing of MC-DCT Compressed Video,” IEEE Journal of Selected Areas in Communications, Special Issue on Intelligent Signal Processing, 1994.

[23] Tihao Chiang and Dimitris Anastassiou, “Hierarchical Coding of Digital Television,” IEEE Communications Magazine, May 1994

[24] Generic Coding of Moving Pictures and Associated Audio (MPEG-2), ITU-T Draft Recommendation H.262, ISO/IEC 13818 Draft International Standard, 1994.

[25] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[26] Shih-Fu Chang, “New Algorithms for Processing Images in the Transform Compressed Domain” to appear on SPIE Visual Communications and Image Processing ‘95, (also in CU/CTR Technical Report 390-94-37).

[27] T.L. Kunii, *Visual Database Systems*, Elsevier Science Publishers, 1989.

[28] E. Knuth and L.M. Wegner, *Visual Database Systems II*, Elsevier Science Publishers, 1992.

[29] T. Caelli and D. Reye, “On the Classification of Image Regions by Color, Texture, and Shape,” Pattern Recognition, Vol. 26, No. 4, pp.461-470, 1993.

[30] R. W. Picard and T. Kabir, “The Brodatz Texture Database: Characterization by Shift-Invariant Principal Components”, *M.I.T. Media Laboratory Perceptual Group Report*, 1993, No. 203, pp. 1--13.

[31] E. Binaghi, I. Gagliardi, R. Schettini, “Indexing and Fuzzy Logic-Based Retrieval of Color Images”, in Visual Database Systems II, Elsevier Science Publishers, 1992.

[32] M. Swain and D. Ballard, “Color Indexing,” International Journal of Computer Vision, &:1, pp. 11--32, 1991.

[33] S.-F. Chang and J. R. Smith, “Extracting Multi-Dimensional Signal Features for Content-Based Visual Query,” to appear on SPIE Visual Communications and Image Processing ‘95, Taipei, 1995.

[34] MPEG-2 Systems Working Draft and Working Draft Extension, ISO/IEC/JTC1/SC29/WG11.

[35] The Common Object Request Broker: Architecture and Specification. Object Management Group document 93-12-43.

[36] A. Eleftheriadis and D. Anastassiou, “Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video,” Proceedings, 5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV-95), Durham, New Hampshire, April 1995, pp. 95-106.

[37] A. Eleftheriadis and D. Anastassiou, “Constrained and General Dynamic Rate Shaping of Compressed Digital Video,” Proceedings, 2nd IEEE International Conference on Image Processing (ICIP-95), Washington, DC, October 1995 (to appear).

Figure and Table Captions

FIGURE 1. R & D Issues in Advanced VOD Systems.

FIGURE 2. The system architecture of Columbia's VOD testbed.

FIGURE 3. Current user interface of Columbia University's VOD testbed. This snapshot shows the video scene browser, the MPEG-2 playback interface, the QoS setup panel, and the bibliographic search database.

FIGURE 4. Mapping multiple components of each image frame of each video sequence to multiple disks. L,M,H represents different rate components, I,B,P are different frame types of MPEG.

FIGURE 5. (a) A disk partitioning technique [13] for reducing the single-disk access delay. (b) & (c) Mapping multi-rate multi-stream video data to multiple disks. In (b), each group of frames (GOF) is completely mapped to a single disk; while in (c) each GOF is further separated into multiple segments, which are striped across multiple disks.

FIGURE 6. Use progressive display and scalable MPEG-2 coding to improve utilization/ interactivity performance.

FIGURE 7. Comparison between rate shaping and rate control. B and D are constraints on bandwidth and delay respectively.

FIGURE 8. Processing retrieved video in (a) the uncompressed domain (b) the compressed domain.

FIGURE 9. One example scenario of manipulating retrieved compressed images. Multiple retrieved images are scaled, translated, and composited into a single displayable stream.

FIGURE 10. A system model of the content-based visual query engine in the VOD server.5

FIGURE 11. The Video Browser User Interface with Scene Change Detection

TABLE 1. Bitrate Allocations in Our Scalable Video Coding Scheme