Hierarchical Coding of Images and Continuous Media for Transmission over Packet-Switching Networks

George C. Polyzos

Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114

UCSD Technical Report CS93-273
January 1993

Abstract

Hierarchical coding techniques split signals into components of varying importance. The aggregation of these components reconstructs the original data, but subsets of them can also provide various degrees of approximation to the original signal. Hierarchical coding is very important for the efficient use of high-speed packet-switching networks. The main issue for such network architectures is the significant congestion control problems that can arise due to the statistical multiplexing of many very high burstiness signals. A key property of any congestion control approach, which is not based on resource reservation at the peak rate of the sources, is the ability to shed load quickly without causing an avalanche of retransmissions of dropped traffic. With hierarchical coding of continuous media, when network congestion arises it is possible to drop the less important signal components without causing service interruption, and without the need for retransmissions. Since it is expected that continuous media will constitute the bulk of the network traffic, this technique can be very effective as a last resort for congestion control. Hierarchical coding can also play an important role at the receiver because it provides the system software with the capability of allocating resources based on local specifications and priorities. This might, for example, entail deciding to gracefully and dynamically degrade the quality of the received and played-back signal when resources are limited.

Video conferencing for scientific collaboration and animation of scientific visualization sequences, are *Sequoia 2000* applications that might benefit from the use of hierarchical coding for communication over packet-switching networks. Image database browsing can also take advantage of the reduced perceived latency of either progressive transmission or transmission of only the most important signal components.

1. Introduction

Images are a natural means of communication for humans; therefore, effective handling of images is very important for every advanced communication and information system. On the other hand, the information content of images is typically extremely high when compared with traditional messages exchanged over existing communication networks. This is particularly true for high resolution images, video, animations of satellite imagery, and scientific visualization sequences.

The high volume of data presents particular difficulties when the transmitted messages have stringent delay constraints, such as in the cases of video or fixed-rate animations. An application of particular interest, because of its potential for widespread use, is that of real-time interactive video communication. In this case the performance requirements are dictated by a combination of technological, economic, and psychological constraints. A similar situation arises when still images are animated and a fixed inter-frame time delay must be maintained.

The author can be contacted by E-mail: `polyzos@cs.ucsd.edu`, Tel.: 619-534-3508, Fax: 619-534-7029.

Video conferencing for scientific collaboration and animation of scientific visualization sequences, are *Sequoia 2000* applications that might benefit from the use of hierarchical coding for communication over packet-switching networks. Image database browsing can also take advantage of the reduced perceived latency of either progressive transmission or transmission of only the most important signal components.

A form of packet-switching, called the Asynchronous Transfer Mode (ATM) has been chosen as the universal switching method for the future Broadband Integrated Services Digital Network (B-ISDN), which promises to integrate the traditional telephone network, current data networks, and provide effective and efficient support for the newly arising multimedia applications. Even though there is still debate over the form and the implementation details, statistical multiplexing is a central feature of ATM (and packet-switching more generally), and one of the arguments for its selection over the alternative, Synchronous Transfer Mode (STM), a form of circuit-switching.

Statistical multiplexing is based on the premise that for independent sources the total amount of resources required (or used) to satisfy the traffic demands at any time is considerably less than the sum of the peak demands of the sources. Statistically, this is explained by the law of large numbers. Therefore, it is expected that applications that require high Quality-of-Service (QoS) will be able to reserve resources, not at the level of their peak demands, but at a much lower level, hopefully close to their average demand. This difference has significant economic implications.

Of course, there will be instances where peak demands are presented to the system simultaneously. In most of these cases the system will be unable to immediately satisfy these demands. In traditional packet-switching networks these situations have been handled through queueing of the requests. With real-time services, however, queueing might introduce unacceptable delays. Therefore, one of the techniques considered for traffic smoothing and controlling user perceived latency in times of high congestion in high-speed networks transporting continuous media, is the dropping or delaying of parts of the signal that might be of secondary importance. This approach is made possible by the use of hierarchical coding and appropriate traffic labeling.

Hierarchical coding is a technique for coding images and continuous media that separates the source signal into independent or hierarchically dependent components that are coded separately (and possibly with different coding schemes), and can therefore be decoded separately. By careful design, the first or first few components in the hierarchy can be a good approximation to the overall signal, providing a good first impression of the signal without requiring all the components to be received (and decoded) first.

This feature has been exploited by various image transmission schemes using low-speed communication channels in order to reduce the perceived latency of the transmission. For example, an image on a screen can be painted progressively, by painting a low resolution version of the whole image first (with low delay), and then by progressively adding detail and increasing the resolution. An obvious use of this technique arises for applications that involve browsing image databases; when a user realizes that the image shown is not the desired one, he or she can advance to the next image without any further delay. This determination can usually be made much sooner than the end of the transmission, decoding, and display process, depending of course on image size and resolution, and the communication bandwidth.

For successful interactive retrieval of images over a communications network, compression is essential. Interestingly, basic hierarchical coding features have already been incorporated into various coding and compression standards, such as JPEG and MPEG, and work is underway for extension of these standards towards further exploitation of hierarchical coding. Taking advantage of these provisions in packet-switching networks, however, has not yet been pursued aggressively, and the required mechanisms have not yet been adequately developed.

In the remaining of this paper we discuss the special characteristics of image transmission and interactive continuous media (in the remaining of section 1), and their network requirements (in section 2), including a brief presentation of compression standards. Then, in section 3, we present the concept of hierarchical coding and its possible applications for packet-switching networks. In section 4 we focus on multicasting of continuous media, an area where hierarchical coding seems particularly promising. Finally, in section 5, we provide a summary and conclusions.

1.1. Image Transmission

Images have been transmitted over conventional communication channels for a long time. The only real communications issue is the transmission delay of high resolution images over low capacity channels.[1] For example a 2000 by 2000 full-color (i.e., 24 bits per pixel), uncompressed image requires the transmission of 96 Mb. Using a modern, standard, voice-quality telephone line (a Narrowband-ISDN B-channel at 64 Kb/s) takes 1500 seconds, i.e., 25 minutes! Even a 1.5 Mb/s T1 line induces a 64 second delay. Compression becomes then essential for interactive applications. Compression techniques are discussed in more detail later.

Another latency-hiding technique (which can be combined with compression) is the use of *abstracts* [Fine92]. *Abstracts* are more flexible than compression and hierarchical coding and can provide arbitrary compression ratios, however, they require application dependent computation to achieve these advantages. On the other hand, hierarchical coding is perception oriented and uniform across applications. Of course, the most important components of a hierarchically coded signal can serve as an abstract (or a progressively finer series of abstracts).

Techniques for decreasing the latency of image browsing, usually based on various forms of hierarchical coding, have been and are still being developed, particularly for scientific applications [Tilton83, Beaulieu89, Tilton89, Tilton90, Moigne92]. Various systems have started to become available to a wider community of scientists. For example, SSABLE is a satellite data browsing system developed at the Scripps Institution of Oceanography at UCSD, that provides one-bit dithered browse products consisting of a decimated satellite pass and a swath map. One can also view and order processed (i.e., calibrated and cloud screened) products which are of substantially higher resolution, although still dithered to one bit [Simpson92]. Xbrowse is another software package, developed at the University of Rhode Island, that uses a local client to connect to a remote server to retrieve satellite and oceanographic data. It uses the technique of image pyramiding to send progressively finer resolutions of the data [Ma88, Kowalski90].

1.1.1. Image Compression Standards

Compression can be lossy or lossless. The selection of a particular approach depends on the application. Critical applications, such as medical imaging and scientific research in the advanced stages of hypothesis testing, are likely to use the lossless approach, even though it provides lower compression ratios (about 2:1). On the other hand, most commercial, industrial, consumer, and non-critical scientific applications can benefit significantly from the higher compression ratios of the lossy approach (up to about 50:1 for images and 200:1 for video). Highly reduced I/O and communication bandwidth requirements of the lossy compression approach make many more modes of operation and investigation feasible, e.g., browsing of animated sequences by "digital fast-forwarding."

Three standards have been prepared by ISO and CCITT for coding still pictures, videotelephony, and full-motion video. The JPEG (Joint Photographic Experts Group) standard is for continuous-tone still images, both gray-scale and color. The px64 standard is meant for use in video-telephony. And finally, the MPEG (Moving Picture Experts Group) standard is for full-motion video. We present these standards in later subsections, starting with JPEG below, and then discuss their features that support hierarchical coding and the use of non-constant bit rate channels for transmission (e.g., packet switching networks) in section 3.

1.1.2. The JPEG Standard

This standard was developed under the auspices of ISO and CCITT, and supports both lossy and lossless compression. The lossy methods are based on the Discrete Cosine Transform (DCT) and the lossless one is based on predictive coding. The standard specifies four modes of operation: sequential encoding, lossless encoding, progressive encoding, and hierarchical encoding [Wallace91]. The progressive and hierarchical modes are of particular interest here since both allow for decompression of a partially received signal; we present these two modes in section 3. Even though this standard was developed with still images in mind, it is expected that it will also be used for video transmission by providing intra-frame compression (only). Even though intra-frame compression provides much lower compression ratios for video, it has many advantages, particularly as far as error resiliency is concerned, because of the independence between frames. Therefore, the development of the JPEG standard is considered and important step for multimedia communications.

Sequential encoding is probably the most common mode for most applications. It involves encoding each image

---

[1] There are also many compatibility issues for format, aspect ratios, etc. We do not really address these questions here, except for pointing out in section 3 that hierarchical coding can help in many of these problems. See also [Gonzales92].

component[2] in a single left-to-right top-to-bottom scan. In the baseline encoding algorithm, each component of the source image is divided into 8x8 pixel non-overlapping blocks. The pixel values in each such block are first shifted from unsigned to signed integers and then input to the forward DCT. The resulting 64 DCT coefficient values can be regarded as the relative amount of 2D spatial frequencies contained in the input image. The DC coefficient is a measure of the average value of the 64 image pixels.

The next step is to quantize the DCT coefficients. The purpose of this step is to achieve further compression by quantizing high-frequency components with a larger step size (i.e., more coarsely). This is because high spatial frequencies require less detailed coding. This step discards visually unimportant information and thus makes the approach "lossy." These quantized coefficients are then entropy encoded, with the DC coefficients being treated specially. Since the DC coefficients are a measure of the average value of the pixels in the block, they are expected to show less variation within the same component, and therefore, are differentially encoded.

The quantized AC coefficients are ordered in a zig-zag sequence starting from the top-left corner and traversing the nearest cells first. This ordering puts the low-frequency coefficients before the high-frequency ones, and thus facilitates entropy coding. The coefficients are first run-length encoded and then coded using Huffman or arithmetic coding. The output from the entropy encoder is the output of the JPEG encoder. The JPEG decoder simply reverse this process, using an entropy decoder, dequantizer, and the inverse DCT to reconstruct the image.

The *lossless encoding* mode uses a simple predictive method. The encoder "predicts" the value of each pixel in the image component, by combining the values of up to three neighboring pixels using a simple predictor. This predicted value is then subtracted from the actual value of that particular pixel and the difference is encoded by either Huffman or arithmetic coding. In this mode the source image can have 2 to 16 bits/pixel of precision, and the typical compression observed is 2:1 [Wallace91].

1.2. Continuous Media

Due to the very high data volumes of video and audio (when compared with traditional media such as text and graphics) it is important to provide efficient network mechanisms even when high-speed fiber-optic networks are available. For example, CD quality audio requires 1.4 Mb/s and HDTV quality video requires 1.4 Gb/s. Compression, particularly for video, can reduce the required bandwidth by 1-2 orders of magnitude, e.g. MPEG-compressed HDTV quality video requires bandwidths of 20-40 Mb/s [LeGall91], still a significant amount especially considering that this is for only a single video stream.

In addition to very high throughput requirements, audio and video are usually associated with real-time interactive applications and thus present rather stringent delay requirements. More precisely, in order for audio and video to be effectively used in interactive communication, i.e. without forcing the communicating subjects to modify their behavior from that of face-to-face communication, the delay between transmitter and receiver is expected to be minimal. Various guidelines set this tolerance between 40 and 600 ms [AT&T84, Ferrar90, Hehman90]. However, it is not just the mean or the maximum delay that needs to be bounded; the jitter (i.e., the variance of the packet delay), also needs to be bounded (e.g., because of buffer limitations at the receiver) [Verma91].

While real-time interactive applications expect interactions with delays that are nearly imperceptible by humans, those real-time applications that are not necessarily interactive (as perceived by humans) are also interesting. For example, typical digital video distribution using a "TV model" could be real-time, but with no feedback (interaction) expected from the human receiver. On the other hand, if a "VCR model" is adopted for this distribution, the application should probably be characterized as interactive since the user could issue commands to control the flow of information and expect them to take effect immediately (by human reaction time standards).[3]

Multimedia applications, and more precisely continuous media components of such applications, may be classified into two categories [Clark92]:

---

[2] A color image can be represented in various color systems using complementary, independent)components. The most widely used systems are R-G-B (Red-Green-Blue) and Y-U-V (Luminance-Chrominance, i.e., a basic black-and-white intensity signal Y and the color difference signals U=Y-R and V=Y-B).

[3] However, there is still more tolerance to delay and less tolerance to errors, which is uncharacteristic of the bulk of interactive applications.

(1) intolerant and rigid ones, which require performance guarantees (which typically need to be provided through explicit reservations, even if the guarantees are of statistical nature and allow for occasional misses), and

(2) adaptive and tolerant ones, which expect a given quality of service, but can adjust or adapt to temporary degradation of service (in return for much lower cost); these could be satisfied without explicit reservation and micro-management of resources within the network (however, some admission control policy would still be required at the periphery of the network to secure the expected network performance).

1.2.1. VBR Coding of Continuous Media

Traditionally, transmission of continuous media, such as voice and video, has been based on constant bit rate circuit-switched channels. Therefore, source coding schemes for such media were designed specifically to produce output at a specific Constant Bit Rate (CBR), typically that of the channel to be used, independent of the instantaneous information content of the signal to be transmitted. This coding typically results in variable signal quality, even though when bandwidth is not severely limited, potential temporary quality degradation is usually engineered to be imperceptible.

For transmission over packet switched networks, the problem of continuous media coding changes. There is typically no a priori constraint on the maximum instantaneous bit rate produced by the coding scheme, but instead, an effort is made to produce constant quality signal at a given average bit rate (or level of quality) by using Variable Bit Rate (VBR) schemes. For example, video using interframe compression produces much lower information rates when there is little motion or change between frames, but generates very high bit rate peaks when there are scene changes or fast motion. Peak-to-mean bit-rate ratio for VBR codecs is usually very high. In [Ghanbari89] the value 4.7 is reported for one particular codec.

The use of VBR coding is based on the capability of packet switching networks to use statistical multiplexing for efficient utilization of network resources. One can then economize on resources when the information content of the signal is low and expend them when it is high, achieving the best overall (constant) quality at a given cost. A more complete discussion of VBR video coding in a high-speed network environment is provided in [Nomura89].

1.2.2. The MPEG Standard

The MPEG standard, proposed by ISO, has three parts. *MPEG-video* addresses the compression of full-motion, TV quality video signals at 1.5 Mb/s. The *MPEG-audio* and MPEG-system parts, address audio compression and audio-video synchronization issues respectively [LeGall91]. The rate of 1.5 Mb/s is suitable for use with CD-ROM, Digital Audio Tape, and T1 communication channels.

The MPEG video compression algorithm uses block-based motion compensation to reduce temporal redundancy, and DCT-based compression to reduce spatial redundancy. The standard only defines the bit-stream syntax and the decoding process [Jurgen92]. All MPEG decoders are expected to have the capability to decode a constrained parameter bit-stream. The constrained parameter set specifies a resolution equal to 720x576 pixels, a bit rate of 1.86 Mb/s, and a refresh rate of 30 frames per second.

Three types of pictures are considered in MPEG: intra-pictures (I frames), predicted pictures (P frames) and interpolated pictures (B frames - for bidirectional prediction). The I frames provide access points for random access but only moderate compression. The P frames are coded with reference to a past picture (I or P frame) and are used as a reference for future P frames. The B frames require both a past and future reference and provide the highest compression. When a picture is coded with respect to a reference, motion compensation is used to improve coding efficiency.

To exploit spatial redundancy, MPEG uses the similar techniques with JPEG, on 8x8 blocks of I, B or P frames. The quantization step, in this case uses different quantization tables based on various external parameters and the type of frame (I, P or B) being coded. The entropy encoding step uses Huffman-like tables with variable length codes. The syntax of an MPEG bit stream is six layered, with each layer supporting a particular function. This structuring separates entities in the bit-stream that are logically distinct. The MPEG bit-stream must satisfy particular constraints related to the buffer size required for decoding at the receiving end. This minimum required buffer size and the bit rate to be used are specified in the header of the bit-stream.

1.2.3. The *p*x64 Standard

The px64 standard (CCITT recommendation H.261) was proposed to enable (Narrowband) ISDN to be used for providing video services. This standard can be used for video transmission using p 64 Kb/s channels (p = 1,..., 30). For p = 1 or 2, the available bit-rate is severely limited and only videophone quality is possible [Sutherland92]. If p is greater than or equal to 6, higher quality video can be transmitted, e.g., suitable for video conferencing [Liou91].

The CCITT encoder is a hybrid coder, as it uses both DCT-based coding and predictive coding, in which a block in the current frame is predicted from a block in the previous frame using a feedback loop. The standard also has an optional specification for motion compensation. The video formats specified for use with this standard are the Common Intermediate Format (CIF) and the Quarter-CIF (QCIF). The standard specifies that all codecs must be able to operate with QCIF. CIF has a resolution of 360x288 pixels, and QCIF has a resolution of 180x144 pixels. The use of CIF is optional.

The first frame in the video sequence to be transmitted is intra-frame coded. Like JPEG, each 8x8 pixel block of a frame is encoded with DCT and then quantized. At this point there are two signal paths: one leads to the output channel through a lossless entropy coder; the other undergoes inverse quantization and inverse DCT to yield a reconstructed block used as feedback. The reconstructed block is needed for predictive coding, which requires tracking the behavior of the decoder to prevent the decoder's reconstructed image from diverging form the original input. After processing the entire frame, the full reconstructed image is stored in the frame memory of the encoder.

Next, inter-frame compression is done. For motion compensation, each 8x8 block in the current frame is matched with a search window in the frame memory. Then the motion vector that represents the offset between the current block and a block in the frame memory that forms the best match, is coded and sent to the output channel. The difference between the motion-compensated block and the original block is DCT coded, quantized and entropy coded, and then sent to the output channel.

The decoder is simpler than the encoder. It first decodes the entropy coded data. Then the data passes through the inverse quantizer and the inverse DCT, to yield the DCT coefficients. The reconstructed blocks of a frame are stored in the frame memory and are also output as the decoded image. In inter-frame mode, the motion vectors from the entropy decoder are used to provide the location of predicted blocks in frame memory. These predicted blocks are then output [Ang91].

2. Network Support for Interactive Multimedia Applications

Real-time applications provide many challenges for packet-switched networks. Traditionally, these applications have been handled by circuit-switched networks with fixed (channel) data rates, typically in analog form and with no error control and variable quality (depending on instantaneous information content and channel quality). The use of VBR digital transmission for applications such as video and audio makes packet-switching the desired mode of network transport because of the advantages of statistical multiplexing. Statistical multiplexing can realize large economies of scale and can support constant quality rather than constant bandwidth services.

Many of the techniques used in conventional packet-switched networks are based on assumptions of relatively slow, essentially non-interactive communication. It is important to re-examine those assumptions and reformulate the problems in the context of real-time, high-volume traffic. For example, one such assumption is the requirement for error-free transmission. The delay constraints of continuous media do not permit retransmissions for error recovery (at least end-to-end for transcontinental links). Therefore, the (open loop, end-to-end) packet error rate of the transport mechanism becomes an issue. Fortunately, however, real-time audio and video usually exhibit some tolerance to transmission errors, depending on the particulars of their encoding scheme and the level of compression. Thus, audio and video may be characterized as *soft real-time* traffic, i.e., they require some kind of statistical delay guarantees, such as percentile of packets (or bits) received within the given delay tolerance [Ferrari90].

A new problem introduced by the use of multiple continuous media is inter-media synchronization. One approach is targeted for traditional, physical or virtual, circuit-based communications, where all media streams are bundled together in a single wide stream (for example, by appending the corresponding audio segment to a video frame [Leung90]), and greatly simplifies the problem (assuming a single multimedia source). However, there is a loss in the opportunity for medium specific treatment within the network and the operating system at the destination.

The alternative approach, transmitting the different media separately, leads to a more general and flexible solution, albeit more complex. This solution introduces independence among the media, which has the potential to diminish the effects of communication errors, such as those leading to packet loss due to congestion. This is possible with protocols that support priorities based on the type of information transferred and which use these priorities to control congestion by "dropping" the least important packets, rather than being indiscriminate. Congestion control schemes based on these ideas have been proposed and are now under consideration for B-ISDN using ATM [Network92]. For example, graceful video quality degradation in the presence of congestion is usually more acceptable than audio degradation, and can be facilitated through the use of hierarchical coding.

The traditional debate between connections and datagrams resurfaces again in the case of continuous media. In an internetwork, communication is typically stateless, with all information traveling in datagrams. This datagram model is supported by the bursty nature of traffic in conventional networks. Additional semantics, such as connections, are then added on top of the network layer. The real-time requirements of continuous media demand a consistent quality of service. The cost of providing guaranteed quality of service (through resource reservation) is believed to be high enough that it is not worth repeating for each datagram. Furthermore, the long-livedness of the communication sessions further boosts the move towards connections. Interestingly, the traditional Internet (TCP/IP-type) architecture specifies some basic quality of service levels (which, however, are not adequate for real-time continuous media), and priority transmission [Comer88], which could prove effective in many situations. Unfortunately, priority forwarding is not typically implemented by the IP network software (and the quality of service specifications are typically ignored).

Many existing packet-switching networks tend to be naive at dealing with traffic admission. Every new packet is admitted into the network, regardless of the impact it may have on congestion, and the quality of service that existing connections may have been enjoying. This is undesirable when applications bargain for a specific quality of service which they expect to be maintained at more or less the same level. Admission control has to be introduced that determines the impact of a new multimedia connection on the existing network connections, and only admits new traffic when there is no negative impact [Clark92, Network92]. Tolerant and adaptive applications can be satisfied without explicit reservations and the necessary per-packet switching overhead within the network, whose cost can be significant. Some admission control at the connection level would probably be required, however, to secure the expected overall network performance.

For the *Sequoia 2000 Network* guaranteed performance protocols [Ferrari90] being developed at UC Berkeley will provide the necessary functionality. In this case, hierarchical coding should be able to significantly increase the available capacity by relaxing the performance requirements of a significant proportion of the (less important components of the) traffic, while the most important signal components take advantage of the performance guarantees.

We believe that flexibility in architecture design and implementation is more important than short-term implementation convenience, particularly since detailed requirements for key applications are still not entirely clear and the underlying network and device technology is not expected to be homogeneous. Furthermore, approaches supporting distributed designs for multimedia terminals, based on multiple specialized processors for I/O and network access working in parallel, will be important in the future. Hierarchically coded media transmission is ideally positioned to take advantage of these developments.

3. Hierarchical Coding

Hierarchical coding techniques (also referred to as component, layered, or sub-band coding) split signals into components of varying importance [Ghanbari89, Karlsson89]. The aggregation of these components reconstructs the original data, but subsets of them can also provide various degrees of approximation to the original signal. For example, a simple form of hierarchical coding could decompose a video frame into two sub-frames: (1) a low resolution component containing one quarter of the pixels, and (2) a high-resolution component containing the remaining three quarters of the pixels. For a receiver which uses a presentation window of size one quarter the size of the frame generated by the transmitter, receiving the second part of the signal would not be useful, and can actually be counter-productive because it can be a severe strain on local resources. This type of image coding is known as *pyramid decomposition* [Lippman89].

It seems therefore, that the use of hierarchical coding can be very important for continuous media transmission as it gives the system software at the receiver the capability of allocating resources based on local (i.e., the receiver's) specifications and priorities. This might entail deciding to gracefully and dynamically degrade the quality of the received and played-back signal when resources are limited. Also, it might facilitate local processing and presentation of the signal in ways not intended by the sender [Lippman91].

Hierarchical coding is also very important for the efficient use of high-speed networks. The main issue for such network architectures (e.g., ATM) is the significant congestion control problems which can arise due to the statistical multiplexing of very high burstiness signals [Eckberg92, Trajkovic92]. A key property of any solution (that is not based on resource reservation at the peak rate of the sources) is the ability to shed load quickly without causing an avalanche of retransmissions of dropped traffic [Lucantoni90]. With layered coding of continuous media, when network congestion arises it is possible to drop the less important signal components without causing service interruption, and without the need for retransmissions. This should only lead to a reduction in service quality (which may be unnoticeable in many cases). Since continuous media will constitute the bulk of the network traffic, this technique can be very effective as a last resort for congestion control. Many of the proposed congestion control techniques rely on this feature [Lucantoni90, Eckberg92, Trajkovic92].

Many different proposals have been made for hierarchical coding of video (and continuous media in general). [Karlsson89] presents a general approach to video transmission over packet-switching networks, emphasizing the role of hierarchical coding. One of the most basic and conceptually simple methods for hierarchical or layered coding is *bit-plane separation* [Karlsson89]. This method applies to images with representations that use multiple bits per pixel. It encodes first, separately, subsets of the image that contain the most significant bits (MSBs) of each pixel, which have the most important image information. Then, it progresses down through the bit-layers, towards the least significant bits, which provide the high resolution sub-signal. Thus each bit-plane (or each bit-plane subset) can be separately encoded, progressively transmitted, and decoded independently from the others. Interestingly, with bit-plane separation, the most (visually) important components (i.e., those produced by the MSBs), are also the more highly compressible ones.

More elaborate hierarchical coding is based on sub-band coding, where spectral decomposition is carried out on the luminance (intensity) portion of the video signal. This splits the frequency spectrum of the luminance temporally, vertically, and horizontally, to give 11 separate sub-bands. Then, the 13 sub-bands (i.e., the 11 luminance bands plus 2 chrominance components) are coded individually. In [Lazar90] the authors have considered a sub-band coding scheme with 11 bands, of which however, only 4 were determined to be important and were included in the measurements. The study revealed that source correlation can generate severe problems for naive statistical multiplexing schemes, emphasizing the importance of hierarchical coding.

Two-layer hierarchically coded video was considered in [Ghanbari89]. It was reported there that the video was of acceptable quality even with 10% packet loss at the second layer when the first layer was transmitted over a guaranteed performance channel, carrying approximately 20% of the total signal.

A traditional method of progressive coding uses quad-tree segmentation [Roy91, Nasrabadi90], where an image region is subdivided into four equal blocks if a given criterion is not met by that region. Each subdivision is further subdivided until the criterion is met or a minimum block size is reached. Typically the criterion is a bound on the variance of the pixel values. Areas like the slowly varying background in an image occupy the leaf nodes of the quadtree, whereas the foreground areas with most variation are in the high levels of the quadtree. A breadth-first traversal of the quadtree can then provide progressive coding.

The latest method proposed for progressive coding is the adaptive tree-structured segmentation [Wu92]. This method is similar to the quadtree method, but uses recursive partitioning of the image into convex n-gons ($2<n<9$), instead of equal sized squares. This is shown to achieve better compression. Also, the method embeds the segmentation in a binary tree, with each node having the shape and the intensity of the polygon. Through breadth-first traversal of the binary partition tree, the decoder receives gradual refinement of the entire image.

3.1. Potential Applications of Hierarchical Coding in Imaging

The following are some potential uses of hierarchical coding for images:

(1) *Multimedia database browsing* – While browsing through such a database, a user will experience very slow response time due to transmission and decoding delays. But if hierarchical encoding is used while storing the image, then the low-resolution encoding can be transmitted and displayed (only). As mentioned earlier, the JPEG encoder has provisions for this technique.

(2) *Resolution refinement* – This is the ability to produce still images at higher resolution than when viewed in motion [Lippman89]. If the various frames in a motion video are hierarchically coded, then only the acceptable resolution components can be decoded at the receiver, while the higher resolution components can be cached, to be used only for still frame display. This reduces the decoding time for the common case.

(3) *Zooming* – This is the capability of scaling an image up or down. This can be accomplished by switching between low-resolution downscaled picture icons [Fox91] and the full-scale picture at high-resolution, provided the picture has been encoded hierarchically.

(4) *Multiresolution capability* – High resolution images can easily be used by low resolution devices through hierarchical coding.

(5) *Low bandwidth transmission* – High quality images can be transmitted over low-bandwidth channels, with some loss in resolution.

(6) *Selective reduction* – Images can be reduced according to the importance of a particular portion of the picture. For example, in an x-ray retrieval system, if only a particular portion of the picture is of interest, this can be specified at the time of storage, and hierarchical coding can produce a more compact image.

(7) *Security* – Consider a multimedia database system using hierarchical coding to allow only some of the users to view them at full resolution, while others can only view low-resolution versions. This might be useful in cases where users need to see a scene but the faces of people in the scene should not be recognizable for security reasons.

3.2. Hierarchical Coding Provisions in Compression Standards

3.2.1. The JPEG Standard

The JPEG standard allows for progressive and hierarchical modes of operation. The progressive mode of JPEG called "successive approximation" is quite similar to bit-plane separation. The basic coding method in the hierarchical mode is a modification of sub-band coding.

In the *progressive encoding* mode, the image is encoded in multiple scans and the viewer (at the receiving end) can observe the image build up in successively more detailed versions. The DCT and quantization steps of the encoder are the same as in the sequential DCT mode. A coefficient buffer is added between the quantizer and the entropy encoder to store the quantized coefficients. The buffer is scanned in multiple passes, with each scan partially encoding the stored coefficients. The partial coding can be achieved by either or both of the following two methods: *spectral selection* or *successive approximation*. In *spectral selection*, "bands" of coefficients (in zig-zag order) may be encoded one by one in each successive scan. Each band contains coefficients occupying a particular part of the spatial frequency spectrum. With *successive approximation*, only the *N* most significant bits of a coefficient are encoded in the first scan, with less significant bits being encoded in subsequent scans.

The *hierarchical encoding* mode encodes an image at multiple resolutions, each differing from its adjacent level by a factor of two in the horizontal or vertical directions, or both. This is similar to the pyramid decomposition technique mentioned in [Lippman89]. The image is first bandsplit on the basis of spatial frequency and subsampled by the desired number of multiples of 2 in both dimensions. This new reduced size image is encoded using one of the sequential, progressive or lossless modes described previously. Then the encoded reduced-size image is decoded, interpolated and upsampled by 2 horizontally and/or vertically. This upsampled image is then used as a prediction of the original image at this resolution and the difference image is computed. The difference image is then encoded using one of the three previously described methods. Finally, the last two steps are repeated until the original image at full resolution has been encoded.

3.2.2. The MPEG and *p*x64 Standards

The MPEG standard does not specify a separate hierarchical mode of operation but is quite generic and flexible, in that it specifies only the bit-stream syntax and the decoding process. The encoder can thus use hierarchical coding, if desired, and communicate this to the decoder via the top layer of the bit-stream syntax.

The *p*x64 standard has a fast encoding and decoding process and is suitable for real-time video. The bit-rates are flexible, but in multiple of 64 Kb/s, and multiple resolutions are supported. Hierarchical coding-like features can be incorporated, by progressive transmission of Blocks and Macro Blocks.

3.3. Advantages and Disadvantages of Hierarchical Coding

There are, of course, tradeoffs between the benefits derived from the relative independence of the components produced by hierarchical coding and the total volume of data generated by compressing components individually (in order to achieve their independence). Interestingly, as we have seen, many compression standards support various forms of hierarchical coding. Another related observation is that with hierarchical coding, the basic (e.g., low resolution) components, which are important for signal continuity, are highly compressible. A reasonable strategy then is to transmit the low resolution, high priority components through connections with explicit performance guarantees, and send the other components on connections without explicit guarantees or require less stringent guarantees.

4. Hierarchical Coding for Multicasting Continuous Media

One of the ways communication can be characterized is by the number of receivers targeted by a sender. Traditional communication modes have been one-to-one, or unicast, and one-to-all, or broadcast. The generalization of these two extremes is multicasting, the communication of a single message to a select group of recipients which may or may not include the sender.

In multicasting, a sender communicates explicitly with a "group," rather than with individual actors which happen to be members of the group. Consequently the sender need not be knowledgeable of group membership, and in fact, the group membership may change over time. This model of communication supports applications where control and/or data are distributed amongst multiple actors. Examples include updates to replicated databases, obtaining a service from one out of a group of servers, and inter-process communication between cooperating processes.

One interesting class of applications which require group communication includes those with multimedia I/O components. It is only recently that technology has become available for workstations to be equipped with audio and video devices. As these devices proliferate and become standard equipment in personal computers and workstations, and as high-speed fiber optics provide the necessary bandwidth for continuous media networking, multimedia group applications will become standard tools for communication and collaboration. A canonical example of such applications is video-conferencing [Sabri85, Casner90]. More generally, the class of applications called groupware [Ellis91], and computer-supported cooperative work (CSCW) [Grief92], enhanced with audio and video I/O, require group communication.

Efficient multicasting is a fundamental issue for the ultimate success of these multimedia group applications. While in the past multicasting has been viewed as a service of limited use, often provided as an afterthought, this can no longer be the case. A major market for Broadband ISDN (B-ISDN) is expected to be selective video distribution [Sincoskie90, Sutherland92], analogous to CATV channels, but where the relatively small number of active receivers and the large number of channels (sources) make broadcast solutions impractical.

In [Pasquale92a], we discuss the problem of multicasting multimedia data, particularly multiple time-correlated streams of continuous media (e.g., audio and video). Briefly, the relevant characteristics of this problem are: (1) data must be sent to multiple destinations, (2) the data is large, and requires high bandwidth, and (3) the value of the data is sensitive to delay; if it is received beyond a threshold dictated by human perception, it becomes useless. Since the data is large, minimizing duplication of transmission is important. And since the data is sensitive to delay, finding short routes which can transport data concurrently to the multiple destinations is important.

4.1. The Feedback Control Problem

A number of dynamic control mechanisms in protocols employ feedback from the receiver to the sender. For flow control, the receiver tells the sender to slow down or speed up to adjust to the receiver's current ability to consume packets. For error control, the receiver tells the sender to resend corrupted packets or packets that have not arrived within some time interval, and to acknowledge packets that have been successfully received. However, when there are multiple receivers and a single sender, how to provide feedback efficiently (or whether it should be provided at all) is a non-trivial question.

There are four basic approaches. The first solution is to ignore the problem, and is the simplest. This may be appropriate for connectionless services such as datagrams, where the sender's responsibility is simply to transmit the datagram, and if problems arise along the way (e.g., the packet is corrupted, or the communication subnet or receiver is overloaded and the packet must be dropped), resolution is delegated to a higher level protocol.

In the second solution, the sender transmits a single packet, but then acts as if it just sent n packets to the n receivers, maintaining state information about each of the n "virtual packets" in transit. The sender must then wait until a stable state is reached before sending the next packet, i.e., until it is satisfied that all the receivers are ready for the next packet.[4] For flow control, this means slowing down to the consumption rate of the slowest receiver, forcing all other receivers to be delayed. For error control, this means taking corrective actions based on the most problematic receiver, and forcing all other receivers to be delayed. Unlike flow control, it may not be possible to return to a stable state which applies to all the receivers, as when a maximum number of retransmissions to some receiver have occurred without acknowledgement. In this case, the bad receiver may have to be treated special, perhaps by removing its membership from the multicast group, and allowing any special recovery actions to take place, while the rest of the group can go on receiving packets. Clearly, this is a complex solution, and performance degrades to that of the worst receiver (e.g., the slowest, the most overloaded, or the one connected through the slowest communication channel).

The third solution is to distribute the control mechanism over the nodes of the multicast tree. Control is now hierarchical (as dictated by the existing multicast tree). A receiver's feedback need not propagate all the way back to the sender. When an intermediate node receives the feedback, it either takes corrective action itself, or it propagates the feedback (which is possibly fused with feedback received from other downstream nodes) upstream to the next higher level node, and the action is recursive. This solution attempts to localize feedback and corrective actions to that part of the tree containing the problems (e.g. misbehaving links or receivers). A potential benefit is that receivers, which have paths to the sender that do not go through nodes currently reacting to such problems, achieve better performance. While this solution may lend itself to elegant algorithms for distribution of complexity, the fact that the complexity of intermediate nodes rises, may be too costly, especially given the trend in networks towards simple switching nodes.[5]

---

[4] This simplified description captures the essence of this approach. However, many similar but more sophisticated schemes have been devised; for example, see [Chandran92, Mase83] and the references therein.

[5] However, this cost may be completely acceptable for less dynamic control mechanisms which are not overly time critical, e.g., resource reservation for connection setup. The discussion here mainly applies to dynamic control mechanisms.

Finally, the fourth solution is to take preventive action to minimize any reliance on feedback. For error control, this means using forward-error correction [Shacham90, Biersack92] to anticipate errors and provide information redundancy, allowing the receivers to reconstruct the information without asking the sender to retransmit. For flow control, this means reserving resources so that receivers and intermediate switching nodes are always able to support the flow rate dictated by the sender. The value of this approach is that there is no complexity due to feedback control mechanisms, and yet is an improvement over simply doing nothing (as in the first solution). The cost is the additional complexity due to these anticipatory/preventive action schemes. Note that the resource reservation scheme will generally employ a protocol which uses feedback to determine resource allocations. However, since resource reservation takes place once, before the start of actual information flow (rather than multiple times throughout the information flow, as do the dynamic feedback control mechanisms described above), its cost may be acceptable (even if a distributed scheme is used).

The fourth solution seems to be the most appropriate for multicasting of continuous media. Given the non-critical error-tolerance characteristics of most continuous media applications, forward-error correction schemes are appropriate (of course, as errors become more critical, the level of redundancy can be increased). As for resource reservation, not only is this appropriate for transporting continuous media, but we see an opportunity where hierarchical coding, with a separate reservation per component, can be used to great advantage to deal with heterogeneous receivers. This is described in more detail below.

4.2. Error Control

In point-to-point channels, error-free communication is usually achieved by error detection and retransmission. Error control is particularly difficult with multipoint connections because of the feedback control problem. A naive multicast error control scheme could be based on positive acknowledgements. In that case, if communication is hampered by transmission errors, retransmissions will reach all members of the group, including the ones that received the packets correctly. In order to minimize the overhead, alternative schemes use negative acknowledgments, taking advantage of the high reliability of links. These schemes are based on the observation that if errors occur only sporadically, then recipients need respond to the sender only on errors, asking for retransmissions. This requires synchronization packets to be sent periodically to update all receivers on which packets were transmitted successfully. Various schemes improving on traditional Automatic Repeat reQuest (ARQ) techniques have been devised [Mase83, Chandran92].

However, the real-time requirements of multimedia applications do not typically permit ARQ solutions (at least in general settings, e.g., when transcontinental links are included in the path). Various combinations of forward error correction schemes and error recovery and concealment techniques will probably be necessary. Interestingly, the choice of open-loop error control suggested by the real-time character of continuous media also solves the multipoint feedback control problem by eliminating the need for feedback at the lowest layers of the protocol hierarchy.

An important characteristic of real-time interactive applications such as videoconferencing is that the applications can tolerate some number of errors without significant degradation in quality. The consequence is that even without fully reliable transmissions, applications can be made to recover gracefully from loss of segments (at least for some media components). There is, thus, a tradeoff for real-time continuous media between smooth progress and complete reliability. Typical video and audio information will be compressed for efficient transportation. This decreases the redundancy in the data, and thus limits the error resiliency of the applications in question. Hierarchical coding for continuous media can improve the error resiliency of multimedia applications.

4.3. The Value of Hierarchical Coding for Multicasting

It is expected that there will be many types of multimedia terminals and that possibly many formats for representation of each medium type will exist [Jurgen92]. This has been the case for traditional text-based computer communications (where the compatibility problems are considerably simpler than for multimedia). The presentation layer in the OSI architecture addresses the issues of data representation compatibility problems. When incompatible transmitters and receivers communicate, translation is necessary. This service can be provided in one of three places: (1) at the transmitter, (2) at the communications channel (or within the network), or (3) at the receiver.

If the third solution is possible, there is no real compatibility problem. The second approach is the typical solution (e.g., traditional protocol converters). The first solution is similar to the third in the point-to-point case, but is rather inefficient in the multipoint case. It requires the sender to translate its data format once for every incompatible receiver. This translation consumes sender resources that are usually highly contended. Furthermore, it requires the network to transport a higher volume of information because sharing of links is not possible. In essence, severe compatibility problems limit the effectiveness of multicast solutions and the communication model reverts to a series of unicasts.

Facilitating translation at the receiver through synthesis of the signal from separate components can be achieved through hierarchical coding. In addition, network routing algorithms may take into account the different needs and capabilities of the destinations by, for example, forwarding only usable components to select destinations [Shacham92]. Furthermore, there is an interesting interaction between hierarchical coding and multicasting. Hierarchical coding enables destinations to adjust the quality of the signal they receive each, independently and without the source actually being aware of this adjustment (of course, the adjustment is only possible towards lower quality). This is a very important property considering the feedback control problems of multipoint communication, and can also be used to effectively address many compatibility problems.

Thus, hierarchical coding can help diminish compatibility problems for continuous media and also address other problems such as real-time delivery, error control, and network congestion control, as well as minimize communications cost.

4.4. The Multimedia Multicast Channel

The Multimedia Multicast Channel (MMC) is a programming abstraction which supports dissemination-oriented communication. The abstraction is analogous to that of a cable television channel: a source transmits onto a specific channel, and receivers which have subscribed to that channel receive media streams (e.g., audio and video) without explicit interactions with the source. The MMC communication paradigm is a major departure from more traditional ones, in that a source and a set of receivers are very loosely coupled in their control and data exchange interactions. In general, the source's main concern is to push various media streams onto a channel, without emphasis on where they end up (i.e., who the actual receivers are), and how they are used (i.e., what specific receivers will extract from any or all of the streams). A receiver's main concern is what to extract from a channel, which is viewed as offering multiple media streams, some or all of which are of interest.

Perhaps the most unique feature of this communication paradigm is in how it addresses heterogeneity, in particular that it is not unusual for the forms of the media streams, generated by the source and required by the different receivers, to be quite different. For example, the source may generate HDTV-quality video and CD-quality audio, whereas some receivers can only use NTSC video and its associated audio, while other receivers can use only audio and no video. Indeed, every receiver may have very different and independent requirements. Therefore, it is expected that the receivers will individually tailor, to a high degree, what streams are actually received. These streams may be a subset of the source streams, or new ones computed from the source streams.

The use of hierarchical coding at the source to facilitate tailoring is encouraged. We believe this communication paradigm is highly appropriate for distribution services required by a large class of multimedia multipoint applications. A prime example is "video distribution" [Sincoskie90, Gusella91], as in cable television systems where a single source generates video (and associated audio) distributed to a large set of receivers who generally have little or no interaction with the source. Another application is video conferencing, where each member is both a source and receiver. This application would require a separate MMC per source to support base-level audio-video distribution. However, video conferencing also requires other control mechanisms which are outside the scope of what is provided by the MMC.

More generally, one can distinguish between higher-level application-specific control mechanisms such as floor-control and voting, and lower-level media-oriented control mechanisms such as modifying the resolution, granularity, or intensity of a media stream, and mixing or synchronizing multiple media streams. Media-oriented control does not imply (explicit) control between source and receivers, and is generally useful in most multimedia applications. Consequently, the MMC supports media-oriented control through its filter mechanism. All other required control, particularly application-specific control between source and receivers, is expected to be provided by the application itself (e.g., through the use of libraries/toolkits). The architecture of the MMC is introduced in [Pasquale92].

5. Conclusions

Various hierarchical (or progressive) coding schemes for images and video have existed for a long time and have been used to improve the user perceived characteristics of low-speed communication channels. Furthermore, research into new coding schemes with improved characteristics continues. Hierarchical coding is now catching the attention of network specialists who are trying to reconcile the great economic advantages of statistical multiplexing in high-speed packet-switching networks with the complexity of controlling the congestion phenomena that it produces. It promises to provide the necessary freedom to communications engineers to design networks without reverting into an absolute, tight control of all network resources. It also might help into individual, economical customization of multicast signals, and to address the problems of (multimedia) terminal heterogeneity.

The two areas of application that seem to have the highest potential to be influenced by and help in the adoption of various hierarchical coding schemes are (i) image and video database browsing, and (ii) real-time continuous media transmission, particularly to multiple destinations (multicasting). These two application areas have different expectations with respect to the required preparation for adoption of hierarchical coding techniques.

Common to both application areas is the requirement for applications and network system software to support transmitters and receivers that use progressive coding of media. Even though the concept is not new, there is little support for it in current tools, protocols, etc. A first step in this direction is our Multimedia Multicast Channel proposal [Pasquale92].

Another requirement is for communication networks to provide the appropriate quality of service. For the *Sequoia 2000 Network* guaranteed performance protocols [Ferrari90] being developed at UC Berkeley will provide this functionality. Interestingly, the traditional Internet (TCP/IP type) architecture specifies some basic quality of service levels (which, however, are not adequate for real-time continuous media), and priority transmission, which could prove effective in many situations. Unfortunately, priority forwarding is not typically implemented by the IP network software (and the quality of service specifications are typically ignored).

In order to take full advantage of the potential benefits in the case of image browsing, in addition to the necessary application modifications, it is important that the database and the storage system is organized in a way that individual media components be directly accessible. This probably entails restructuring existing databases, plus incorporating new functionality into database management systems in order to provide the retrieval operations efficiently and according to the hierarchical coding spirit. To further reduce latency for applications retrieving images across wide area packet-switching networks, the components of high visual importance should be transmitted as high-priority traffic.

Acknowledgements

This document has been influenced by our work on multicasting continuous media with Joseph Pasquale, Eric Anderson, and Vach Kompella, and on layered video coding and priority IP forwarding with Scott McMullan. It has also benefited from discussions with all members of the UCSD Computer Systems Laboratory. I'm indebted to all of them for their contributions and I thank them all.

References

[Ang91] P.H. Ang *et al.*, "Video Compression Makes Big Gains," IEEE Spectrum, October 1991, pp.16-19.

[AT&T84] AT&T Bell Laboratories, Engineering and Operations in the Bell System, 1984.

[Beaulieu89] J.-M. Beaulieu, "Hierarchy in Picture Segmentation: A Stepwise Optimization Approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11(2), pp. 150-163, February 1989.

[Biersack92] E. W. Biersack, "Performance Evaluation of Forward Error Correction in ATM Networks," Proc. SIGCOMM '92, pp. 248-257 (August 1992).

[Casner90] S. Casner, K. Seo, W. Edmond, and C. Topolcic, "N-Way Conferencing with Packet Video," Third International Workshop on Packet Video (March 1990).

[Chandran92] S. R. Chandran and S. Lin, "Selective-Repeat-ARQ Schemes for Broadcast Links," IEEE Transactions on Communications Vol. 40(1), pp. 12-19 (January 1992).

[Clark92] D. D. Clark, S. S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," Proc. ACM SIGCOMM '92, pp. 14-26 (August 1992).

[Denatale91] F.G.B. Denatale *et al.*, "Hierarchical Image Coding via MSE-Minimising Bilinear Approximation," Electronics Letters, October 1991, Vol. 27, No. 22, pp. 2035-2037.

[Eckberg92] A. E. Eckberg, "B-ISDN/ATM Traffic and Congestion Control," IEEE Network Vol. 6(5), pp. 28-37 (September 1992).

[Ellis91] C. A. Ellis, S. Gibbs, and G. L. Rein, "Groupware -Some Issues and Experiences," Communications of the ACM Vol. 34(1), pp. 38-58 (January 1991).

[Ferrari90] D. Ferrari and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," IEEE Journal on Selected Areas in Communications Vol. 8(3), pp. 368-379 (April 1990).

[Ferrari90a] D. Ferrari, "Client Requirements for Real-Time Communication Services," IEEE Communications Magazine Vol. 28(11), pp. 65-72 (November 1990).

[Ferrari92] D. Ferrari, J. C. Pasquale, and G. C. Polyzos, "Network Issues for Sequoia 2000," Proc. IEEE Compcon Spring '92, (February 1992).

[Fine92] J. A. Fine, T. E. Anderson, M. D. Dahlin, J. Frew, M. Olson, and D. A. Patterson, "Abstracts: A Latency-Hiding Technique for High-Capacity Mass-Storage Systems," Sequoia 2000 TR, March 1992.

[Fox91] E.A. Fox, "Advances in Interactive Digital Multimedia systems," IEEE Computer, Vol. 24, No. 10, pp. 9-21 (October 1991).

[Ghanbari89] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks," IEEE Journal on Selected Areas in Communications Vol. 7(5), pp. 771-781 (June 1989).

[Gonzales92] C. Gonzales and E. Viscito, "Flexibly Scalable Digital Video Coding," submitted for publication to Image Communications, 1992.

[Grief92] I. Grief, Ed., Computer Supported Cooperative Work, Morgan Kaufmann Publishers (1992).

[Gusella91] R. Gusella and M. Maresca: Design Considerations for a Multimedia Network Distribution Center: Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany (November 1991).

[Hehmann90] D. Hehmann, H. Salmony, and H. Stuettgen, "Transport Services for Multimedia Applications on Broadband Networks," Computer Communications, Vol. 13(4), pp. 197-203 (1990).

[Jurgen91] R. K. Jurgen, "The Challenges of Digital HDTV," IEEE Spectrum, April 1991, p. 28.

[Jurgen92] R. K. Jurgen, "An Abundance of Video Formats," IEEE Spectrum, March 1992, pp. 26-28.

[Jurgen92a] R. K. Jurgen, "Digital Video," IEEE Spectrum, March 1992, pp. 24-30.

[Karlsson89] G. Karlsson and M. Vetterli, "Packet Video and Its Integration into the Network Architecture," IEEE Journal on Selected Areas in Communications Vol. 7(5), pp. 739-751 (June 1989).

[Kompella92] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicasting for Multimedia Applications," Proc. IEEE INFOCOM'92, pp. 2078-2085 (May 1992).

[Kowalski90] J. Kowalski, J. Gallagher, and P. Cornillon, “Remote Browsing of AVHRR Images Based on Dynamically Selectable Resolution,” Proceedings of MTS 90, Washington DC (September 1990).

[Lazar90] A. A. Lazar, G. Pacifici, and J. S. White, "Real-Time Traffic Measurements on MAGNET II," IEEE Transactions on Communications Vol. 8(3), pp. 467-483 (April 1990).

[LeGall91] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," Communications of the ACM Vol. 34(4), pp. 46-58 (April 1991).

[Leung90] W. H. Leung, T. J. Baumgartner, Y. H. Hwang, M. J. Morgan, and S.-C. Tu, "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks," IEEE Journal on Selected Areas in Communications Vol. 8(3), pp. 380-390 (April 1990).

[Liou91] M. Liou, "Overview of the px64 kbit/s Video Coding Standard," Communications of the ACM, Vol. 34, No. 4, pp. 59-63 (April 1991).

[Lippman89] A. Lippman and W. Buttera, "Coding Image Sequences for Interactive retrieval," Communications of the ACM Vol. 32(7), pp. 852-860 (July 1989).

[Lippman91] A. Lippman, "Feature Sets for Interactive Images," Communications of the ACM Vol. 34(4), pp. 92-102 (April 1991).

[Lucantoni90] D. M. Lucantoni and S. P. Parekh, "Selective Cell Discard Mechanisms for a B-ISDN Congestion Control Architecture," 7th International Teletraffic Congress Symposium (1990).

[Ma88] M. Ma, R. L. Danielson and W. C. Likens, 'A Preliminary Investigation of Communication Techniques for Local and Remote Access to Image Databases," Geocarto International (3) 1988.

[Mase83] K. Mase, T. Takenada, H. Yamamoto, and M. Shinihara, "Go-Back-N ARQ Schemes for Point-to-Multipoint Satellite Communications," IEEE Transactions on Communications Vol. 31(4), pp. 583-589 (April 1983).

[Moigne92] J. Le Moigne and J. C. Tilton, "Refining Image Segmentation by Integration of Edge and Region Data," Proceedings of the 1992 International Geoscience and Remote Sensing Symposium, League City, Texas, pp. 1406-1408 (May 1992).

[Nasrabadi90] N.M. Nasrabadi *et al.*, "Hierarchical Block Truncation Coding of Digital HDTV Images," IEEE Transactions on Consumer Electronics, Vol. 36, No. 3, pp. 254-261 (August 1990).

[Network92] Special Issue: Congestion Control in ATM Networks, IEEE Network Vol. 6(5) (September 1992).

[Nomura89] M. Nomura, T. Fujii, and N. Ohta, "Basic Characteristics of Variable Rate Video Coding in ATM Environment," IEEE Journal on Selected Areas in Communications, Vol. 7(5), pp. 752-760 (June 1989).

[Pasquale90] J. Pasquale and G. Polyzos, "System Support for Multimedia Applications," Proc. First International Workshop on Network and Operating System Support for Digital Audio and Video, International Computer Science Institute, Berkeley, CA (November 1990).

[Pasquale91] J. C. Pasquale, G. C. Polyzos, E. W. Anderson, K. R. Fall, J. S. Kay, V. P. Kompella, S. R. McMullan, and D. Ranganathan, "Network and Operating System Support for Multimedia Applications," Technical Report CS 91-186, University of California, San Diego, La Jolla, CA (March 1991).

[Pasquale92] J. C. Pasquale, G. C. Polyzos, E. W. Anderson, and V. P. Kompella, "The Multimedia Muticast Channel," Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, CA (November1992).

[Pasquale92a] J. C. Pasquale, G. C. Polyzos, and V. P. Kompella, "The Multimedia Multicasting Problem," submitted for publication, 1992.

[Polyzos92] G. C. Polyzos, "Networking for Sequoia 2000," Proc. Fourth IEEE International Workshop on Multimedia Communications, MULTIMEDIA '92, Monterey, CA (April 1992).

[Roy91] J.U. Roy and N.M. Nasrabadi, "Hierarchical Block Truncation Coding," Optical Engineering, Vol. 30, No. 5, pp. 551-556 (May 1991).

[Sabri85] S. Sabri and B. Prasada, "Video Conferencing Systems," Proceedings of the IEEE Vol. 73(4), pp. 671-688 (April 1985).

[Shacham90] N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management," Proc. IEEE INFOCOM '90, pp. 124-131 (June 1990).

[Shacham92] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," Proc. IEEE INFOCOM'92, pp. 2107-2114 (May 1992).

[Simpson92] J. J. Simpson and D. N. Harkins, "The SSABLE System: Automated Archive, Catalog, Browse and Distribution of Satellite Data in Near-Real Time," submitted for publication to IEEE Transactions on Geoscience and Remote Sensing, 1992.

[Sincoskie90] W. D. Sincoskie, "Video On Demand," IEEE GLOBECOMM '90 (December 1990).

[Sutherland92] J. Sutherland and L. Litteral, "Residential Video Services," IEEE Communications Magazine Vol. 30(7), pp. 36-41 (July 1992).

[Tilton83] J. C. Tilton and S. C. Cox, "Segmentation of Remotely Sensed Data Using Parallel Region Growing," Proceedings of the Ninth International Symposium on Machine Processing of Remotely Sensed Data, (IEEE Catalog No. 83 CH 1835-8 MPRSD), Purdue University, W. Lafayette, IN, pp. 130-137 (June 1983).

[Tilton89] J. C. Tilton, "Image Segmentation by Iterative Parallel Region Growing and Splitting," Proceedings of the 1989 International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, pp. 2420-2423 (July 1989).

[Tilton90] J. C. Tilton and M. Manohar, "Hierarchical Data Compression: Integrated Browse, Moderate Loss, and Lossless Levels of Data Compression," Proceedings of the 1990 International Geoscience and Remote Sensing Symposium, College Park, MD, pp. 1655-1658, May 20-24, 1990.

[Trajkovic92] L. Trajkovic and S. J. Golestani, "Congestion Control for Multimedia Services," IEEE Network Vol. 6(5), pp. 20-26 (September 1992).

[Verma91] D. C. Verma, H. Zhang, and D. Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network," Proc. TriComm '91, pp. 35-43 (April 1991).

[Wallace91] G. K. Wallace, "The JPEG Still Picture Compression Standard," Communications of the ACM Vol. 34(4), pp. 30-44 (April 1991).

[Wu92] X. Wu, "Image Coding by Adaptive Tree-structured Segmentation," IEEE Transactions on Information Theory, Vol. 38, No. 6, pp. 1755-1767 (November 1992).