A Layered Compression Scheme for Multicasting Medical Images across Heterogeneous Networks *

Anindo Banerjea[a] Wai-tian Tan[b]
Avideh Zakhor[b]

[a]Philips Research, 1070 Arastradero Road
Palo Alto, CA 94043

[b]Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

ABSTRACT

In this paper we evaluate a layered coding technique based on subband coding for the purpose of encoding medical images for realtime transmission over heterogeneous networks. The objective of this research is to support a medical conference in a heterogeneous networking scenario. The scalable coding scheme under study in this paper generates a single bit-stream, from which a number of sub-streams of varying bit-rates can be extracted. This makes it possible to support a multicast transmission scenario, where the different receivers are capable of receiving different bit-rate streams from the same source, in an efficient and scalable way. The multirate property also allows us to provide graceful degradation to loss when used over networks which support multiple priorities. This paper evaluates the quality of the video images encoded with the layered encoding technique at different bit-rates in terms of the Peak Signal to Noise Ratio (PSNR) for cine-angiogram video. It also describes experiments with the transmission of the video across an Aysnchronous Transfer Mode (ATM) Local Area Network (LAN), using a two layer encoded video stream, and assigning different network service classes to the two layers. We study how the quality of the reconstructed signal changes with the ratio of the bit-rates of the high and low priority layers, for various levels of congestion in the ATM network.

Keywords: Multi-rate coding, scalable compression, medical imaging, heterogeneous networks, remote conferencing

1. INTRODUCTION

The widespread deployment of high-speed networks has spurred the development of multimedia applications, such as voice and video. In the medical domain, the transmission of medical images over networks opens the possibility of improved education by allowing remote participation in clinical conferences, or improved and more cost-effective diagnosis by allowing remote consultations with experts. This development is aligned with trends such as the rise of the managed health care organizations and the increased pressures for cost reduction in medical care. The medical community is looking for ways to use technology to increase the cost efficiency of the delivery of medical care.

Thus the use of networking and multimedia technologies in the medical profession are likely to expand. However, current networking infrastructures are heterogeneous in terms of widely different link bandwidths, Quality of Service (QoS) support, protocols, and other characteristics. We also find heterogeneity in the terminal equipment connected to the network. In a medical scenario, we might expect to see fast high resolution medical imaging workstations with high bandwidth network connections, coexisting with personal computers on slower local area networks, or hand-held devices over wireless networks for mobility. Heterogeneity is another trend that is likely to continue in the foreseeable future.

Bandwidth heterogeneity poses a problem in the context of realtime network services, such as audio or video conferencing, or the realtime transmission of medical images, because the bit-rate at which the communication can

*This work was supported by NSF grant MIP-9057466, ONR grant N00014-92-J-1732, AFOSR contract F49620-93-1-0370, California State Program MICRO, Philips, SUN Microsystems, LG electronics and Tektronix.

Anindo Banerjea is now at the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada.
E-mail: banerjea@comm.utoronto.ca, dtan@eecs.berkeley.edu, avz@eecs.berkeley.edu

be carried out depends on the capacity of the entire chain of links that the data must traverse, as well as the processing and display capability of the receiver. For realtime communication, if the bit-rate available at any point is not sufficient for the complete signal, one option is to reduce the resolution and/or rate of the signal. In the case of a video signal, this could be a reduced spatial resolution (a smaller picture), reduced temporal resolution (fewer frames per second), or reduced SNR resolution (lower signal quality).

For the multicast transmission of data to multiple receivers, which is the case for multiparty conference scenarios, the bit-rate that the source choses to transmit at depends on the capabilities of all the receivers. In the traditional approach, the source must pick one bit-rate, and this determines the video quality seen by all the receivers. If the source transmits at the lowest common denominator, the receivers with higher capacity network connections, display resolution and processing powers would get the same low quality picture that the slowest participant in the conference receives, because the quality is being determined on the basis of the capability of the slowest receiver. If the source transmits at the bit-rate corresponding to the fastest receiver, other receivers would observe random losses in the received signal, because the data would be coming in faster than the link or receiver itself could handle, and packet losses due to buffer overflow would occur.

One possible solution to this is simulcast transmission, where the source transmits multiple copies of the signal, each at a different resolution and/or rate. However, this implies inefficient use of the capacity of the source and parts of the network, since the sum of the bit-rates of all the different signals is greater than the single bit-stream that is required to represent the signal at the highest resolution. The other alternative is layered transmission, where the data is encoded into a low resolution "base" stream, and a series of enhancement streams, such that the total rate of all the streams is close to the rate required for a single stream at the highest resolution.

Consider a layered encoder that generates k layers, numbered $0..k-1$, with layer 0 being the base layer, and the higher numbered layers being successive enhancement layers. It is possible to reconstruct a videosignal by combining layers 0 through $l-1$ for any $l < k$. As we increase l, the number of layers selected, the combined bit-rate of the resulting stream increases, and so does the video quality of the decoded video. We obtain increased spatial and temporal resolution, as well as better signal quality. Thus, it is possible to tradeoff the required bandwidth and the resulting video quality on the fly, without having to reencode the data. Moreover, the choice of the number of layers to pass through the network (and hence the required bit-rate) can be made at any point in the path from the source to the receiver. The overhead of the layered encoding scheme, as compared to an unlayered video signal, depends on the specific encoding algorithm.

The different layers of a multi-resolution layered encoded video have different perceptual importance to the quality of the resulting video. Thus, the occurence of errors or losses in the different layers have different degrees of importance. A loss or error in a higher numbered layer has less effect on the decoded video than a loss or error in a lower layer. If the network supports multiple service levels, it us useful to provide higher protection or Quality of Service (QoS) to the lower numbered layers, so as to obtain graceful degradation of the overall video quality when congestion occurs in the network.

This paper evaluates the performance of a layered coding scheme for the transmission of medical images in terms of its compression efficiency, speed, as well as quality of received image under different network conditions. Section 2 describes the details of the encoding process. Section 3 presents an evaluation of the encoder, in terms of a comparison of the signal quality as a function of the bit-rate to a standard compression scheme (MPEG1). Section 4 presents some results concerning the transmission of the layered signal over a high speed optical Asynchronous Transfer Mode (ATM) network. We conclude with a summary of the achieved results and plans for future work.

2. SCALABLE CODING SCHEME

By scalability, we mean the ability to produce an *embedded* bit stream, from which one can extract arbitrary number of subsets to decode at different rates. There are many existing approaches to code video in a scalable fashion. While MPEG based scalable coding schemes[4] typically offers few rates, one alternative way to achieve scalability is to use subband coding, a method which uses iterated filter banks to decompose video frames into different spatio-temporal frequency subbands.

Because different subbands have different visual importance, one way to achieve scalability is to drop subbands of lower visual importance,[6] typically the higher frequency subbands. The number of possible rates cannot exceed

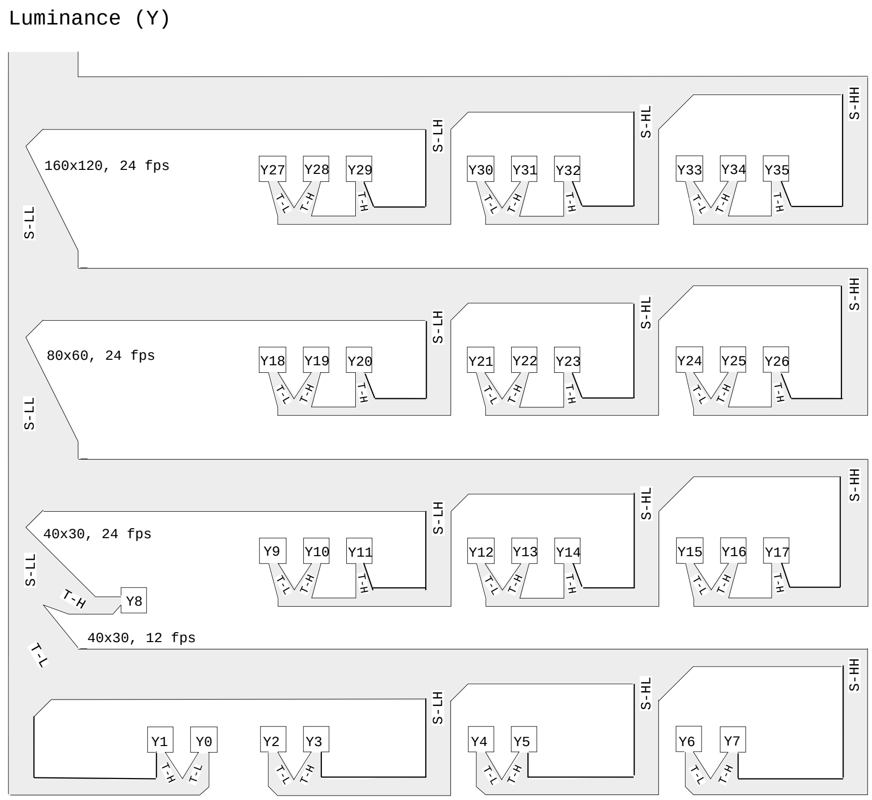

Figure 1. Spatio-temporal subband structure. The letters S and T stand for "Spatial" and "Temporal" decompositions. L and H stands for "Low" and "High" frequencies.

the number of subbands in such a scheme and because subbands are typically of very different sizes, rate granularity is not even.

Another approach is based on progressive quantization of subbands.[2] In such schemes, a set of embedded quantizers is used to quantize subband coefficients. For example, Fig. 2 shows one such set of quantizers $Q0$ through $Q3$ for quantizing coefficients that range from $-R$ to R. Quantizer $Q0$ has two bins and any value that falls between $-R$ and 0 is quantized to $-R/2$ while values that fall between 0 and R are quantized to $R/2$. Similarly, $Q1$ has four bins and any value that falls into each bin is quantized to the bin's middle value. If there is no prior knowledge of the distribution of the coefficient to be quantized, it takes one bit to record the output of quantizer $Q0$ and given we know which bin of $Q0$ we are in, it takes 1 more bit to record which $Q1$ bin we are in and so forth. Therefore, using outputs from the first few quantizers corresponds to a low rate representation of the original coefficients. The maximum error in using each quantizer is equal to half the bin width, hence, using more quantizers increases the fidelity of the representation. Because schemes based on progressive quantization can drop quantization layers rather than the whole subband, much finer rate granularity can be achieved.

For instance, the codec of Taubman *et.al.*[2] is capable of generating bit rates from tens of kilo bits per second to several mega bits per second with fine granularity of available bit rates. This codec is based on 3-D subband coding, progressive quantization of subband coefficients, followed by layered arithmetic coding, a high complexity coding method that nevertheless offers good compression. Because no motion estimation is involved, this codec also has sysmmetric complexity at the encoder and the decoder. However, its high complexity does not permit real-time software implementation on existing workstations.

Instead, in this paper we will consider the codec developed by E. Chang *et.al.*[1] that still enjoys the same fine granularity of bit rates but has much lower complexity. This codec is based on hierarchical block coding of Section 2.3.

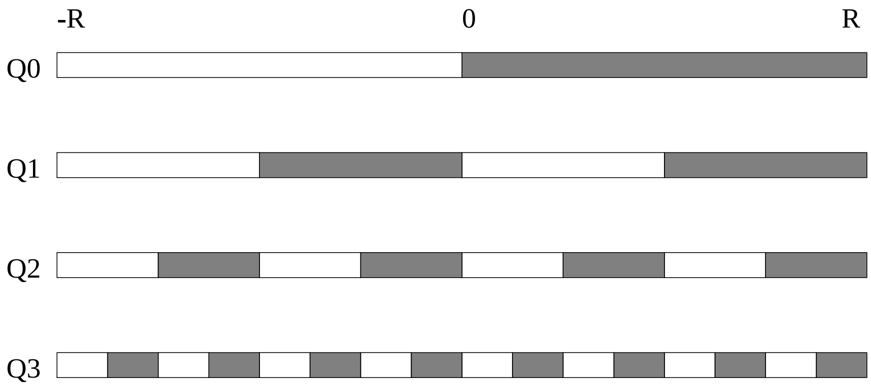

Figure 2. Embedded quantizers

instead of arithmetic coding and can achieve real time encoding and decoding in software. In the remainder of this section, we will consider various components of this codec.

2.1. 3-D subband decomposition

Three-dimensional subband analysis is first performed on the medical sequence to generate a set of spatio-temporal subbands. This is achieved by separable applications of one dimensional filter banks. One possible subband hierarchy is shown in Fig. 1.

2.2. Multi-rate quantization

Each subband coefficient is then progressively quantized using embedded deadzone quantizers. An example is shown in Fig. 3. The first quantizer consists of three bins where the middle bin is called *deadzone* because its quantized output is zero.

We choose a dead zone quantizer in which the width of the deadzone is twice as large as the width of each quantization bin, for all quantization layers, as shown in Fig. 3. In successive quantization of subband coefficients, each non-deadzone quantization bin is divided into two equal size bins. If a subband coefficient falls into the deadzone, it is quantized to zero, otherwise, it is *significant.* A significance map is then the binary map showing whether each subband coefficient is in the deadzone. Because many subband coefficients have values close to zero and fall into the deadzone, one way to code significance maps is to break it into equal size blocks and then use a variable length code that assigns short codewords to blocks with many zeros. One such variable length code is Kunt's block coding.[3]

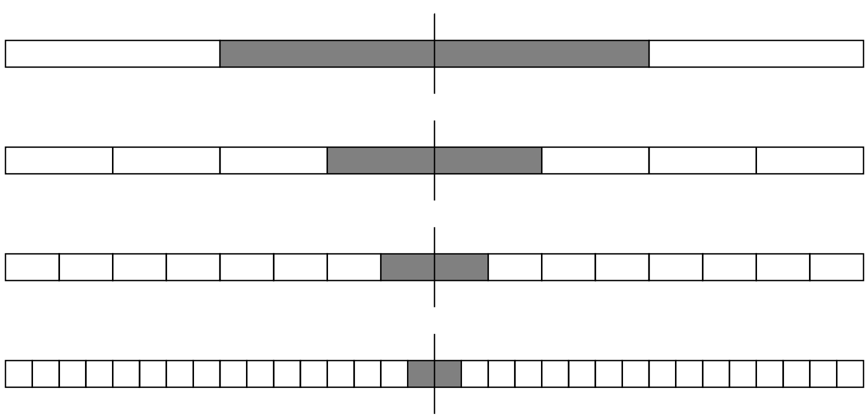

Figure 3. Deadzone quantizers

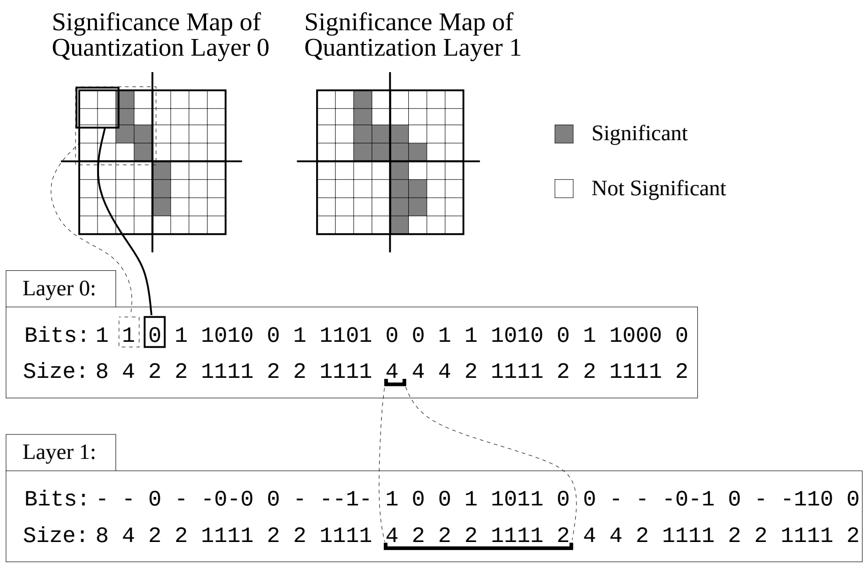

Figure 4. Example of two layers of block coding

2.3. Hierarchical block coding

Kunt's method begins by partitioning an image into 16×16 blocks. If a block contains all zeros, the block is coded as a "0", and the algorithm proceeds to the next block. Otherwise, the block codeword begins with a "1", and the block is subdivided into four 8×8 blocks, each of which are coded the same way. In this manner, the coding proceeds in a recursive manner until 1×1 blocks. This method is used to code the first layer only.

To code the next layer, we use the information in the previous layer to avoid coding redundant bits. Specifically, any bits that are marked "1" in the previous layer are also assumed to be "1" in the following layer. For example, consider layer 1 in Fig. 4. We assume the decoder has both bitstreams for layers 0 and 1, and layer 0 has been successfully decoded. To decode layer 1, the decoder cycles through the layer 0 bitstream again, filling in needed information as follows. The first task in decoding layer 1 is to decide whether or not the entire 8×8 block has any significant bits. Since layer 0 has significant bits, and layer 1 is a superset of layer 0, layer 1 must also have some significant bits. Therefore, the decoder assumes a "1" for the size 8 bit and does not require additional information. Since the decoder does not require additional information, the encoder will not send any; this is indicated by the "-" in the size 8 bit for layer 1, indicating that no bits are sent. The same process occurs for the first 4×4 block: the corresponding 4×4 block in layer 0 is significant, so nothing is coded for layer 1. The first 2×2 block, however, is empty in layer 0, and so the decoder does not know a priori whether the block contains any pixels in layer 1. Thus one bit must be sent to encode that information. As seen, the block is empty in layer 1 also, and so a 0 is coded for the size 2 bit in layer 1. Note that this is the first coded bit in the layer 1 bitstream, as it is the first piece of information that is not completely known from layer 0.

2.4. Overview of scalable coding

The flow chart in Fig. 5 summarizes the logical operations described in Section 2. The input video frames are first decomposed into spatio-temporal subbands according to the structure shown in Fig. 1. The subband coefficients are progressively quantized and then hierarchically encoded. The resulting bit stream is packetized for transmission, separating the various layers into separate network streams, and putting headers to allow the network to route and the decoder to combine the streams appropriately. The decoder reverses this process. It first removing the data from the packets. Then it performs heirarchical block decoding and subband reconstruction. These two operations are combined for the sake of efficiency. Finally, 3-D subband synthesis is performed to generate the video frames.

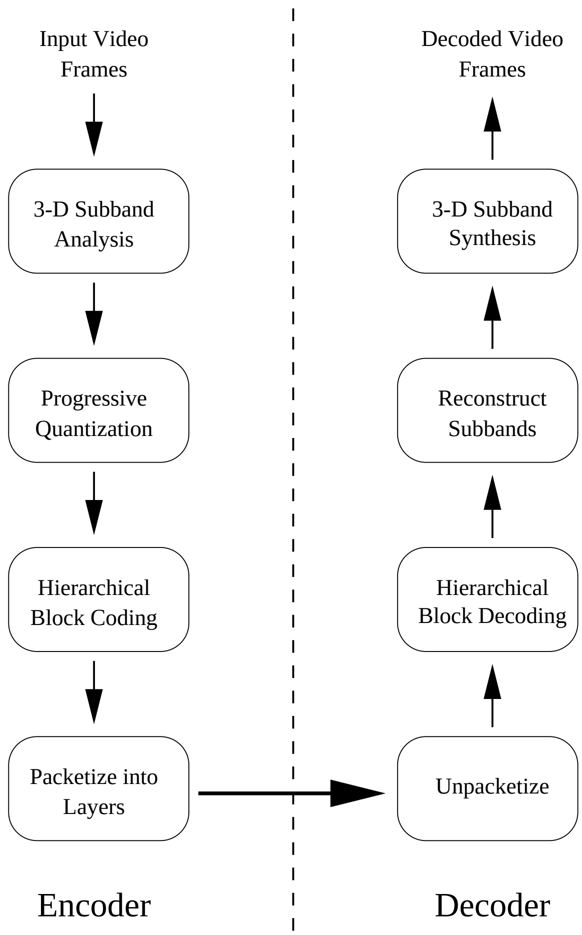

Figure 5. Flow chart of scalable coding

3. SIGNAL QUALITY

This section looks at a comparison of the video quality of images as a function of the bit-rate available in the network for the scalable and MPEG1 encoders. We use a test sequence consisting of cardiac cine-images, digitized to 8 bits of gray level per pixel, 256×240[†] pixels per frame, and 30 frames per second. Thus, the uncompressed video rate is about 14 Mbps.

After compression using the scalable encoder, we obtain a file containing a layered representation, consisting of 50 layers. Each layer is about 63.6 Kilobits per second. The relationship between the signal quality obtained by decoding a subset of the layers and the corresponding combined bit-rate is shown in Table 1. The signal quality is measured using the average Peak Signal to Noise Ratio (PSNR) metric. This is computed by taking the log of the mean square error between the decoded images and the original images, and averaging the result over all frames.

The MPEG1 measurements were performed using the public domain MPEG1 codec from the Plateau Multimedia Group at the University of California, Berkeley.[5] Since MPEG1 offers a great deal of flexibility in choice of Group of Pictures pattern for encoding, we performed the experiments for four different GOP patterns as shown in Table 2. We also encoded the video stream at five different target bit rates, resulting in twenty MPEG1 encoded files. These were then decoded, and the resulting images compared to the original images as described for the scalable encoder.

[†]The original images from the imaging equipment were captured at 512x480 pixels per frame. However, we chose to perform the comparison at a lower spatial resolution because of the limitations of the MPEG1 encoder we used for the comparison. The MPEG1 encoder was designed to compress SIF format images and did not compress well at the larger image sizes. The scalable encoder gave excellent results at the 512x480 image size.

Table 1. Rate vs. signal quality for Scalable encoding

Rate (Kbps)	PSNR
63.6	32.36
127.2	34.55
254.4	36.69
381.6	37.63
508.8	38.29
826.8	39.48
1017.6	39.94
1526.4	40.69

The PSNR values obtained are shown in Table 2. The MPEG PSNR obtained for GOP pattern of IP is compared to that of the scalable encoder in Fig. 6.

Table 2. Rate vs. signal quality for MPEG1 encoding with different GOP patterns (a) IP (b) IBPB (c) IBBPBB (d) IBBPBBPBBPBBPBB

Rate (Kbps)	PSNR (a)	PSNR (b)	PSNR (c)	PSNR (d)
256	34.66	36.76	36.59	36.49
500	38.30	38.22	38.36	37.70
750	39.40	39.01	39.00	38.26
1000	40.10	39.66	39.60	38.72
1500	41.02	40.65	40.30	39.31

We see from Table 2 that the video quality of MPEG1 varies with the choice of the GOP pattern. We obtain better signal quality with long GOP sequences at lower bit-rates and short GOP sequences at higher bit-rates. This is because I and P frames are naturally better at achieving very high quality, but require more bits. However, even if we consider the upper envelope of the MPEG1 curves (implying the best possible choice of GOP patterns at all bit-rates), we see that the scalable encoder achieves comparable quality. Note, that the scalable encoder was run only once, producing a layered bit stream, whereas the MPEG1 encoder was run multiple time, producing a separate encoded file for each target bit-stream. Yet, the video quality and bandwidth overhead of MPEG1 and the scalable encoder are comparable at all relevant bit-rates.

Table 3 shows the encoding and decoding speeds comparisons of MPEG1 (IBPB) and the scalable codec. The results were obtained using a 170 Mhz ultra sparc workstation. In both cases, the decoding speeds include disk access from a local disk and display, while the encoding speeds exclude disk access.

Table 3. Encoding and decoding speed comparisons (frames per second)

Rate (kb/s)	64	256	500	1000	1500
Scalable Decode	30.0	27.4	22.9	21.1	19.1
MPEG Decode	30.0	30.0	30.0	30.0	30.0
Scalable Encode	30.0	27.66	23.6	19.3	17.9
MPEG Encode (Exhaustive search)	0.5	0.5	0.5	0.5	0.5
MPEG Encode (Logarithmic search)	3.0	3.0	3.0	3.0	3.0

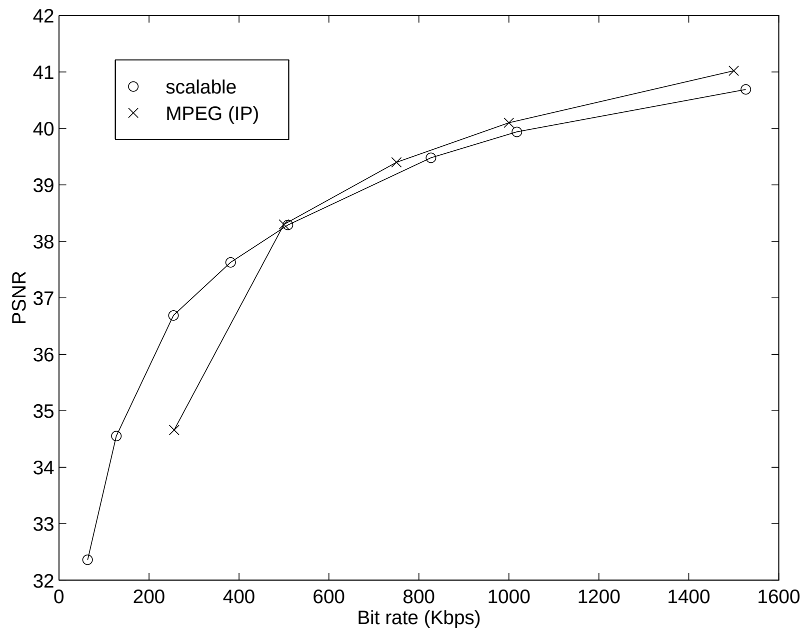

Figure 6. Comparison of Scalable and MPEG encoding

Unlike MPEG, the encoding and decoding complexity of the scalable codec is symmetric. The scalable encoder is at least an order of magnitude faster than exhaustive search MPEG1 encoding. The scalable decoder, however, is slower than the MPEG decoder but still runs in real-time (better than nineteen frames per second for a 1.5 Mbps encoded stream. For software implementations of real-time applications involving both encoding and decoding, such as video conferencing, the scalable coding technique is more promising than MPEG1.

4. ATM EXPERIMENTS

In this section, we describe some experiments describing the transmission of the layered encoded video over an ATM network with Quality of Service support. Regarding the video quality of the decoded signal, the different layers of the scalable video stream display different levels of sensitivity to errors and losses. Thus, when transferring video over a network with Quality of Service (QoS) support, such as an ATM network, it is useful to use different levels of QOS for the different layers. In these experiments, we use two levels of service priority in the ATM network to explore the benefits in terms of graceful degradation of video quality in the presence of network congestion.

The experiments were performed at the Philips Research Laboratory, Palo Alto, California on an Asynchronous Transfer Mode (ATM) fiber-optic Local Area Network (LAN) testbed consisting of a number of Sun workstations connected together by a Fore ASX-200 switch. Fig. 7 shows the physical topology of the testbed. Delhi, shanghai, and savant are Sun Ultra 1-170 workstations running Solaris 2.4, medfly is a Sun Sparcstation 20 running Solaris 2.4, and p8 is a Pentium-PC running Linux.

We set up the ATM experiments by partitioning the scalable video layers into two groups, which we call the base group and the enhancement group. The base group was transmitted on a ATM Virtual Circuit (VC) with reserved resources. The Virtual Circuit Identifier (VCI) of this VC was 110. We used the Constant Bit Rate (CBR) traffic class with appropriate peak rate and cell delay variance parameters to guarantee no loss. The enhancement layers were transmitted on a separate VC (VCI 111) with no reservations using the Unspecified Bit Rate (UBR) traffic class. These traffic classes were selected and their parameters set using the Usage Parameter Control (UPC) function of the Fore switch.

In order to emulate a network with multiple hops, with only one switch available for experimentation, we used a fiber looped back from one port of the switch to another as shown in Fig. 7. Both the VCs (110 and 111) were

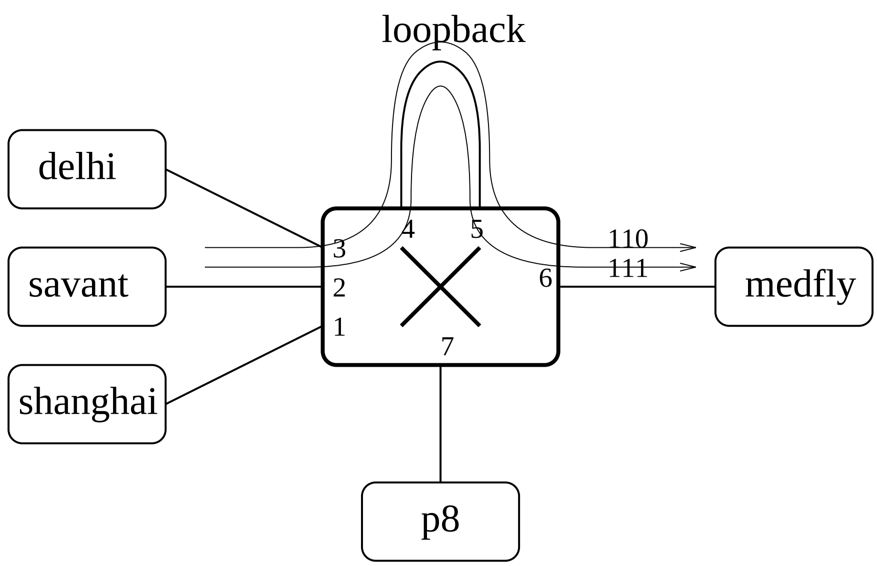

Figure 7. Experimental setup for ATM measurements

routed so as to go from the source machine (savant) to the switch, over the loopback fiber, back to the switch and then to the destination machine (medfly).

We introduced load from two machines (delhi and shanghai) to load the output port numbered 4 on the switch. We needed to use two machines because a single machine could not generate enough load to saturate the 155 Mbps (OC-3) output port. However, each machine could generate up to 130 Mbps (using UDP/IP over ATM), thus in tandem they could generate enough load to drive the OC-3 port to saturation. We routed the VC carrying the loading traffic over the loopback fiber and then to a different destination machine (p8). This way we isolated the effect of the load to a single point in the network. All other components (such as the source (savant) or destination (medfly) are not overloaded, thus cell loss only happens at one point, the output port number 4 in Fig. 7, when the output port buffer overflows.

The traffic from delhi and shanghai was generated based on a trace of real traffic the authors obtained from Richard Edell of UC Berkeley. The traffic trace is collected in the University of California (Berkeley) on January 24, 1996 from 1:00 p.m. to 2:00 p.m, on an FDDI ring in the EECS department. This trace had an average bit-rate of about 8 Mbps. In order to use this to generate higher loads, we merged this trace file with time shifted versions of itself, from one to ten times. This gave us a number of traces at multiples of approximately 8 Mbps. However, these traces consisted of a very large number of very small packets. Due to the per-packet overhead, we could not directly source such a large number of very small packets from one or two machines. In order to preserve the average characteristics of the traffic, we sacrificed the realism of our trace at small time intervals by merging adjacent packets if the due time of the later packet would arrive before the transmission of the earlier packet was complete. This merging process was based on a empirical model of ATM transmission, based on time to transmit packets of varying sizes.

The resulting traces allow us to generate load in real-time up to 80 Mbps. Using two machines we can generate any load level in multiples of 8Mbps up to 160Mbps, which completely saturates the OC-3 port.

The experiment consisted of varying the number of layers transmitted in the base group, while keeping the average total bit-rate the same. At the destination (medfly) the base and enhancement layers were merged, the resulting video stream decoded, and Peak Signal to Noise Ratio (PSNR) calculated by comparing the resulting video to the original video sequence. This was done at different levels of load in the network.

The coded sequence transmitted from the source was Variable Bit Rate (VBR), in order to keep the SNR of the encoded stream without loss as close to constant as possible. This makes it easier to see the variations in PSNR due to loss in the network. Because of this, the total number of layers transmitted varies with time, but the average number of layers transmitted is 24. This corresponds to a 1.5 Mbps stream.

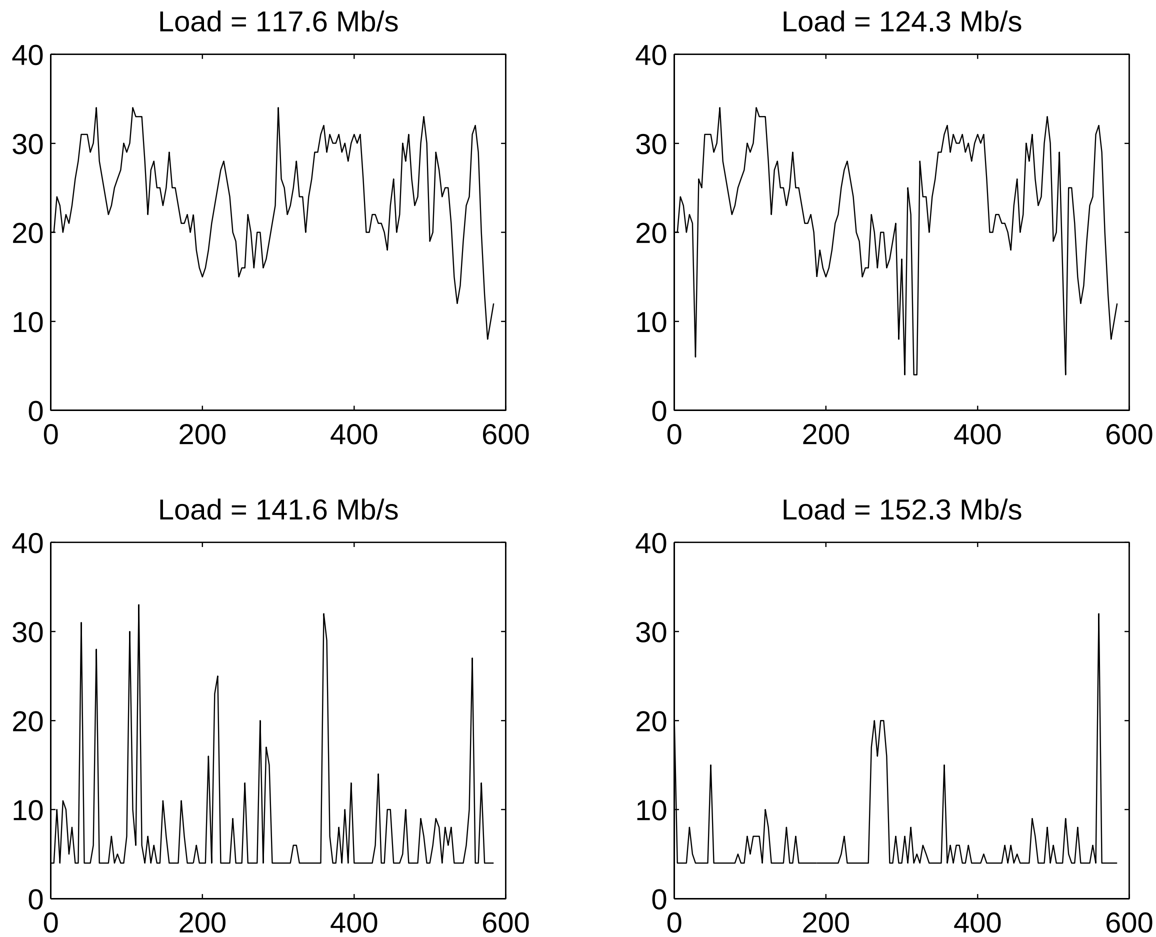

Figure 8. Number of received layers

The results of the experiments are shown in Figures 8 to 10. The upper left graph in Figure 8 shows the number of layers successfully decoded as a function of time in the absence of congestion in the ATM network. This implies that there was no loss in the network, and the variation in the number of layers decoded is attributed solely to the variation in the number of layers transmitted from the source. This variation is due to the fact that the video was VBR encoded. The other three graphs in Figure 8 show the same information for increasing levels of load in the network, for a partition corresponding to four base layers, and the rest of the layers transmitted on the best-effort VC. As the level of load increases, we notice losses in the network, so the number of layers decoded decreases. However, we can see that even at extremely high load, the base layer is not affected by the congestion. Thus, at least four layers always get through to the destination. Figure 9 shows the variation of PSNR as a function of time for the same set of experiments. The different lines correspond to experiment conducted at different load levels. A certain minimal image quality is maintained even in the presence of congestion due to the reservations on the VC carrying the base layers. This minimal quality corresponds to the PSNR achievable by a four layer (approximately 256 Kbps) video stream.

Graph 10 shows how the minimal guaranteed video quality can be improved by placing more layers into the base or reserved group. It shows the mean PSNR for different choices of the partition between the base and the enhancement groups. We can see that as we increase the number of layers in the base layer, the mean PSNR of the image sequence improves. Thus, depending on the minimal Quality of Service requirement of the application, we would chose a different partition between the base and enhancement layers. If the minimal acceptable QoS is very high, such as if the images are being used for diagnostic purposes, all the layers could be transmitted with high

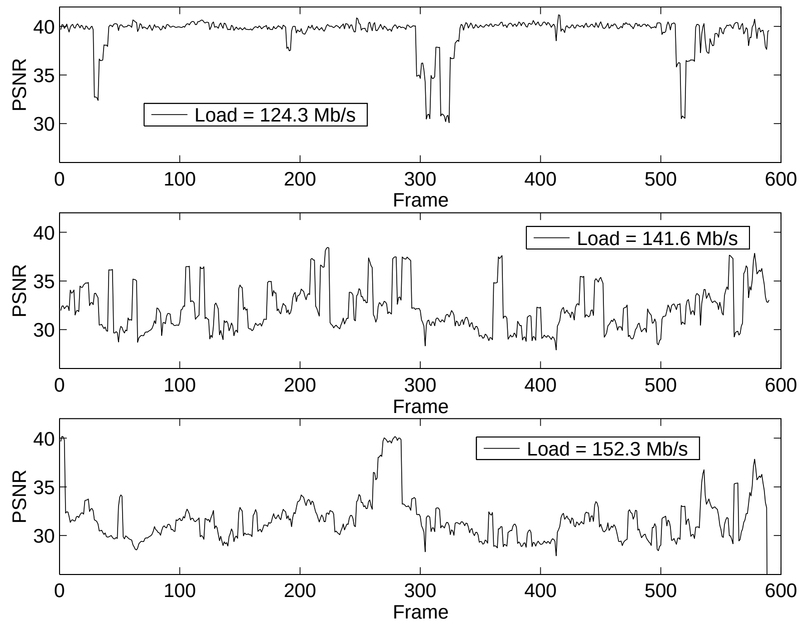

Figure 9. PSNR for four base layers

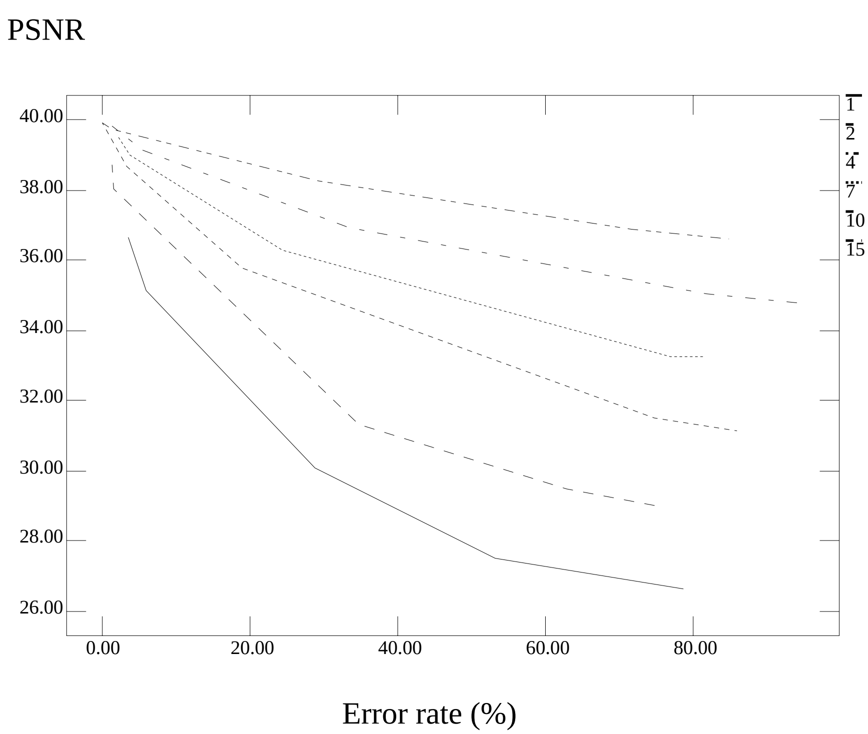

Figure 10. PSNR results summary

QoS support. Of course, the tradeoff of such a choice is in the cost of making the reservations for such high QoS communication, since the reserved resources cannot be shared with any other applications until the reservations are released. Thus, this graph allows the application to make cost versus quality trade-offs. If the expected loss rate in the network is known, the application can easily determine the expected video quality for each choice of the partition.

5. CONCLUSION

In this paper, we have presented the application of layered encoding to medical images by comparing the video quality obtained from the application of the scalable codec to that obtained from MPEG1 compression. The results show that the PSNR of the decoded video signal for the two codecs is comparable at different bit-rates, which leads us the conclude that the benefits of layering are obtained without significant bandwidth overhead.

We also presented some experimental results from the transmission of the layered video over an ATM network, using the quality of service features of the network to provide protection to the base layers. We observe that the signal quality degrades gracefully in the presence of cell loss in the network, and the minimal acceptable quality can be traded off against the reserved bandwidth, if the expected loss rates in the network is known.

We are currently working on network support for a video conferencing application which uses the layered video encoding to allow participation of heterogeneous receivers across heterogeneous networks. We are also exploring the use of the layered encoder, in conjunction with differential Quality of Service support or multi-level forward error correction, to provide graceful degradation in lossy environments.

REFERENCES

1. W. Tan, E. Chang, and A. Zakhor, "Real Time Software Implementation of Scalable Video Codec," *Proc. Int. Conf Image Processing,* Lausanne, Switzerland, September 1996.
2. D. Taubman and A. Zakhor, "Multirate 3D Subband Coding of Video," *IEEE Trans. Image Processing,* Vol. 3, No. 5, September 1994.
3. M. Kunt, "Block Coding of Graphics: A Tutorial Review," *Proceedings of the IEEE*, Vol. 68, No. 7, July 1980
4. M.Ghanbari. Two-Layer Coding of Video Signals for VBR Networks. *IEEE Trans. Selected Areas in Commun.*, Vol. 7, No. 5, June 1989.
5. K.Patel, B.Smith and L.Rowe, Performance of a Software MPEG Video Decoder, *Proc. ACM Multimedia '93.*
6. M.Uz, M.Vetterli and D.LeGall. Interpolative multi-resolution coding of advanced television with comparable subchannels. *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 1, No. 1, Mar 1991.