

Semiannual Status Report

**Implementation Issues in
Source Coding**

NAG5-916

(NASA Grant NAG 5-916)

Submitted to

Instrument Division
Engineering Directorate
Goddard Space Flight Center
Greenbelt, MD 20771
Attn: Warner Miller and Pen Shu Yeh

K. Sayood, Y. C. Chen, and A. C. Hadenfeldt
Department of Electrical Engineering
Center for Communication and Information Science
University of Nebraska-Lincoln

1 Introduction

The research during this reporting period can be divided into three areas:

- Development of image coding algorithm for lossy and lossless coding.
- Development of a packet video simulator, and investigation of coding algorithms for packet video.
- Development of implementation strategies for coding algorithms.

We have had considerable progress in the first two areas. Detailed reports on these are included as appendices. We are in the initial stages of the third area and will report on them later.

2 Development of Image Coding Scheme

We have developed an edge preserving image coding scheme which can be operated in both a lossy and a lossless manner. The technique is an extension of the lossless encoding algorithm developed for the Mars observer spectral data [1]. It can also be viewed as a modification of the well known DPCM algorithm. As the DPCM algorithm has already been implemented at high speeds [2], this algorithm could be used in environments where fast processing is desired. It has minimal memory requirements, which make it suitable for situations in which there are size and weight restrictions on the instrument. The algorithm has the highly desirable property of preserving edges. This is a necessity if the algorithm is to be used for the compression of scientific data where the preservation of the edges is a must. Finally, the algorithm can be operated at different rates under user or system control which makes it suitable for implementation over a variable rate channel. The current status of the algorithm is described in some detail in appendix A. This appendix is a paper which is to be presented at the 1989 Phoenix Conference on Computers and Communication.

The algorithm in its current form has a fixed predictor and quantizer. The next step in its development is the inclusion of an adaptive predictor and an adaptive quantizer. Of special interest is the inclusion of an ARMA predictor which we had previously shown to be effective in the reduction of edge degradation [3] because of more efficient prediction of edge pixels. While the current algorithm is edge preserving in nature, it does pay for inefficient prediction of edge pixels by

an increase in rate. A strategy which would reduce the prediction error would result in a reduction in rate. To further reduce the rate we also propose to develop a modified run-length coding scheme similar to the one developed for the Mars Observer spectral data. Finally, we are developing a perceptual testing methodology for evaluating the perceptual quality of the reconstructed coded image.

3 Packet Video

We have modified an existing packet network simulator to function as a packet video simulator. While some further modifications are necessary to fully simulate the different environments under which a packet video should operate, the system is functional enough to test a packet video system. The proposed coding scheme for the packet video system is a modification of the mixture block coding (MBC) scheme described in the last report. Details of the coding scheme and the simulator are presented in Appendices B and C. Appendix B is a draft copy of an MS thesis, while appendix C is the first draft of a paper to be submitted to the IEEE Transactions on Communications.

The MBC coding scheme allows the efficient use of the channel as described in the appendices. However, there is one provision that has not yet been implemented, and that is the control of the coding rate by the channel conditions. By this, we mean that when the available channel capacity is low, the scheme should operate at a lower rate than when the available capacity is high. This can be done by adjusting the threshold rate used by MBC as a function of some channel parameter. The channel parameter being examined for this role is a function of the delay information available at each node.

The efficiency of coding in packet video has to be evaluated perceptually. This is because when viewed as a motion sequence certain distortions which were apparent in a frame-by-frame viewing get masked while other distortions which were not apparent before may now be very clear. We are currently working on developing a system for the viewing and grabbing motion video sequences. The MBC scheme is somewhat inefficient in terms of computation and memory requirements. We are examining different approaches to alleviate this problem and make the system amenable to real-time implementation. Finally, the coding rate for this system can be substantially reduced by the use of motion compensation strategies. We are examining different strategies, in terms of coding rate, as well as complexity and robustness to channel errors.

4 References

- [1] K. Sayood and M. C. Rost, "A Robust Compression System for Low Bit Rate Telemetry—Test Results with Lunar Data," *Proceedings of the Scientific Data Compression Workshop*, Snowbird, Utah, pp. 237–250, May 1985.
- [2] W. H. Miller, "Image Processing using Gallium Arsenide (GaAs) Technology," *Proceedings of the Scientific Data Compression Workshop*, Snowbird, Utah, pp. 451–466, May 1988.
- [3] K. Sayood and S. M. Schekall, "Use of ARMA Predictors in the Differential Encoding of Images," *IEEE Trans. on Acoust., Speech, and Signal Processing*, Vol. ASSP-36, pp. 1791–1795, Nov. 1988.

5 Appendix A

An Edge Preserving Differential Image Coding Scheme¹

Martin C. Rost* and Khalid Sayood†

*Sandia National Laboratories

Albuquerque, NM

†Department of Electrical Engineering and

Center for Communication and Information Science

University of Nebraska

Lincoln, NE 68588-0511

(402) 472-6688

Abstract

Differential encoding techniques are fast and easy to implement. However, a major problem with the use of differential encoding for images is the rapid edge degradation encountered when using such systems. This makes differential encoding techniques of limited utility especially when coding medical or scientific images, where edge preservation is of utmost importance. We present a simple, easy to implement differential image coding system with excellent edge preservation properties. The coding system can be used over variable rate channels which makes it especially attractive for use in the packet network environment.

¹This work was supported by the NASA Goddard Space Flight Center under Grant NAG 5-916.

Please address all correspondence to Khalid Sayood, Department of Electrical Engineering, University of Nebraska, Lincoln, NE 68588-0511.

1 Introduction

The transmission and storage of digital images require an enormous expenditure of resources, necessitating the use of compression techniques. These techniques include relatively low complexity predictive techniques such as Adaptive Differential Pulse Code Modulation (ADPCM) and its variations, as well as relatively higher complexity techniques such as transform coding and vector quantization [1,2]. Most compression schemes were originally developed for speech and their application to images is at times problematic. This is especially true of the low complexity predictive techniques. A good example of this is highly popular ADPCM scheme. Originally designed for speech [3], it has been used with other sources with varying degrees of success. A major problem with its use in image coding is the rapid degradation in quality whenever an edge is encountered. Edges are perceptually very important and occur quite often in most images. Therefore, the degradation of edges can be perceptually very annoying. If the images under consideration are medical or scientific, the problem becomes even more important, as edges provide position information which may be crucial to the viewer. This poor edge reconstruction quality has been a major factor in preventing ADPCM from becoming as popular for image coding as it is for speech coding.

While good edge reconstruction capability is an important requirement for image coding schemes, another requirement that is gaining in importance with the proliferation of packet switched networks, is the ability to encode the image at different rates. In a packet switched network, the available channel capacity is not a fixed quantity, but rather fluctuates as a function of the load on the network. The compression scheme must therefore be capable of operating at different rates as the available capacity changes. This means that it should be able to take advantage of increased capacity when it becomes available while providing graceful degradation when the rate decreases

to match decreased available capacity.

In this paper we describe a DPCM based coding scheme which has the desired properties listed above. It is a low complexity scheme with excellent edge preservation in the reconstructed image. It takes full advantage of the available channel capacity providing lossless compression when sufficient capacity is available, and very graceful degradation when a reduction in rate is required.

2 Notation and Problem Formulation

The DPCM system consists of two main blocks, the quantizer and the predictor (see Fig. 1). The predictor uses the correlation between samples of the waveform to predict the next value. This predicted value is removed from the waveform at the transmitter and reintroduced at the receiver. The prediction error is quantized to one of a finite number of values which is coded and transmitted to the receiver. The difference between the prediction error and the quantized prediction error is called the quantization error or the quantization noise. If the channel is error free, the reconstruction error at the receiver is simply the quantization error. To see this, note from Figure 1. that the prediction error $e(k)$ is given by

$$e(k) = s(k) - p(k) \quad (1)$$

where the predicted value is given by

$$p(k) = \sum a_j \hat{s}(k - j) \quad (2)$$

and

$$\hat{s}(k) = e_q(k) + p(k). \quad (3)$$

Assuming an additive noise model, the quantized prediction error $e_q(k)$ can be represented as

$$e_q(k) = e(k) + n_q(k) \quad (4)$$

where $n_q(k)$ denotes the quantization noise. The quantized prediction error is coded and transmitted to the receiver. If the channel is noisy this is received as $\tilde{e}_q(k)$ which is given by

$$\tilde{e}_q(k) = e_q(k) + n_c(k) \quad (5)$$

where $n_c(k)$ represents the channel noise. The output of the receiver $\tilde{s}(k)$ is thus given by

$$\tilde{s}(k) = \tilde{p}(k) + \tilde{e}_q(k) \quad (6)$$

where

$$\tilde{p}(k) = p(k) + p_n(k) \quad (7)$$

the additional term $p_n(k)$ being the result of the introduction of channel noise into the prediction process. Using (1), (4), (5), and (7) in (6) we obtain

$$\tilde{s}(k) = s(k) + n_q(k) + n_c(k) + p_n(k). \quad (8)$$

If the channel is error free, the last two terms in (8) drop out and the difference between the original and reconstructed signal is simply the quantization error.

When the prediction error is small, it falls into one of the inner levels of the quantizer, and the quantization noise is of a type referred to as granular noise. If the prediction error falls in one of the outer levels of the quantizer, the incurred quantization error is called overload noise. Because of the way the granular noise is generated it is generally smaller in magnitude than the overload noise and is bounded by the size of the quantization interval. The overload noise on the other hand is essentially unbounded and can become very large depending on the size of the prediction error. As edge pixels are rather difficult to predict, the corresponding prediction error is generally large, and this leads to a large overload noise value. Furthermore, because this error affects not only the reconstruction of the current pixel, but also future predictions, the prediction errors corresponding to the next few pixels also tend to be large, leading to a “smearing” out effect.

Reduction of the edge degradation can therefore be obtained by reducing or eliminating the slope overload noise. Reduction of the slope overload noise can be obtained by improving the prediction process Gibson [4] analyzed ADPCM systems with backward adaptive prediction, and showed that the tracking ability of the adaptive predictor can be improved by the addition of zeros. Motivated by these results, Sayood and Schekall [5] designed ADPCM systems for image coding with ARMA predictors. These results clearly show that some reduction in the edge degradation is possible with the use of adaptive zeros in the predictor. While the use of these predictors improves the edge reconstruction there is still significant degradation in the edges. One technique to further improve the edge performance was developed by Schekall and Sayood [6], which uses the Jayant quantizer as an edge detector. The overload noise is then reduced by sending a quantized representation

of the noise through a side channel. The advantage of this approach is that it can be added on to existing ADPCM systems. The disadvantage is the use of the side channel which introduces synchronization problems. In this paper we propose a different approach for edge preservation which does not require a side channel. This approach is described in the following section.

3 Proposed Approach

The approach taken in this paper is a variation on the standard rate-distortion tradeoff. The basic idea is that the slope overload noise can be reduced by increasing the rate. However rather than increasing the rate for encoding each and every pixel, there is only an instantaneous rate increase whenever slope overload is encountered. The way this is implemented is outlined in the block diagram of Figure 2. A DPCM system is followed by a lossless encoder at the transmitter. At the receiver the inverse operations are performed. The DPCM system differs from standard DPCM systems in that the quantizer being used has an unlimited number of levels. In practice what this means is that if the input has 256 levels, which is standard for monochrome images, then the DPCM quantizer will have 512 levels. This effectively eliminates the overload noise making the distortion a function of the quantizer stepsize Δ . Of course by itself it also eliminates any compression that may have been desired, in fact it requires an increase of one bit in the rate. The compression is obtained by use of the lossless encoder. The lossless encoder output alphabet consists of N codewords. These codewords correspond to N consecutive levels in the quantizer. Let the smallest level be labeled x_L and the largest level be labeled x_H . If the quantizer output $e_q(k)$ is a level between x_L and x_H , then the lossless encoder puts out the corresponding channel symbol. If, however $e_q(k)$ is greater than x_H the encoder puts out the symbol corresponding to x_H . A new value $e_{q1}(k)$ is then obtained by

subtracting x_H from $e_q(k)$. If this value is less than x_H then it is encoded using the corresponding codeword in the lossless encoder output alphabet. Otherwise, x_H is again subtracted from $e_{q1}(k)$ to generate $e_{q2}(k)$. This process is continued till some $e_{qn}(k)$ where

$$e_{qn}(k) = e_q(k) - nx_H$$

and $e_{qn}(k)$ is less than x_H . A similar strategy is followed when $e_q(k) \leq x_L$. Thus the instantaneous rate is increased by a function of n whenever the prediction error falls outside the closed interval $[x_L, x_H]$.

Example : Consider a DPCM system with a stepsize Δ of 2 where the input output relationship is given by

$$Q[x] = 2k \quad \text{if} \quad 2k - 1 \leq x < 2k + 1; \quad k = 0, \pm 1, \pm 2, \dots$$

Let the lossless encoder output alphabet be of size eight with $x_L = -4$, and $x_H = 10$. If the input $e(k)$ is 7, the output $e_q(k) = 8$ which is in the lossless encoder output alphabet. If $e(k) = 15$, then $e_q(k)$ is 16 which is larger than x_H . In this case, the encoder puts out the codeword corresponding to x_H and generates $e_{q1}(k) = 16 - 10 = 6$ which is in the encoder output alphabet. If the input is -7 , $e_q(k) = -6$ which is less than x_L . Thus the lossless encoder output consists of two symbols. One corresponding to the value of $x_L(-4)$ and one corresponding to the value of -2 . Note that if the input is 10 or -4 (i.e. x_H or x_L) then the output will be the sequence 10, 0 or $-4, 0$.

One of the consequences of this type of encoding is the generation of runs of x_L and x_H whenever the image contains a large number of edges. Fortunately the encoding scheme also provides a

significant number of special symbols that can be used to encode the runlengths. For example, the sequence x_H followed by a negative value and the sequence x_L followed by a positive value would not occur in the normal course of events. These sequences can therefore be used to encode the runlengths of x_L and x_H . Furthermore these special sequences can also be used to signal a change in rate. A change in rate can be obtained in two different ways. Either by restricting the number of levels or by changing the stepsize Δ of the quantizer.

Several of the systems proposed above were simulated. The results of these simulations are presented in the next section.

4 Results

Two systems of the type described in the previous section have been simulated. Another two are in the process of being simulated and results from these will be available shortly. The two systems already simulated, both use a one tap fixed predictor. One of the systems contains the lossless encoder followed by a runlength encoder while the other contains only the lossless encoder without the runlength encoder. The test images used were the USC GIRL image, and the USC COUPLE image. Both are 256 X 256 monochrome eight bit images and have been used often as test images. The objective performance measure were the Peak Signal to Noise Ratio (PSNR) and the Mean Absolute Error (MAE) which are defined as follows:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\langle (s(k)) - \tilde{s}(k) \rangle^2}$$

$$\text{MAE} = \langle |s(k) - \tilde{s}(k)| \rangle$$

Several initial test runs were performed using different number of levels, different values of x_L and different values of Δ to get a feel for the optimum values of the various parameters. We found that an appropriate way of selecting the value of x_L was using the relationship

$$x_L = -\lfloor \frac{N-1}{2} \rfloor \Delta$$

where $\lfloor x \rfloor$ is the largest integer less than or equal to x , and N is the size of the alphabet of the lossless coder. This provides a symmetric codebook when the alphabet size is odd, and a codebook skewed to the positive side when the alphabet size is even. The zero value is always in the codebook.

As the alphabet size is usually not a power of two, the binary code for the output alphabet will be a variable length code. The use of variable length codes always bring up issues of robustness. With this in mind, the rate was calculated in two different ways. The first was to find the output entropy, and scale it up by the ratio of symbols transmitted to the number of pixels encoded. We call this rate the entropy rate, which is the minimum rate obtainable, if we assume the output of the lossless encoder to be memoryless. While this assumption is not necessarily true, the entropy rate gives us an idea about the best we can do with a particular system. We will treat it as the lower bound on the obtainable rate. We also calculated the rate using a predetermined variable length code. This code was designed with no prior knowledge of the probabilities of the different letters. The only assumption was that the letters representing the inner levels of the quantizer were always more likely than the letters representing the outer levels of the quantizer. The code tree used is shown in Figure 3. Obviously, this will become highly inefficient in the case of small alphabet size and small Δ , as in this case, the outer levels x_L and x_H will occur quite frequently.

This rate can be viewed as an upper bound on the achievable rate.

The results for the system without the runlength encoder are shown in Tables 1 and 2. Table 1 contains the results for the COUPLE image, while Table 2 contains the results for the GIRL image. Recall that for image compression schemes, systems with PSNR values of greater than 35 dB are perceptually almost identical. As can be seen from the PSNR values in the tables there is very little degradation with rate, and in fact if we use the 35 dB criterion there is almost no degradation in image quality until the rate drops below two bits per pixel. This can be verified by the reconstructed images shown in Figure 4. It is extremely difficult to tell the images apart, even though the rate varies from 4. bits per pixel to 1. bits per pixel. To remove the effect of the photographic process itself, we are in the process of setting up perceptual tests to obtain a more subjective evaluation of the perceived degradation. A variable rate system was constructed where the rate was changed during transmission. The perceptual tests will also be used to determine whether the viewer can perceive the transitions between various rates in an image.

We can see from the results that if the value of Δ and hence x_L is fixed, the size of the codebook has no effect in on the performance measures. This is because the only effect of reducing the codebook size under these conditions is to increase the number of symbols transmitted. While this has the effect of increasing the rate, because of the way the system is constructed it does not influence the resulting distortion. The drop in rate for the same distortion as the alphabet size increases can be clearly seen from the results in Tables 1 and 2.

Table 3 shows the decrease in rate when a simple runlength coder is used. The runlength coder encodes long strings of x_L and x_H using the special sequences mentioned previously. As can be seen from the results the improvement provided by the current runlength encoding scheme is significant

only for small alphabets and small values of Δ . This is because it is under these conditions that most of the long strings of x_L and x_H are generated. However we are not as yet using many of the special sequences in the larger alphabet codebooks, so there is certainly room for improvement.

The two other systems currently being simulated are systems in which one tap predictor is replaced with fixed and adaptive ARMA predictors. Based on our previous results, we feel that this will reduce the number of prediction error values that will lie outside the interval $[x_L, x_H]$ and therefor result in a reduction of the rate.

5 Conclusion

We provide a simple image coding scheme which is very easy to implement in realtime and has excellent edge preservation properties over a range of rates.

This system would be especially useful in transmitting images over channels where the available bandwidth may be vary. The edge preserving quality would be especially useful in the encoding of scientific and medical images.

6 References

- [1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [2] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349-389, Mar. 1981.
- [3] C. C. Cutler, "Differential quantization for communication signals," U.S. Patent 2 605 361, July 29, 1952.
- [4] J. D. Gibson, "Backward adaptive prediction as spectral analysis within a closed loop," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1166-1174, Oct. 1985.
- [5] K. Sayood and S. M. Schekall, "Use of ARMA Predictors in the Differential Encoding of Images," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-36, pp. 1791-1795, Nov. 1988.
- [6] S. M. Schekall and K. Sayood, "An Edge Preserving DPCM Scheme for Image Coding," *Proc. 31st Midwest Symposium on Circuits and Systems*, St. Louis, pp. 904-907, Aug. 1988.

le 1(a). Performance Results for the COUPLE image, alphabet size = 3

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.5067	51.0830	6.1615	7.1418
	1.4790	42.7898	3.8909	4.0587
	2.4676	38.6565	2.9577	3.0137
	3.3697	36.0009	2.4314	2.4972
	5.1359	32.3682	1.8277	1.9800

le 1(b). Performance Results for the COUPLE image, alphabet size = 4

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.5067	51.0830	5.4968	7.1118
	1.4790	42.7898	3.6148	3.9903
	2.4676	38.6565	2.8040	2.9367
	3.3697	36.0009	2.3324	2.4224
	5.1359	32.3682	1.7765	1.9157

le 1(c). Performance Results for the COUPLE image, alphabet size = 5

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.5067	51.0830	4.9334	6.8635
	1.4790	42.7898	3.3637	3.7982
	2.4676	38.6565	2.6553	2.7729
	3.3697	36.0009	2.2327	2.2756
	5.1359	32.3682	1.7233	1.7963

le 1(d). Performance Results for the COUPLE image, alphabet size = 6

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
2	0.5067	51.0830	4.7338	6.7528
4	1.4790	42.7898	3.2860	3.7436
5	2.4676	38.6565	2.6139	2.7401
3	3.3697	36.0009	2.2067	2.2554
2	5.1359	32.3682	1.7118	1.7855

le 1(e). Performance Results for the COUPLE image, alphabet size = 7

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.5067	51.0830	4.5324	6.7822
	1.4790	42.7898	3.2020	3.7248
	2.4676	38.6565	2.5678	2.7172
	3.3697	36.0009	2.1775	2.2350
	2.9033	32.3682	1.6982	1.7698

le 1(f). Performance Results for the COUPLE image, alphabet size = 8

ca	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.5067	51.0830	4.4404	6.6884
	1.4790	42.7898	3.1673	3.6939
	2.4676	38.6565	2.5490	2.7023
	3.3697	36.0009	2.1662	2.2267
	5.1359	32.3682	1.6930	1.7669

le 2(a). Performance Results for the GIRL image, alphabet size = 3

ta	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	6.2821	7.8120
	1.4889	42.7206	4.0088	4.3976
	2.4847	38.5513	3.0819	3.2547
	3.5086	35.6855	2.5543	2.6860
	5.5074	31.8820	1.9426	2.1122

le 2(b). Performance Results for the GIRL image, alphabet size = 4

ta	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	5.6677	7.3303
	1.4889	42.7206	3.7481	4.0803
	2.4847	38.5513	2.9262	2.9964
	3.5086	35.6855	2.4442	2.4645
	5.5074	31.8820	1.8709	1.9373

le 2(c). Performance Results for the GIRL image, alphabet size = 5

ta	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	5.0554	7.4713
	1.4889	42.7206	3.4714	4.0592
	2.4847	38.5513	2.7570	2.9279
	3.5086	35.6855	2.3272	2.3783
	5.5074	31.8820	1.8046	1.8439

le 2(d). Performance Results for the GIRL image, alphabet size = 6

α	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	4.8664	7.1315
	1.4889	42.7206	3.3889	3.9006
	2.4847	38.5513	2.7097	2.8325
	3.5086	35.6855	2.2972	2.3147
	5.5074	31.8820	1.7917	1.8138

le 2(e). Performance Results for the GIRL image, alphabet size = 7

α	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	4.6531	7.3549
	1.4889	42.7206	3.3025	3.9549
	2.4847	38.5513	2.6646	2.8433
	3.5086	35.6855	2.2707	2.3110
	5.5074	31.8820	1.7809	1.8053

le 2(f). Performance Results for the GIRL image, alphabet size = 8

α	MAE	PSNR (dB)	Entropy Rate (Lower Bound)	Average Length (Upper Bound)
	0.4968	51.1693	4.5635	7.1275
	1.4889	42.7206	3.2668	3.8740
	2.4847	38.5513	2.6468	2.8063
	3.5086	35.6855	2.2617	2.2931
	5.5074	31.8820	1.7786	1.8009

le 3. Comparison of Entropy rates between system with Runlength (RL) Encoder and without RL Encoder for COUPLE image.

Number of Levels = 3		Number of Levels = 5		Number of Levels = 8	
Without RL Encoder	With RL Encoder	Without RL Encoder	With RL Encoder	Without RL Encoder	With RL Encoder
6.16	5.44	4.93	4.34	4.44	4.29
3.89	3.60	3.36	3.25	3.16	3.15
2.96	2.81	2.66	2.63	2.55	2.55
2.43	2.35	2.23	2.22	2.17	2.17
1.83	1.80	1.72	1.72	1.69	1.69

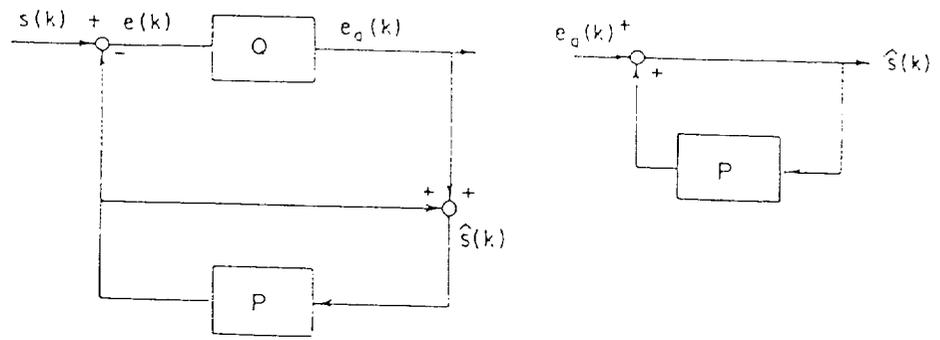


Fig. 1. DPCM structure.

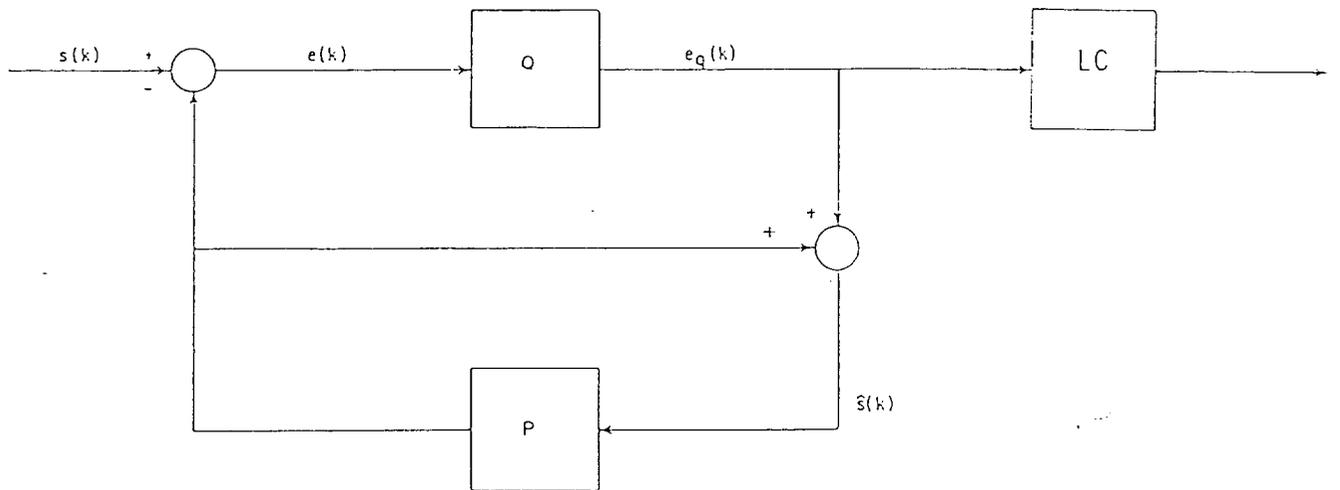


Fig. 2 Proposed Encoder Structure

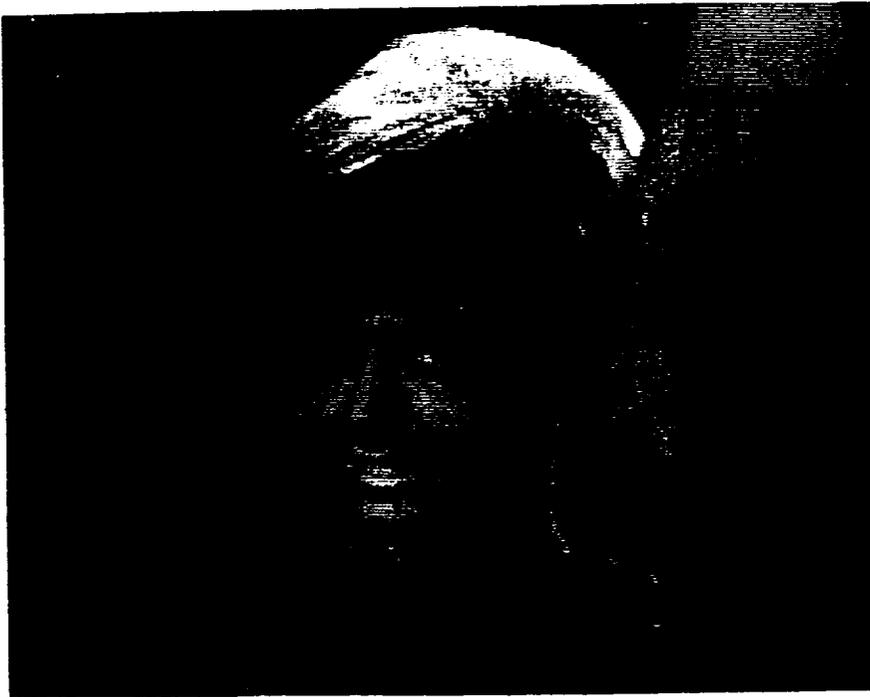


Fig. 4(a). GIRL image coded with Entropy rate 4.56 bpp
and Average Length 7.13 bpp

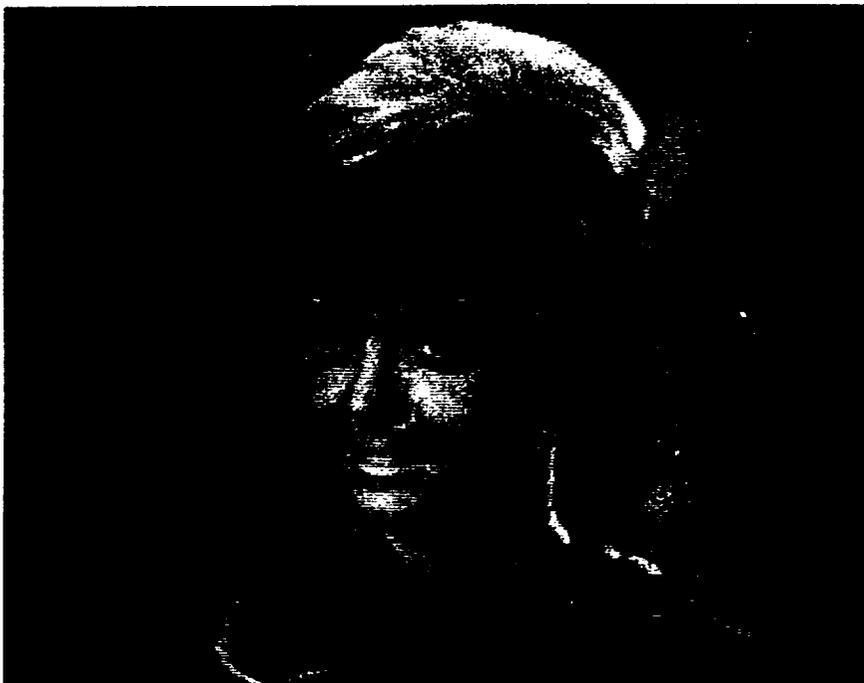


Fig. 4(b). GIRL image coded with Entropy rate 2.65 bpp
and Average Length 3.87 bpp

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

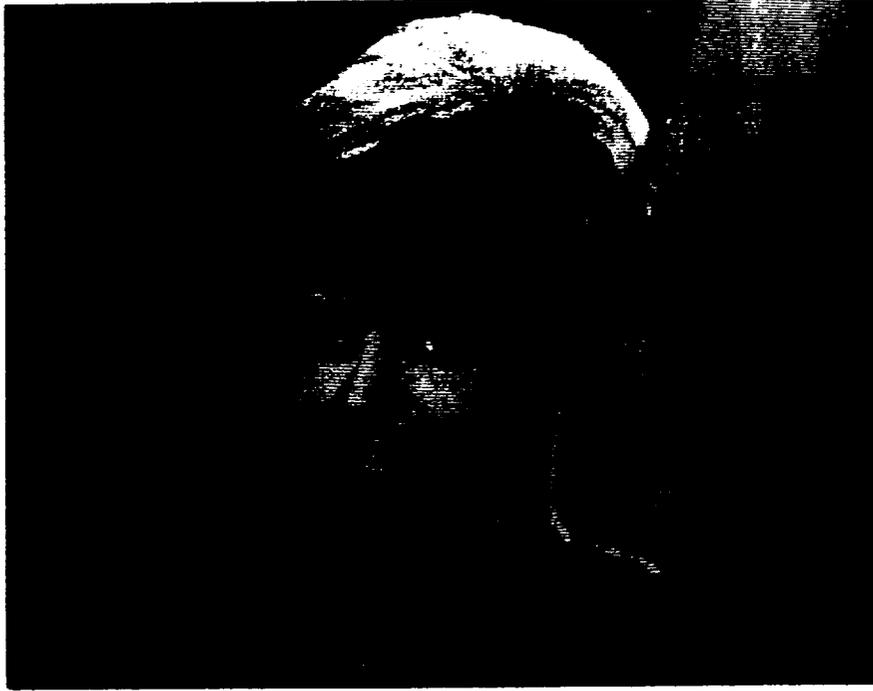


Fig. 4(c). GIRL image coded with Entropy rate 1.77 bpp
and Average Length 1.80 bpp

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

6 Appendix B

MIXTURE BLOCK CODING WITH PROGRESSIVE TRANSMISSION
IN PACKET VIDEO

by

Yun-Chung Chen

A THESIS

Presented to the Faculty of
The Graduate College in the University of Nebraska
In Partial Fulfillment of Requirements
For the Degrees of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

August, 1989

**Mixture Block Coding with Progressive Transmission
in Packet Video**

Table of Contents

Abstract.....	i
List of Figures.....	ii
List of Tables.....	iii
Chapter 1. Introduction.....	1
Chapter 2. Packet Video.....	4
2.1 Broadband Integrated Service Digital Network	
2.2 Video Transmission over Packet-Switching Network	
2.3 Interaction between Signal Processing and networking	
Chapter 3. Image Data Compression and Mixture Block Coding with Progressive Transmission.....	10
3.1 Image Data Compression	
3.2 Quantization	
3.2.1 Scaler Quantization	
3.2.2 Vector Quantization	
3.3 Transform Coding and DCT	
3.3.1 Transform Coding	
3.3.2 Discrete Cosine Transform	
3.4 Mixture Block Coding with Progressive Transmission	
3.4.1 Progressive Coding	
3.4.2 Structure of MBCPT Coder	
3.4.3 Designing Consideration	
3.4.4 Distortion and Blocking Effect	
3.4.5 Application in Packet-Switching Network	
Chapter 4. Network Simulator.....	36

4.1	Introduction	
4.1.1	Topology, Traffic and Preparation	
4.1.2	Simulation Philosophy	
4.1.3	Simulator Queue and Queue Processor	
4.2	The Network Layers	
4.2.1	The Session Layer	
4.2.2	The Transport Layer	
4.2.3	The Network Layer	
4.2.4	The Datalink Layer	
4.3	Modifications	
Chapter 5.	Simulation Results.....	47
5.1	Differential Interframe Coding	
5.2	Interaction of the Coder and the Network	
5.2.1	Packetization	
5.2.2	Error Recovery	
5.2.3	Flow Control	
5.2.4	Resynchronization	
5.2.5	Interaction with Protocols	
5.3	Results from Packet Video Simulation	
Chapter 6.	Conclusions.....	73
References.....		75

List of Figures

Figure	Page
3.1.a Structure of pass 16 x 16 for MBCPT.	29
3.1.b Parallel structure for MBCPT.	30
3.2 Example of 16 x 16 block quad tree.	31
3.3 Image reconstructed from pass 16 x 16.	32
3.4 Image reconstructed from passes up to 8 x 8.	32
3.5 Image reconstructed from passes up to 4 x 4.	33
3.6 Image reconstructed from passes up to 2 x 2.	33
3.7 Overhead assignment and zonal coding.	34
3.8 3 x 3 Gaussian spatial filter coefficients.	35
3.9 One-pixel overlap method.	36
5.1 Differential MBCPT coding scheme(1).	58
5.2 Differential MBCPT coding scheme(2).	59
5.3 Performances without packet loss.	60
5.4 Performances with packet losses from pass 4.	61
5.5 Error control coding.	62
5.6 PSNR versus video output rate.	63
5.7 Data output rate of simulation sequence frames.	64
5.8 PSNR of simulation sequence frames.	65
5.9 Frame 3 of simulation sequence.	68
5.10 Frame 4 of simulation sequence.	68
5.11 Difference image of frame 3 and 4.	69
5.12 Frame 9 of simulation sequence.	69
5.13 Frame 10 of simulation sequence.	70
5.14 Difference image of frame 9 and 10.	70
5.15 The effect of pass 4 packet loss for frame 4.	71
5.16 The effect of pass 4 packet loss for frame 10.	71

5.17	The effect of pass 1 packet loss for frame 3.	72
5.18	The effect of pass 1 packet loss for frame 9.	72

List of Tables

Table	Page	
5.1	Output bits rate for each and total pass.	66
5.2	Video output rate for each and total pass.	67

Chapter 1 Introduction

Communicating images have traditionally been the specialty of postal services, and of late, the air freight industry. Early electronic means of communicating photographs used wire services that were essentially forerunners of facsimile. Fax is now moving toward plain paper and eventually color. Fax networks are also becoming increasingly popular for timely distribution of routine communications. But fax is only the tip of an imaging communications iceberg. On the cusp of an explosion in integrated imaging, one effect will be a quantum leap in the demands on networks to move all these images. As color graphics and video become more prevalent, networking capabilities will have to increase further[1].

Due to the rapidly evolving field of image processing and networking, video information is promising to be an important part of tomorrow's telecommunication system. Up to now, telecommunication traffic has been mainly transported over circuit-switched networks. Since packet-switched networks are likely to dominate the communications world in the near future, it is necessary to develop techniques for video transmission over such networks.

The classic approach in circuit switching is to provide a "dedicated path", thus reserving a continuous bandwidth

capacity in advance. Any unused bandwidth capacity on the allocated circuit with circuit-switching is therefore wasted. Rapidly varying frequency signals, like video signals, require too much bandwidth to be accommodated by a standard circuit-switching channel. With a certain amount of capacity assigned to a given source, if the output rate of that source is larger than the channel capacity, quality will be degraded. If the generating rate is less than the limit, the excess channel capacity is wasted. Another point that strongly favors packet-switched networks is the possibility that the integration of services in a network will be facilitated if all of the signals are separated into packets with the same format.

Some coding schemes which support the packet video idea have been exploited. Verbiest and Pinnoo proposed a DPCM-based system which is comprised an intrafield / interframe predictor, a nonlinear quantizer, and a variable length coder[2]. Their codec obtains stable picture quality by switching between three different coding modes: intrafield DPCM, interframe DPCM, and no replenishment. Ghanbari has simulated a two-layer conditional replenishment codec with a first layer based on a hybrid DCT-DPCM and a second layer using DPCM[3]. This scheme generates two type of packets: "guaranteed packets" contains vital information and "enhancement packets" contains "add-on" information. Darragh and Baker presented a sub-band codec which attains user-prescribed fidelity by

allowing the encoder's compression rate to vary[4]. The codec's design is based on an algorithm that allocates distortion among the sub-bands to minimize channel entropy. Kishino et al. describe a layered coding technique using discrete cosine transform coding, which is suitable for packet loss compensation[5]. Karlsson and Vetterli presented a sub-band coder using DPCM with a nonuniform quantizer followed by run-length coding for baseband and PCM with run-length coding for nonbaseband[6]. In this thesis, a different coding scheme called MBCPT is investigated. Unlike those methods mentioned above, MBCPT doesn't use decimation and interpolation filters to separate the signals into sub-bands. But it has the property of sub-band coding by using variable blocksize transform coding. In Chapter 2, some of the important characteristics and requirements about packet video are discussed. In Chapter 3, some details of image data compression, scalar quantization, vector quantization and transform coding are introduced and the coding scheme, called Mixture Block Coding with Progressive Transmission, is discussed. In Chapter 4, a network simulator used in this thesis is introduced. In Chapter 5, the simulation result is discussed. Finally, in Chapter 6 a review of this thesis is summarized.

Chapter 2 Packet Video

In this chapter, the background environment for packet video is presented and some characteristics and requirements of packet video are demonstrated.

2.1 Broadband Integrated Service Digital Network

The demand for various services, such as telemetry, terminal and computer connections, voice communications, and full-motion high-resolution video, and the wide range of bit rates and holding times they represent, provide an impetus for building a Broadband Integrated Service Digital Network(B-ISDN). B-ISDN is a projected worldwide public telecommunications network that will service a wide range of user needs. Furthermore, the continuing advances in the technology of optical fiber transmission and integrated circuit fabrication have been the driving forces to realize the B-ISDN.

The idea of B-ISDN is to build a complete end-to-end switched digital telecommunication network with broadband channels. Still to be precisely defined by CCITT(International Telegraph and Telephone Consultative Committee), with fiber transmission, H4 has an access rate of about 135 Mbps. A user gains access to the B-ISDN by means of a local interface to a "digital pipe" of a certain bit rate. At any given point in time, the pipe to the

user's premises has a fixed capacity, but the traffic on the pipe may be a variable mix up to the capacity limit. Thus a user may access circuit-switched and packet-switched services, as well as other services, in a dynamic mix of signal types and bit rates.

The principal benefits to the user can be expressed in terms of cost savings and flexibility. The integrated services means that the user does not have to buy multiple services to meet multiple needs. Further, the user needs to bear the expense of just a single access line to these multiple services.

The B-ISDN can offer a variety of services, including existing voice and data transmission as well as:

- * Facsimile: services for the transmission and reproduction of graphics, handwritten, and printed material.
- * Teletext: service that enables the subscriber terminals to exchange correspondence.
- * Video: video conferencing, picturephone, DTV, HDTV.

2.2 Video Transmission over Packet-Switched Networks

Packet-switched networks have the unique characteristics of dynamic bandwidth allocation for transmission and switching resources and the elimination of

channel structure[7]. It acquires and releases bandwidth as it is needed. Because a video signals vary greatly in bandwidth requirement, it is attractive to utilize a packet-switched network for video coded signals. Allowing the transmission rate to vary, video coding, based on packet transmission, permits the possibility keeping the picture quality constant, implementing "bandwidth on demand". Summarizing the above, there are three main merits when transmitting video packets over a packet switching network:

- (1) Improved and consistent image quality: if video signals are transmitted over fixed-rate circuits, there is a need to keep the coded bit rate constant, resulting in image degradation when accompanying rapid motion.
- (2) Multimedia integration: as mentioned in section 2.1, integrated broadband services can be provided using unified protocols.
- (3) Improved transmission efficiency: using variable bit-rate coding and channel sharing among multiple video sources, scenes can be transmitted without distortion if other sources, at the same time, are without rapid motion.

But it has the following drawbacks:

- (1) The time taken to transmit a packet of data may change from time to time.

- (2) Packets of data may arrive very late or even get lost.
- (3) Headers of packets may be changed because of errors and delivered to the wrong receiver.

It has to be emphasized that the delay effect can reach very high levels if there are a lot of users accessing the network. Under many conditions, the loss of packet or erroneous receipt of other packets may seriously damage the quality of the image. Otherwise, because of the strong interaction between the coding algorithm and the network on which it is applied, a new video coding approach is required.

2.3 Interaction between Signal Processing and Networking

Video transmission over a packet-switched network, or "packet video" for short, poses a general problem: a signal with high and greatly varying rate has to be transmitted in a constrained period.

When the signals transmitted in the network are nonstationary and circuit-switching is applied, a buffer between the coder and the channel is needed to smooth out the varying rate. If the amount of data in the buffer exceeds a certain threshold, the encoder is instructed to switch into a coding mode that has a lower rate but worse quality to avoid buffer overflow.

In packet-switched networks, Asynchronous Time Division Multiplexing (ATDM) can efficiently absorb temporal variations of the bit-rate of individual sources by smoothing out the aggregate of several independent streams in common network buffers.

It is a difficult resource allocation and control problem to deliver packets in a limited time and provide a real time service, especially when the source generates a high and greatly varying rate. In packet-switching networks, packet losses are inevitable but they yield a better utilization of channel capacity. The video coder will require different channel capacity over time but the network will provide a channel whose capacity changes depending on the traffic in the network.

There are some interactions between the coder and the network which we have to consider and which become a part of specifications when we design the coder:

- (1) Adaptability of the coding scheme: The video source we are dealing with has a varying information rate. So it is expected that the encoder can generate different bit rates by removing the redundancy. When the video is still, there is no need to transmit anything.
- (2) Insensitivity to error: The coding scheme has to be robust to the packet loss so that the quality of the

image is never seriously damaged. Remember of that retransmission is impossible because of the tight timing requirement.

- (3) Resynchronization of the video: Because of the varying packet-generating rate and the lack of a common clock between the coder and the decoder, we have to find a way to reconstruct the received data synchronous to the display terminal.
- (4) Control of coding rate: Sensing the heavy traffic in the network, the coding scheme is required to adjust the coding rate by itself. In the case of a congested network, the coder is switched to another mode which generates fewer bits while degrading image quality.
- (5) Parallel architecture: The coder can be implemented in parallel. That means we can run the coding procedure at the lower rate in many parallel streams.

In the next chapter, we investigate a coding scheme to see if it satisfies the above requirements.

Chapter 3 Image Data Compression and Mixture Block Coding with Progressive Transmission

In this chapter, we introduce the basic concepts of image data compression. We also investigate a coding algorithm called Mixture Block Coding with Progressive Transmission (MBCPT).

3.1 Image Data Compression

Image data compression is a technique used to minimize the number of bits for representing an image. Typical television images have spatial resolution of approximately 512 x 512 pixels per frame. At 8 bits per pixel per color channel and 30 frames per second, this image raw data rate is about 1.8×10^8 bits/s. The large channel capacity and memory requirement for digital image transmission makes image data compression desirable.

There are two categories for image data compression, one is lossless coding which can recover the original image without any loss. The need for perfect recovery limits the compression rate that can be achieved. For larger compression rates, a second kind of coding scheme called lossy coding is applied. Lossy coding relies on many-to-one mappings to get a desired rate which is less than the source entropy.

There are two main ways to do lossy image data compression also. The first method, which is called predictive coding, exploits the redundancy in the data. Because an image is a highly correlated source, there is a lot of predictability, called redundancy, in the image. Techniques such as delta modulation and differential pulse code modulation fall into this group. The second method, called transform coding, transforms the given image into another array such that a large amount of the information is packed into a small number of samples. A more detailed discussion is provided in Section 3.3.

The entropy of an image source with L possible independent symbols with probabilities p_i , $i=0, \dots, L-1$, is defined as

$$H = - \sum p_i \log_2 p_i \quad \text{bits per symbol} \quad (1)$$

In the simulated image used in the thesis, L equals 256. According to Shannon's noiseless coding theorem, it is possible to code, in lossless coding, an image source of entropy of H bits per symbol using $H+\epsilon$ bits per symbol, where ϵ is an arbitrarily small positive quantity. In this case, the compression rate of lossless coding is defined by

$$\frac{\text{average bit rate of the original raw data (B)}}{\text{average bit rate of the encoded data (C)}} \quad (2)$$

In lossy coding, C can be much smaller.

3.2 Transform Coding and DCT

A variety of coding approaches have been developed. Some of the more promising involve segmenting the image into small subimages before coding. Specifically, the original image is divided into subimages, usually of equal size, and then each subimage is coded independently of the others. To reproduce the full image, the separate subimage blocks are reassembled by the decoder. The purpose of segmenting the image is to exploit the image's local characteristics and to simplify hardware implementation of the coding algorithm. Transform coding is a prime example of a coding technique involving image segmentation[8].

As we said above, transform coding is another candidate besides predictive coding for use in data compression. In this section, the characteristics of transform coding are introduced and we investigate one important transform called the discrete cosine transform used in MBCPT.

3.2.1 Transform coding

Block coding, another name for transform coding, transforms a block of data into a set of transform coefficients and quantizes each coefficient independently. An image is divided into equal size blocks when applied in two dimensions, limited by processing and storage ability. For an $M \times N$ image, if an $m \times n$ transform is applied, the

image will be divided into MN/mn blocks. The main storage space for doing the transform is reduced by a factor of MN/mn . Meanwhile, the number of operations will be reduced by a factor of $\log_2(MN)/\log_2(mn)$. That comes from two dimension transform with $O(N\log_2N)$ operations via an N -point FFT.

The aim of the transformation is to convert statistically dependent picture elements (pixels) into a set of essentially independent transform coefficients, preferably packing most of the signal energy (or information) into a minimum number of coefficients[9]. Bit allocation is another problem when designing a transform coder. If a coefficient contains a lot of energy, the absolute value is large, and more bits will be assigned to it. On the other hand, a coefficient with little energy will be represented with fewer bits, even none. Considering bit allocation, there are two approaches. First, only the definite zone of transformed coefficients are transmitted, we call it zonal coding with the zone covering the largest variances of transformed samples. The second one is threshold coding. Those coefficients with amplitude greater than a predetermined threshold are coded. In MBCPT, zonal coding is used.

Asymptotically DPCM and Transform coding have the same performance. However, under practical constraints, transform coding is a much more powerful tool than

predictive coding. It can get the relatively higher compression rate and distribute the error coming from quantization or channel over the entire image. If predictive coding is used, the visual degradation because of error will appear locally.

3.2.2 Discrete Cosine Transform (DCT)

Most unitary transforms pack a large fraction of the average energy of the image into a relatively few components of the transform coefficients. Since the total energy is preserved, this means many of the transform coefficients will contain very little energy. In the viewpoint of energy compaction and decorrelation, the Karhunen Loeve transform is optimum. But the Karhunen Loeve transform depends on the statistics and the size of the image and, in general, the basis vectors are not known analytically. After the transform matrix has been computed, the operations for performing the transformation are quite large for images. The discrete cosine transform is a nice substitute in highly correlated image transformation because it has excellent energy compaction and fast implementations[10].

The discrete cosine transform consists of a set of basis vectors that are sampled cosine functions. The transform matrix $C = \{c(k,n)\}$ may be written

$$\begin{aligned}
c(k,n) &= \frac{1}{N^{\frac{1}{2}}} \quad , \quad k = 0, \quad 0 \leq n \leq N-1 \quad (3) \\
&= \frac{2}{N^{\frac{1}{2}}} \cdot \cos \frac{\pi(2n+1)k}{2N}, \quad 1 \leq k \leq N-1, \quad 0 \leq n \leq N-1
\end{aligned}$$

The two-dimensional DCT may be defined as

$$F(u,v) = C[f(x,y)]C' \quad (4)$$

and the inverse transform

$$f(x,y) = C'[F(u,v)]C \quad (5)$$

As mentioned above, DCT is a fast transform. By the fast algorithm developed by Chen et al. [11], an $N \times N$ image DCT needs only $2N^2 \log_2 N - 2N^2 + 8N$ real multiplications and $3N^2 \log_2(N/2) + 4N$ real additions. Because zonal coding is used in MBCPT and only some of the coefficients need to be calculated, the operations can be reduced further and the real time processor can be practically implemented.

3.3 Quantization

Quantization is the next step after sampling in image digitization. A quantizer maps a continuous variable u into a discrete variable u' , which is a value from a finite set $\{r_1, r_2, \dots, r_n\}$. For most image transform, the dc coefficient is positive because the gray level is usually nonnegative. The ac coefficients have a zero mean and a

distribution very much like the Laplacian model. In the following, the two specific quantizers used in this thesis are discussed.

3.3.1 Scaler Quantizer

A scaler quantizer is an one-dimension quantizer which maps intervals of a line into points. The average distortion for a scaler quantizer is

$$d = \frac{1}{n} \cdot \sum_{i=1}^n \frac{a_{i+1} - a_i}{2} p(x) \cdot d(x, y_i) dx \quad (6)$$

where n is the number of codebook elements and (a_i, a_{i+1}) is the i -th interval containing element y_i .

An optimal Laplacian quantizer is used in this thesis which is developed with MAX's optimization theory for minimum distortion.

3.3.2 Vector Quantizer

Vector Quantization (VQ) has been widely used in low-bit-rate compression. It is a generalization of scalar quantization, and is, therefore, one step closer to the optimum, as given by Shannon's rate distortion theory[7]. In the image coding area, VQ is a new but promising technique for video compression.

There are two steps involved in the type of Vector Quantization used in this study. First, a codebook is generated from a large set of training vectors which should be as large and as varied as possible in order to accurately predict future vectors. The size of the codebook determines the bit rate of the vector quantizer. Second, the codebook is downloaded to both the transmitter and receiver. When the vector comes in, the codebook is then searched for the codevector which is the closest match to it and an alphabet representing the codevector is transmitted. At the receiving end, it only needs to find the matched vector which is much easier than at the transmitter end.

During both the codebook generation and the coding phases of vector quantization, it is necessary to find a "best match" for each vector. This best match should be the codevector which most closely approximates the input vector, or in other words, yields the lowest distortion.

In this thesis, the LBG vector quantizer is used. This LBG algorithm is simple yet powerful, and it can be used for the generation of a codebook for any vector quantization application. The algorithm itself is an iterative one, refining the codebook until the distortion has reached an acceptable value.

The distortion is simply the square of the Euclidian distance between the two vectors. The overall distortion

measure is computed after all training vectors have been partitioned. If this distortion falls below the acceptable threshold, the iterative process stops, and the current codebook is saved. Otherwise, if the distortion is too high, each codevector is replaced by the centroid of all the training vectors assigned to it. Then the training sequence is re-partitioned and the process is repeated.

3.4 Mixture Block Coding with Progressive Transmission

Here we investigate the algorithm and property of MBCPT to see if it can properly fit into the packet-switching environment.

3.4.1 Progressive Coding

The technique that allows an initial image to be transmitted at a lower bit rate and to be refined with an additional bit rate is called progressive coding[12]. Consider, for example, an image with size $xyz = 256 \times 256 \times 8$ bits is transmitted. One way to send it is in the zxy order: transmit all the eight bits of the first pixel in the first row, then stepping along the row (x) for all the pixels in that row, advancing down to the following row (y) until all the pixels in that image are sent. This is probably the simplest and usual way to send an image. Another alternative is to go through the xyz order, where the most significant bit of every pixel is sent first, then

the second one and so on to the least significant bit. In this way, successive approximations converge to the target image with the first approximation carrying the "most" information and the following approximations enhancing it. The process is like focusing a lens, where the entire image is transformed from low-quality into high-quality[13].

In progressive coding, every pixel value or the information contained in it is possibly coded more than once and the total bit rate may increase due to different coding schemes and quality desired. Because only the gross features of an image are being coded and transmitted in the first pass, the processing time is greatly reduced for the first pass and a coarse version of the image can be displayed without significant delay. It is proved that it is very useful for perception to get a crude image in a short time, rather than waiting a long time to get a clear complete image[14].

With different stopping criterion, progressive coding is suitable for dynamic channel capacity allocation. If a predetermined distortion threshold is met, processing is stopped and no more refining action is continued. The threshold value can be adjusted according to the traffic condition in the channel. Successive approximations (or iterations) are sent through the channel in progressive coding and leads the receiver to the desired image. If these successive approximations are marked with decreasing priority, then a sudden decrease in channel performance may

only cause the received image to suffer from quality degradation rather than total loss of parts of the images[13].

3.4.2 Structure of MBCPT Coder

Mixture Block Coding (MBC) is a variable-blocksize transform coding algorithm which codes the image with different block sizes depending upon the complexity of that block area. Low-Complexity areas are coded with large blocksize transform coder while high-complexity regions are coded with a small blocksize one. The complexity of the specific block is determined by the distortion between the coded and original image. A more complex image block has higher distortion.

The advantage of using MBC is that it does not process different complex regions with the same blocksize. That means MBC has the ability to choose a finer or coarser coding scheme to deal with different complex parts of the same image. With the same coding source (coding rate), MBC is able to increase the quality of the whole image than a coding scheme which codes a different complex regions with the same blocksize coder.

When using MBC, the image is divided into maximum blocksize blocks. After coding, the distortion between the reconstructed and original block is calculated. The

processing block is subdivided into smaller blocksize blocks if that distortion fails to meet the predetermined threshold. The coding-testing procedure continues until the distortion is small enough or the smallest blocksize is reached. In this scheme, every block is coded until the reconstructed image is satisfactory, then the next block is coded.

Mixture Block Coding with progressive transmission (MBCPT) is a coding scheme which combines MBC and progressive coding. MBCPT is a multipass scheme in which each pass deals with different blocksizes. The first pass codes the image with a maximum blocksize and transmits it immediately. Only those blocks which fail to meet the distortion threshold go to the second pass. This processes the difference image block from the original and coded image obtained in the first pass, with smaller blocksize blocks. The difference image coding scheme continues until the final pass which deals with the minimum blocksize block. At the receiving end, a crude image is obtained from the first pass in a short time and the data from following passes serve to enhance it. Fig. 3.1.a shows the structure of pass 16x16 for MBCPT. Fig. 3.1.b shows the parallel structure of MBCPT. A coding structure like a quad tree is proposed by Dreizen[15], and Vaisey and Gersho[16] which subdivides those busy blocks into four pieces and will be used in this thesis. In the quad tree coding structure of this thesis, the 16x16 block is coded and the distortion of the block is calculated. If the distortion is greater than

the predetermined threshold for 16x16 blocks, the block is divided into four 8x8 blocks for additional coding. This coding-checking procedure is continued until the only image blocks not meeting the threshold are those of size 2x2. Figure 3.2 shows the algorithm.

3.4.3 Design Consideration

There are several features which have to be considered when designing a MBCPT coder.

They are:

- (1) the blocksize of the transform coder.
- (2) the bits allocation.
- (3) the quantizer.
- (4) the distortion measurement.
- (5) the threshold value.

Considering the block size, it should be small enough for ease of processing and storage requirements, but large enough to limit the inter-block redundancy[17]. Larger block size results in higher image quality, but it is very difficult to build real-time hardware for block sizes larger than 16x16 because the number of calculations increase exponentially with block size for the DCT transform[13]. Besides, if the maximum blocksize is set too large, it is destined to be subdivided and decreases the efficiency of the coder. So, 16x16 is chosen to be the

largest blocksize here.

The minimum blocksize determines the finest visual quality that is achievable in the busy area. If the minimum blocksize is too large, it is likely that the blockiness will be observed in the coded edge of spherical object because the coding block is square. In order to match the zonal transform coding used in this thesis, 2×2 is the smallest blocksize and there are four passes (16×16 , 8×8 , 4×4 , 2×2) in this scheme. Fig. 3.3-6 show the images from 4 passes individually.

The monochrome images used in this thesis are represented with 8 bits of non-negative intensity ranging from 0 to 255. After a discrete cosine transformation, only four coefficients including the dc and three lowest order frequency coefficients are coded and the others are set to zero. The dc coefficient in the first pass is coded with an 8-bit uniform quantizer due to the fact that it closely reflects the average gray level for that image block and is hard to predict. It is easy to predict the dc coefficient in the following pass because it is a residual and has a distribution like a Laplacian model. Typically, a 5-bit optimal laplacian nonuniform quantizer is used. The three ac coefficients, as mentioned above, distribute like a Laplacian model with a variance greater than that of the dc coefficient. Because different variances are exhibited for different coefficients, the input samples are first normalized so that they have unit variance and therefore

can be quantized with the same 5-bit Laplacian quantizer. As an alternative, a LBG vector quantizer with a 512 codebook size is used to quantize the vector which comprises the three ac coefficients. Along with the blocksize determined above, the maximum and minimum bit rates for this coder ranges from 0.09 to 6.65 bits/pel for the scalar quantizer and 0.07 to 4.66 for the vector quantizer depending upon the complexity of the image.

Any distortion measure can be used in this MBCPT coder. It is possible to use different distortion measures for each different blocksize pass to adjust for the expected radial frequency coding sensitivity of the eye. Each different blocksize represents a different spatial frequency range that is to be coded, and details of distortion induced within each of these blocksizes will be seen differently by the eye. In this thesis, the maximum absolute difference is used:

$$d = \max_i |x_i - y_i| \quad (7)$$

where the range of i is taken over the entire block to be coded, u is the original image pixel while v is the coded image pixel. Because the visual performance mentioned above, a luminance to contrast model called logarithmic law as follows:

$$C = 50 \cdot \log_{10} f \quad , \quad 1 \leq f \leq 100 \quad (8)$$

is used to modify the maximum absolute difference law.

The threshold of each pass has to be selected before the coder is going to work. It is readjustable during the operation. If zero is assigned as the threshold for each pass, no block is going to satisfy that threshold and the maximum data rate is transmitted hoping for a perfect coded image. When using an infinite threshold, only the first pass data will be sent using the minimum bit rate. Any non-negative threshold will fall between these two extreme cases and can be adjusted according to the channel condition and quality required.

Because only partial blocks which fail to meet the distortion threshold need to be coded, there must be some side information to instruct the receiver how to reconstruct the original image back. One bit of overhead is needed for each block. If a block is to be divided, a 1 is assigned to be its overhead; if not, a 0 is assigned. A coding process in Fig. 3.7 has the following overhead: 1,1001,1001,1001,1001,1001.

3.4.4 Distortion and Blocking Effect

When using MBCPT, there are some types of error that can appear in a decoded image. First, high-frequency errors, result from eliminating DCT coefficients using zonal masking and a large thresholds. High-frequency error

is characterized by a general blurring of sharp edges in the reconstructed image. Another type, quantization error, occurs when DCT coefficients are assigned too few bits from the bit assignment map. Quantization error is characterized by sinusoidal rippling of intensity in the originally solid areas; edges remain fairly sharp, but are distorted[8].

In MBCPT, the input image is partitioned into a series of nonoverlapping rectangular blocks or subimages with equal size. Each subimage is a partial scene of the original image and is processed independently. In low bit-rate application, like the first pass, the block boundaries become highly visible and objectionable. Two approaches are used in this thesis to eliminate the blocking effect. First, because the location of these block edges are known exactly in MBCPT, it is reasonable to expect that low-pass filtering the image at or near the subimage boundaries could smooth the unwanted discontinuities. This is the basis of the filtering method[18]. A 3 x 3 Gaussian spatial domain filter (Fig. 3.8) is used. Second, instead of forcing the regions to be exclusive of each other, it is reasonable that a slight overlap around the perimeter of each region could reduce the blocking effect, this is called the overlap method (Fig. 3.9)[18]. The pixels at the perimeter would then be coded in two or more regions. In reconstructing the image, a pixel that was coded more than once would use an average of the coded values.

Both methods are successful in reducing the blocking effect. But the overlap method results in a 13% increase in bit rate while the filtering method, due to its low-pass nature, may degrade edge content in the image.

3.4.5 Application in Packet-Switching Network

Because of the dynamic and adaptive characteristic in MBCPT, we can see some interesting features when applied to packet video:

- (1) The minimal quality is ensured. (i.e. that of the basic channel with higher priority)
- (2) Packet losses on the improvement channel do not impair the received signal below the quality offered by the basic channel.
- (3) Bandwidth on demand can be easily implemented.
- (4) The scheme is very simple since all complexity is in the basic channel codec which operates at low frequency.
- (5) An evolutionary transition from today's synchronous networks to tomorrow's asynchronous networks becomes possible, since the basic channel is implemented now and the improvement channel added in the future on the fast packet network[19].

In this chapter, the structure and basic features about MBCPT was investigated. From that, it can be seen why this

algorithm is able to fit into packet-switching environment.
A lot of details about that will be discussed in Chapter 5.

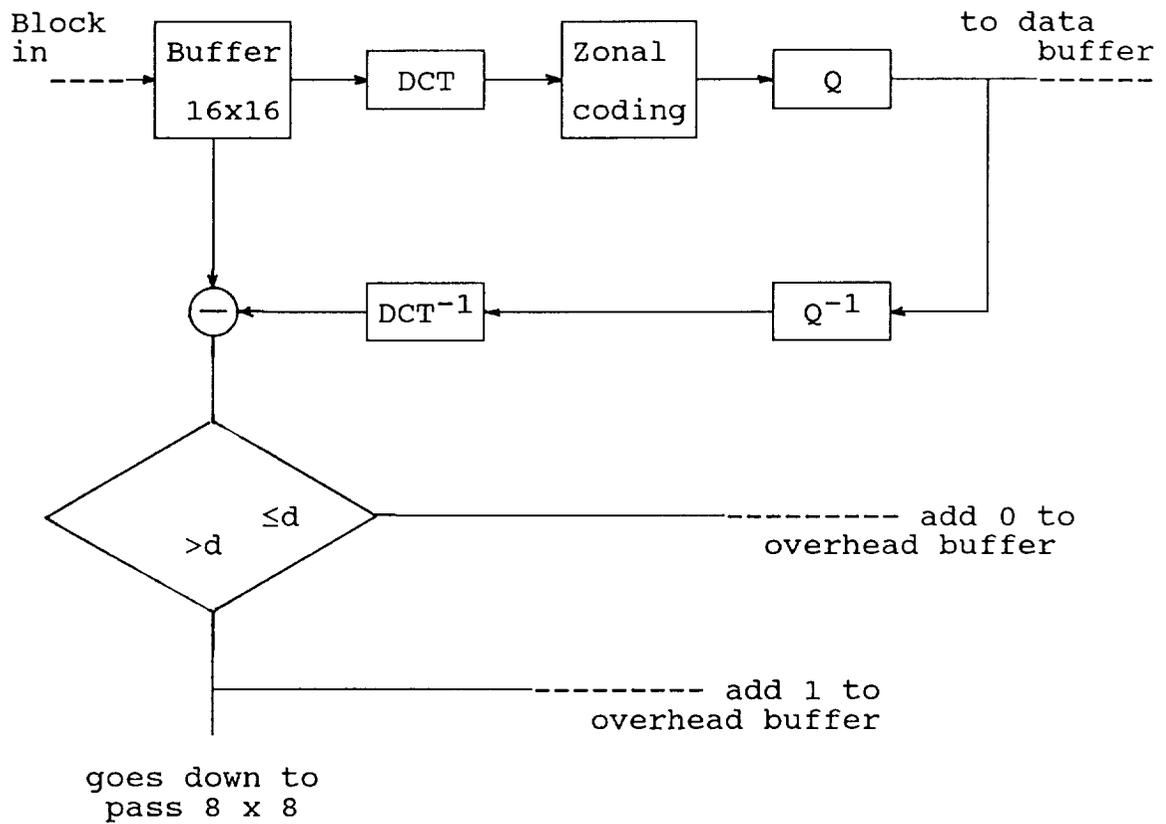
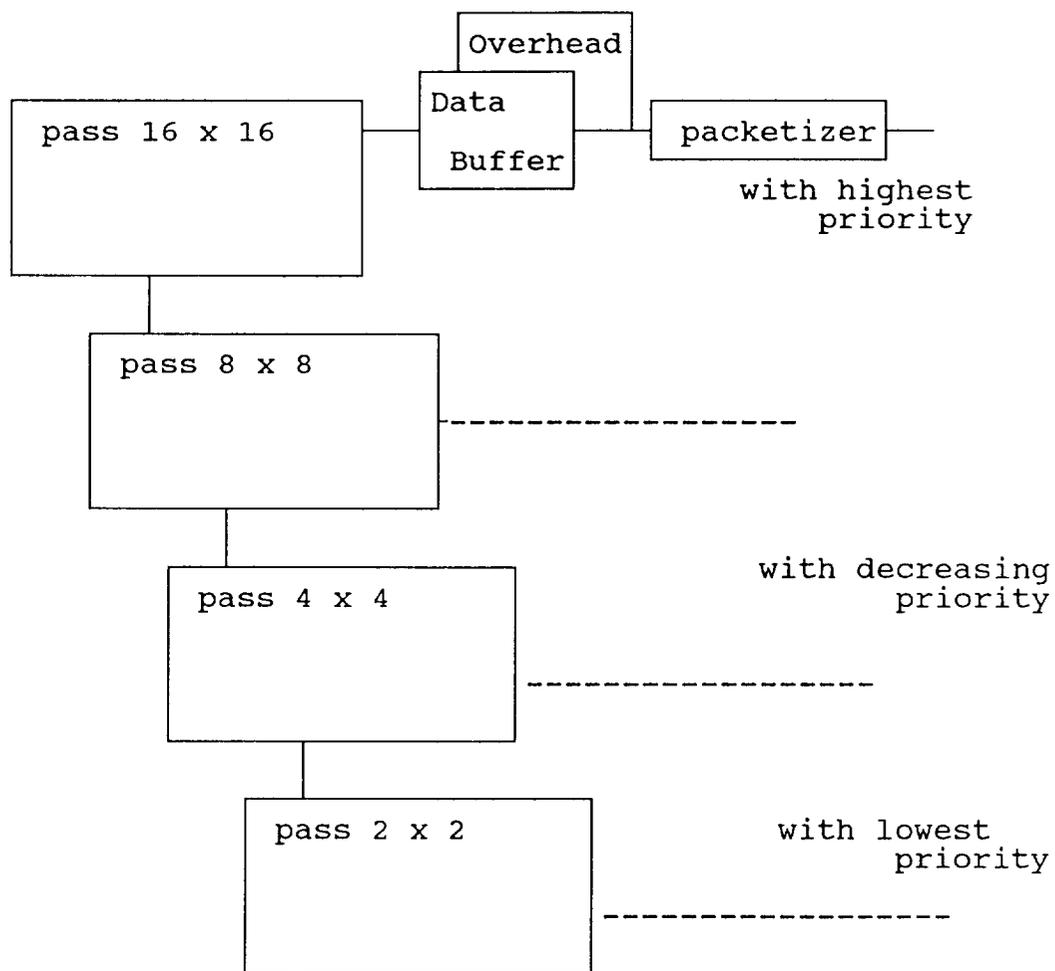


Figure 3.1.a Structure of pass 16 x 16 for MBCPT. d is distortion defined in Eqs.(7).



Figuer 3.1.b Parallel structure for MBCPT

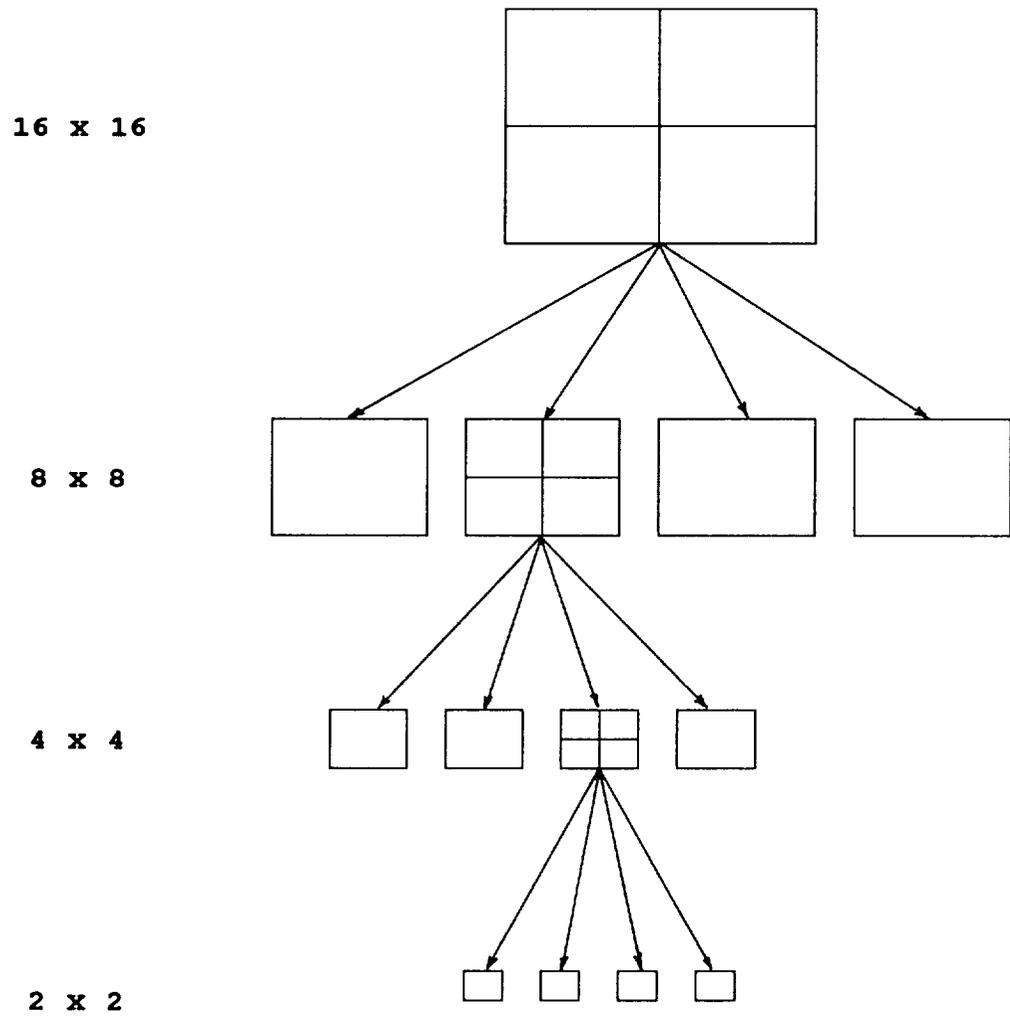


Figure 3.2 Example of 16x16 block quad tree

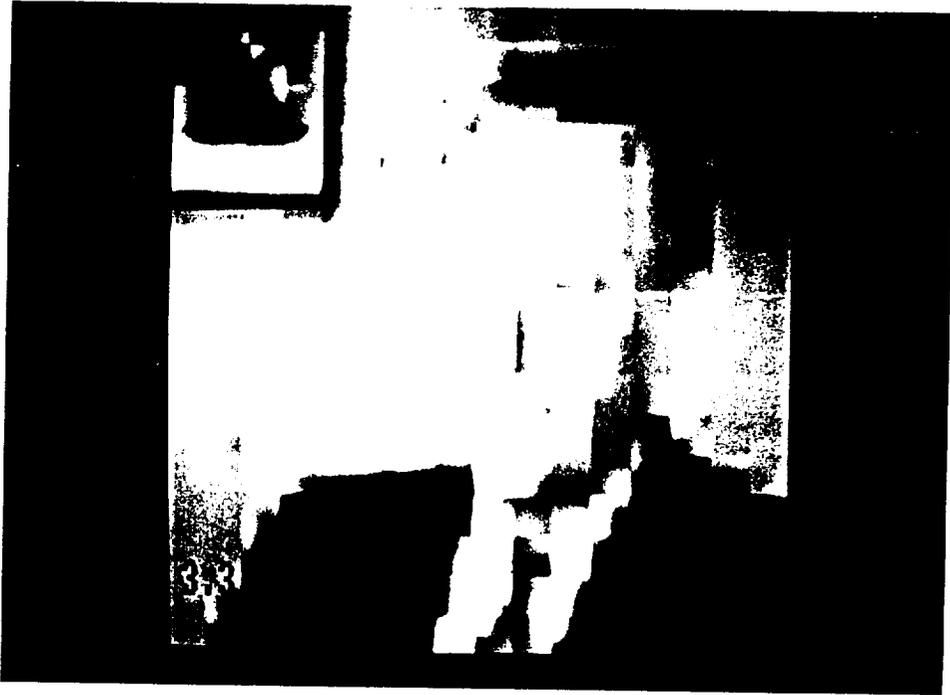


Figure 3.3 Image reconstructed from pass 16x16.



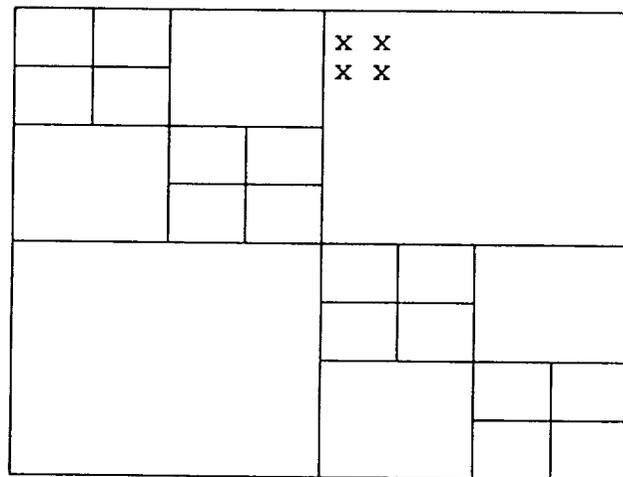
Figure 3.4 Image reconstructed from passes up to 8x8.



Figure 3.5 Image reconstructed from pass up to 4x4.



Figure 3.6 Image reconstructed from passes up to 2x2.



16 x 16

overhead = 1,1001,1001,1001,1001,1001

Figure 3.7 Overhead assignment and zonal coding

.0751	.1239	.0751
.1239	.2042	.1239
.0751	.1239	.0751

Figure 3.8 3x3 Gaussian spatial filter coefficients

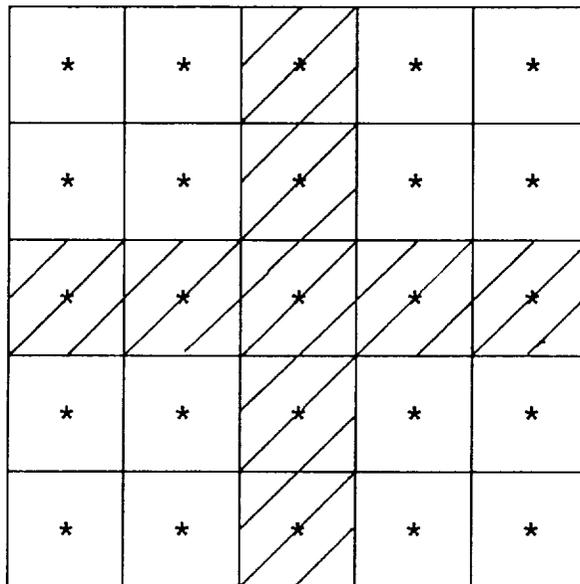


Figure 3.9 One-pixel overlap method. Shading indicates over-lap. Subimage size 3x3 pixels.

Chapter 4 Network Simulator

The network simulator to be used for this thesis is a modification of an existing simulator developed by Nelson et al.[20]. A brief description of the simulator is provided here.

4.1 Introduction

As mentioned in Chapter 2, tomorrow's integrated telecommunication network has a very complicated and dynamic structure. Its efficiency requires sophisticated monitoring and control algorithms with communication between nodes reflecting the existing capacity and reliability of system components. The scheme for communicating information regarding the operating status is called the system protocols.

Since this communication of system information must flow through the channel, it reduces the overall capacity of the physical layers, but hopefully provides a more efficient system overall. Therefore, the optimal system efficiency depends a lot upon these protocols, in turn, upon the system topology, communication channel properties, nodal memory and component reliability. Most network protocols have been developed around high reliability in topological structures with reasonable high channel reliability.

In order to fit into the purpose of this thesis, most modifications which have been made to this simulator are basically in those modules concerning network layers. And this simulator is structured in modules which represent, to some degree, the ISO Model for packet switched networks. Therefore, a more detailed description about the network layer modules will be made in the next section. In this chapter, a whole picture for this simulator will be provided.

4.1.1 Topology, Traffic and Preparation

The program Topology is used to generate a topological description of the network to be simulated. It contains the number of nodes, the definition(includes connectivity and propagation delay) of the links between nodes, and the initial bit error rate for each link.

The program Traffic is used to generate an initial statistical description of the network traffic to be simulated. It contains the average message length for each precedence, the percent of messages generated for each precedence level, the rate of message generation at each node and the distribution of those messages to the other nodes.

The program Simprep is used to generate a checkpoint

file which contains all the data needed for the simulation including the topology, traffic and network parameters for various network layers.

4.1.2 Simulator Philosophy

The principle function of the simulator is to perform tasks at the appropriate time. A queue called SIM_Q drives the simulator. The records in SIM_Q contain:

1. The task to be performed.
2. The time at which the task will be completed.
3. Node 1 (sender).
4. Node 2 (receiver).
5. Line (channel line routed).
6. The message number and the packet number.
7. A pointer to a packet (if one is involved).
8. Queue pointers for a doubly linked list.

The main simulator program has the popping of SIM_Q and the execution of routines which effect the completion of the scheduled task contained in the popped record. These completion routines simulate the completion of the task and may result in other completion tasks to be performed in the same layer or other layers. A new task will be queued in the appropriate queue. If it is for another layer, then if the processor for that layer is idle it will invoke the scheduler for that layer.

4.1.3 Simulator Queue and Queue Processors

Central to the operations of the simulator are the various queues. There are two types of records which are entered into queues. One is the Sim_Q_Record which contains the information required to perform a task and the other is the Packet_Record that contains the information regarding the contents and status of a packet. The main program SIMEX works directly from SIM_Q, which is the queue of Sim_Q_Records. There is only one such queue for the entire simulator but there exists many packet queues. There are three kinds of packet queues which are referred as Memory_Q, Packet_Q and Cleanup_Q in the simulator.

The Message_Q contains all packets originating at this node and the Transfer_Q contains all packets received from other nodes fall into the group of Memory_Q. The Packet_Q is used to simulate the nodal queues in which the packets reside as they progress through the various network layers. These queues are mutually exclusive in that a packet can only reside in one of these queues at any given time. The transport_Q is for those packets waiting for packetization or reassembly in the transport layer. In the case of a packet waiting for routing, it is placed in the Input_Q in the network layer. If a packet is heading for other nodes, it is placed in the Output_Q waiting for transmission by datalink layer.

If piggy-back acknowledgement is allowed, then it is possible that a packet's address from the sending node must be stored for a period of time before the opportunity exists to return the address in an acknowledgement. In the simulator, this is accomplished through the Cleanup_Q.

4.2 The Network Layers

Each layer of the simulator module contains a processor and one or more packet queues. The processor is idle before there is a packet coming into its associated queue. The packet and the task that must be performed are entered into SIM_Q with a completion time. When the task is performed, that means the completion time has arrived, then the queue is checked. If there is another task to be performed, then its completion is scheduled. If the queue is empty the processor is marked idle again.

The layers in the simulator are quite close in operation to the ISO transport, network and datalink layers. A "partial" session layer exists principally as a reporting layer for end to end statistics.

4.2.1 The Session Layer

In the OSI model, the session layer allows users on different machines to establish "sessions" between them. In the simulator, as mentioned above, it is a relatively simple model of the subscribers and an end to end

statistics collector. At message arrival time, the session layer generates the message with all of its randomly selected attributes and if flow control or node hold-down are not in effect, submits it to the transport layer and then builds up the next message arrival time.

During initialization, a task "SL_Rcv_Msg" for each node is queued in SIM_Q for the arrival time of the first message at that node. When this task is executed by the simulator, a message packet is generated and placed in the transport queue. The arrival of the next message is then queued in SIM_Q with the same task and an arrival time determined by the random number generator (Poisson distribution).

The only other task performed at the session layer is the SL_Snd_Msg task which simulates the delivery to the subscriber. In the simulator, this is principally a "bookkeeping task" that records message statistics and "cleans up" the queues containing packets with resolved references.

4.2.2 The Transport Layer

The basic function of the transport layer is to receive the message from the session layer, separate it up to smaller units if necessary, pass these to the network layer and make sure these pieces will arrive sequentially

at the other end. Furthermore, all this work is expected to be done efficiently, and in a way that isolates the session layer from future progress in hardware technology.

In the simulator, the transport layer simulates packetization, reassembly, message acknowledgement and resubmittal in the case that a message acknowledgement is not received in time, transport-layer time-out. There are four tasks simulated by the transport layer. They are TL(Transport Layer)_Packetize, TL_Timeout, TL_Reassemble, and TL_Ack_Send. It is recognized that in some networks, packetization takes place at the network level, leaving the transport layer responsible only for message level structures. Reassembly, depending upon the protocol can take place as low as the datalink level. These tasks were both placed in the transport layer for ease of coding, but are separate modules that could be quite easily extracted and placed elsewhere. Also, the system was originally designed for datagram operation and since the packets will not necessarily arrive in order, it is unlikely that assembly would take place at the datalink level.

4.2.3 The Network Layer

The network layer is concerned with controlling the operation of the network. A key design issue is determining how packets are routed from source to destination. Another issue is how to avoid the congestion caused if too many packets are presented into the network at the same time. In

the simulator, the network layer performs all of the functions related to these two aspects with the exception of flow control which takes place at the session layer, and the recovery protocols which require some service from the datalink layer. It also activates new channels when needed and determines when packets originating at other nodes are to be discarded.

The network layer is currently the most dynamic with regard to the coding of modules. Five modules currently comprise the network layer. These include relatively static modules; one module for dialing up new lines when more line capacity is required and releasing them when not needed; one module for the network processor and queue handling and one module for the routines which are common to most routing algorithms. This leaves two modules for the dynamic parts of the routing and flow control algorithms.

4.2.4 The Datalink Layer

The main task of the datalink layer is to take a raw transmission facility and transform it into a line that appears free of transmission errors to the network layer. It simulates the sending of the message over the channel and the delivery at the other end. When a packet is received, the datalink acknowledgement is initiated either by piggy-back acknowledgement or by generating a datalink acknowledgement packet.

As mentioned previously, the datalink level also simulates the physical layer on a statistical basis. If correct transmission was indicated (through a random number generator) then acknowledgement was also assumed. Current datalink layer simulation modules include generation of acknowledgement packets and simulation of the piggy-back acknowledgement as well. When a line is "brought up", health packets are used to establish initial connections. Also, when a line "goes down", an active node will immediately issue health check packets to ascertain when the channel is again available.

4.3 Modifications

A major problem of using this system as a simulation tool for the study of packet video is that the system doesn't actually transmit the data from node to node. While a packet is transmitted, the data field is empty. Therefore modifications had to be made to the simulator to accommodate the video data.

In the sending node, a field called "Image" which contains real image data is attached to the record "Packet_Ptr" allocated to the message generated in the session layer. There are three new modules in this layer. First, "Get_Image" puts the image data into the image field of a message generated at a specific time and node. Second, "Image_Available" checks if there is still any image data

needed to be transmitted. If that is true, the following message generated at that specific node is still the image message and contains some image data. Third, "Receive_Image" collects the image data in the session layer of the receiving node when the flag "Image_Complete" is on. In module "Session_Msg_Arrive", different priority is assigned to different messages. In module "Session_Msg_Send", some statistics are calculated including the number of lost image packets and the transmission delay for image packets.

Currently, the transport layer simply duplicates the same packet with different assigned sequential packet numbers without actually packetizing the message. The module "Transport_Packetize" is modified to really packetizing the image data which resides in the message record queued in Transport_Q when it is called. The module "Transport_Reassemble" is called to reassemble these image packets according to their packet number when the flag "Image_Content" defined in Packet_Ptr is true.

The network layer is responsible for routing and flow-control. This module is already very well developed, so the modifications to be performed here will be relatively minor.

In the datalink layer, in order to simulate the delivery of packets through the channel, a new packet will

be generated at the receiving node and the information including the image data from the transmitted packet (which will still be resident at the sending node) will be copied into it. With the bit-error-rate defined in the program Topology, transmission success rate will be set and bit errors can be inserted in both the data and control bits in the packet. Errors in the control bits are simulated separately as long as the error rates are consistent. If an error in the control bits occurs, the transmission fails and needs to be sent again depending on the threshold of the timeout number.

Besides the modifications made in those layer modules, we still have to arrange some new memory elements allocated for image messages and packets. In order to make sure the simulation is run in the steady state, image data is available after some simulation time.

In the next chapter, the interaction between this simulator and the coding scheme investigated in the last chapter will be presented.

Chapter 5 Simulation Results

In this chapter, an interframe coder based on MBCPT will be introduced and the simulation results will be discussed.

5.1 Differential Interframe Coding

Teleconferencing, picturephones, and broadcast videos are all transmitted as sequences of two dimensional images and are viewed as a three dimensional video source. An interframe coder is used to exploit the redundancy between the successive frames. The differences between frames basically come from object and camera motion.

The interframe coder used in this thesis is a differential scheme which is based on MBCPT. This coder processes the difference image coming from the current frame and the previous frame which is locally decoded from the first three passes. Fig. 5.1 shows the algorithm of this coder. Fig. 5.2 shows a different scheme which does the local decoding with all four passes. Compare the simulation results from these two approaches. When there is no packet get lost, the performances of these two schemes are quite the same (from Fig. 5.3). But when congestion happened in the network, with the priorities assigned to packets, packets from pass 4 are expected to be discarded first. In this case, the performance (from Fig. 5.4) of

scheme in Fig. 5.1 is much better than the one in Fig. 5.2. Therefore the coding scheme in Fig. 5.1 is used in our simulation. In this thesis, the Kronkite motion picture with 16 frames is used as the simulation source. Every image is 256x 256 pixels with graylevels ranging from 0 to 255. It is similar to a video conferencing type image which has neither rapid motion nor scene changes. Due to this characteristic, advanced techniques like motion detection or motion compensation are not used at here but could be implemented when dealing with broadcasting video.

From the datastream output that is listed in the Table 5.1, we can see that the data in pass 4 which represents 30-40% of the entire data and is deemed as the least significant pass(LSP). This part of the data is going to increase the sharpness of the image and is usually labeled with the lowest priority in the network. With a substantial possibility of being discarded due to low priority, those packets from pass 4 will not be used to reconstruct the locally decoded image and will not be stored in the frame memory. That is supposed to avoid the packet loss error propagating into the following frames if the lost packet truly belongs to pass 4. It is found through simulation that the peak signal-to-noise ratio (PSNR) is increased by 1-2 dB in this scheme over using all pass data to reconstruct the reference image when the packet loss really occurs.

5.2 Interaction of the Coder and the Network

When the video data is packed and sent into a nonideal network, some problems will emerge. These are discussed in the following section.

5.2.1 Packetization

The task of the packetizer is to assemble video information, coding mode information, if it exists, and synchronization information into transmission cells. In order to prevent the propagation of the error resulting from the packet loss, packets are made independent of each other and no data from the same block or the same frame is going to be separated into different packets. The segmentation process in the transport layer has no information regarding the video format. To avoid having the bit stream being cut randomly, the packetization process has to be integrated with the encoder which is in the presentation layer of user's premise. Otherwise, some overhead has to be added into the datastream to guide the transport layer in doing the correct packetization. In order to limit the delay of packetization, it is necessary to stuff the last cell of a packet video with dummy bits if the cell is not completely full.

Every packet must contain an absolute address which indicates the location of the first block it carries. Because every block in MBCPT has the same number of bits in

each pass, there is no need to indicate the relative address of the following blocks contained in the same packet. There always exists a tradeoff between packaging efficiency and error resilience. If error resilience is considerable, one packet should contain a smaller number of blocks. However, since each channel access by a station contains an amount of overhead, the packet should be long for transmission efficiency. Fixed length packetization is used in this thesis for simplicity.

5.2.2 Error Recovery

There is no way to guarantee that packets will not get lost after being sent into the network. Packet loss can be attributed to two main problems. First, bit errors can occur in the address field, leading the packets astray in the network. Second, congestion can exceed the networks management ability and packets are forced to be discarded due to buffer overflow. Effects created by higher pass packet (like pass 4) loss in MBCPT coding will be masked by the basic passes and replaced with zeros. The distortion is almost invisible when viewing at video rates because the lost area is scattered spatially and over time. However, low pass packet (like pass 1) loss, though rare due to high priority, will create erasure effect due to packetization and be very objectionable.

Considering the tight time constraint, retransmission

is not feasible in packet video. It may also result in more severe congestion. Thus, error recovery has to be performed by the decoder alone. In our differential MBCPT scheme, the packets from pass 4 are labeled lowest priority and form a great part of the complete data. These packets can be discarded whenever network congestion occurs. That will release the network congestion and will not cause too much quality degradation. The erasures caused by basic pass loss is simply covered with the reconstructed values from the corresponding area in the previous frame. This remedy appears insufficient even when there is only a small amount of motion in that area. Motion detection and motion compensation could be used to find a best matched area in the previous frame for replacement.

Side information in the MBCPT decoding scheme is very important. So, this vital information is not allowed to get lost. Two methods can be used for protection. First, error control coding, like block codes or convolutional codes, can be applied in both directions along with and perpendicular to the packetization. The former is for bit error in the data field while the latter is for packet loss. Fig. 5.5 demonstrates the second case. The minimum distance that the error control coding should provide depends on the network's probability of packet loss, correlation of such loss and channel bit error rate. Second, from Table 5.1, we can see that the output rate of side information and pass 1 and even pass 2 is quite steady. It seems feasible to allocate an amount of channel

capacity to these outputs to ensure their timely arrival. That means circuit switching can be used for important and steady data.

5.2.3 Flow Control

In order to shield the viewer from severe network congestion, there are some flow control schemes which are considered useful. If there is an interaction between the encoder and the transport layer, then the encoder can be informed about the network condition. Depending on that, the encoder can adjust its coding scheme. In the MBCPT coding scheme, if the buffer is getting full, it means that the bit generating rate is overwhelming the packetization rate and the encoder will switch to a coarse quantizer with fewer steps or loosen the threshold to decrease its output rate. In this way, smooth quality degradation is obtainable. This will also complicate the encoder design.

It is possible to use the congestion control of the network protocols to prevent the drastic quality change by assigning different priorities to packets from different passes. Without identifying the importance of each packet and discarding packets blindly sometimes brings disaster and cause session shut down, for example if the side information gets lost. In the MBCPT coding scheme, side information and packets from pass 1 are assigned with highest priority and higher pass packets are assigned with

decreasing priority.

5.2.4 Resynchronization

Because of the lack of a common clock between transmitter and receiver and the variable packet generating rate used in packet video, resynchronization is an inherent problem in packet transmission. Transmission delay is irrelevant for one-way sessions and resynchronization can be solved by buffering the received packets in the receiver for a duration equal to L units from the start of transmission before transferring to the decoder. That means there is a constant lag of L units between the encoder and decoder. A packet loss occurs when any packet can not arrive in the limited time.

Although transmission delay is tolerable in one-way transmission, it becomes critical in two-way sessions because long delays impede information exchange. There are three methods which can be employed to accomplish the resynchronization task. The first approach is to modify the phase between the sending and receiving clocks by skipping or repeating video frames. The second scheme is to approach the transmitting frequency by the time stamps carried in the packet. Noted that this scheme can not be adopted by a multidrop decoder because it receives signals from more than one source. The third method is to adjust the receiving clock with a phase-locked loop by observing the level of the input buffer at the receiving end.

5.2.5 Interaction with protocols

In the ISO model, physical, datalink and network layers comprise the lower layers which form a network node. The higher layers have transport, session, presentation and application layers and typically reside in the customer's premises.

The lower layers have nothing to do with signal processing and only work as a "packet pipe". The physical layer requires adequate capacity and low bit-error-rate which are determined only by the technology. The datalink layer can only deal with link-management because all the mechanics like requesting retransmission is not feasible in packet video transmission. The network layer has to maintain orderly transmission by deleting the delay jitter with input buffering. Otherwise, it can take care the network congestion by assigning transmission priority.

As the higher layers reside in the customer's premises, they perform all the functions of the packet video coder. The transport layer does the packetization and reassembly. The packet length can be fixed or variable. Fixed packet length simplifies segmentation and packet handling while a variable packet length can keep the packetization delay constant. The session layer supervise set-up and tear-down for sessions which have different

types and qualities. There is always a tradeoff between quality and cost. The quality of a set-up session can be determined by the threshold in the coding scheme and the priority assignment for transmission. Of course, the better the quality, the higher the cost. Fig. 5.6 shows the tradeoff between PSNR and video output rate by adjusting thresholds. The presentation layer does most of the signal processing, including separation and compression. Because it knows the video format exactly, if any error concealment is required, it will be performed here. The application layer works as a boundary between the user and the network and deals with all the analog-digital signal conversion.

5.3 Results from Packet Video Simulation

Results obtained in this packet video simulation show that a pretty high compression and associated image quality can be obtained using this differential MBCPT scheme.

The monochrome sequence used in this simulation contains 16 frames, each of size 256x256 pixels with 8 bits per pixel, corresponding to a bit rate of 15.3 Mbits/s, given a video rate of 30 frames/s. As Table 5.2 shows, the average data rates of our system is 1.539 Mbits/s. The compression rate is about 10 with a mean PSNR equals 38.74 dB as calculated from

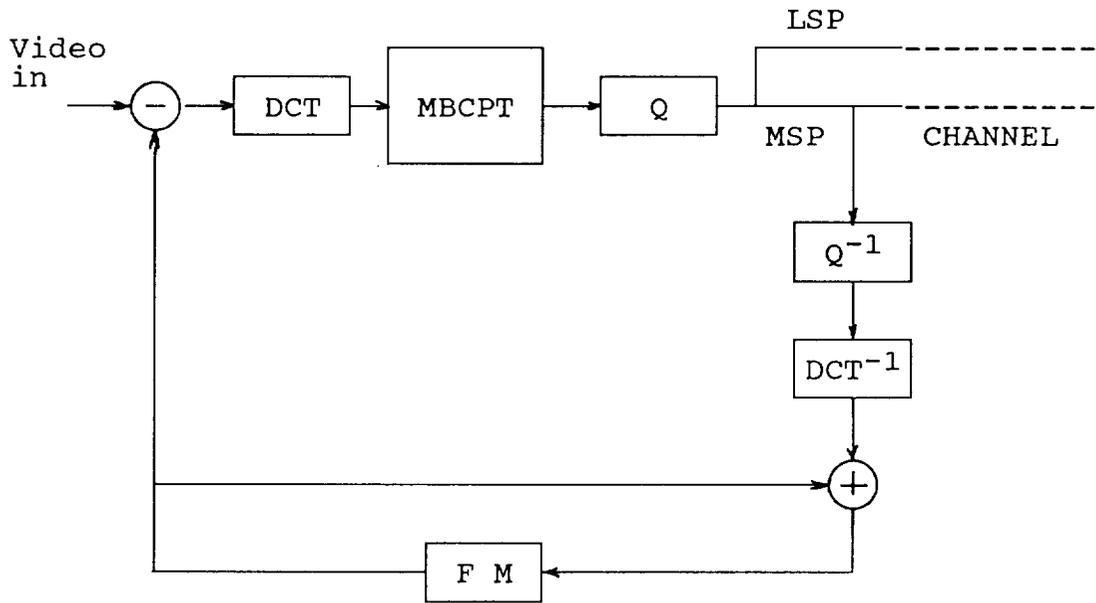
$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{256^2}{\sigma_{\text{diff}}^2} \right) \quad (9)$$

where 256 is the peak intensity of the image pixel and σ^2_{diff} is the variance of the difference between original and reconstructed frames. Fig. 5.7 shows the data rate of the sequence frames with side information, 4 passes and the total rate. It is clear that the data rate of pass 1 is constant as long as the quantization mode remains the same.

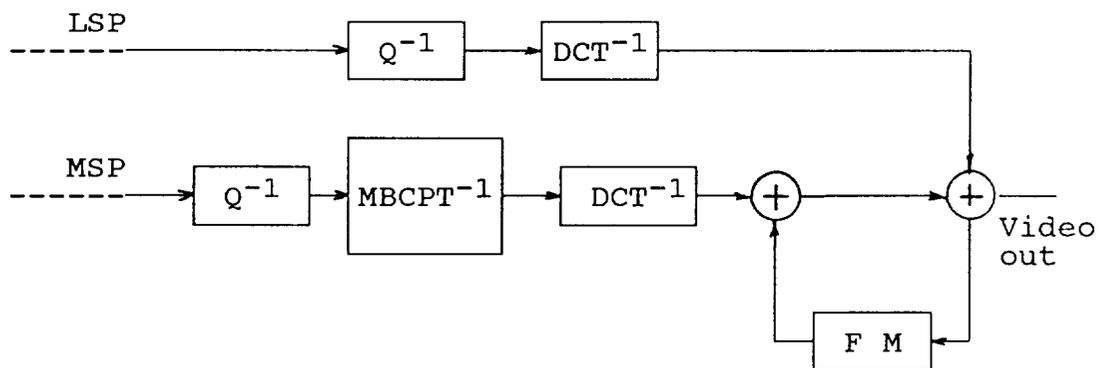
Side information and data from pass 2, and even pass 3, is quite steady and is referred as Most Significant Pass (MSP). The data rate of pass 4 is bursty, highly-uncorrelated and is called Least Significant Pass (LSP). Fig. 5.8 shows the PSNR for each frame in the sequence. The standard deviation is only 0.2 dB. In the simulation, the same threshold is used throughout the sequence. If constant visual quality is desired, a varying threshold can be used for different frames. That will generate a much more varying bit rate, of course motion detection is required. Comparing these two figures, it seems true that a varying bit rate can support constant quality video.

From the difference images of this sequence, frames 1-8 (Fig. 5.9-11) seem quite motionless while frames 9-13 (Fig. 5.12-14) are with substantial motion. We adjust the traffic condition of the network to force some of the packets to get lost in order to check the robustness of the coding scheme.

Transmission delay is not considered in this simulation because it is not the main interest. Heavy traffic is set up in the motionless and motion period separately. The average packet loss percentage is 3.3% which is considered high for most networks. Fig. 5.15-16 show the images which suffer the packet loss from pass 4. As can be seen, the effect of lost packets is not at all severe, even if the lost packet rate is unrealistically high. This is because the performance from the first three pass is relatively good. Fig. 5.17-18 show the case when packet loss occurs in pass 1. Clearly there are visible defects in the motion period. What's worse is that the error will propagate to the following frames. Apparently, the replenishing scheme used here is not sufficient in areas with motion. It is believed that this inconsistency can be eliminated with a motion compensator algorithm, which would find the appropriate area for replenishment, and with error concealment, which limits the propagation of error.



Coder



Decoder

Figure 5.1 Differential MBCPT coding scheme(1)

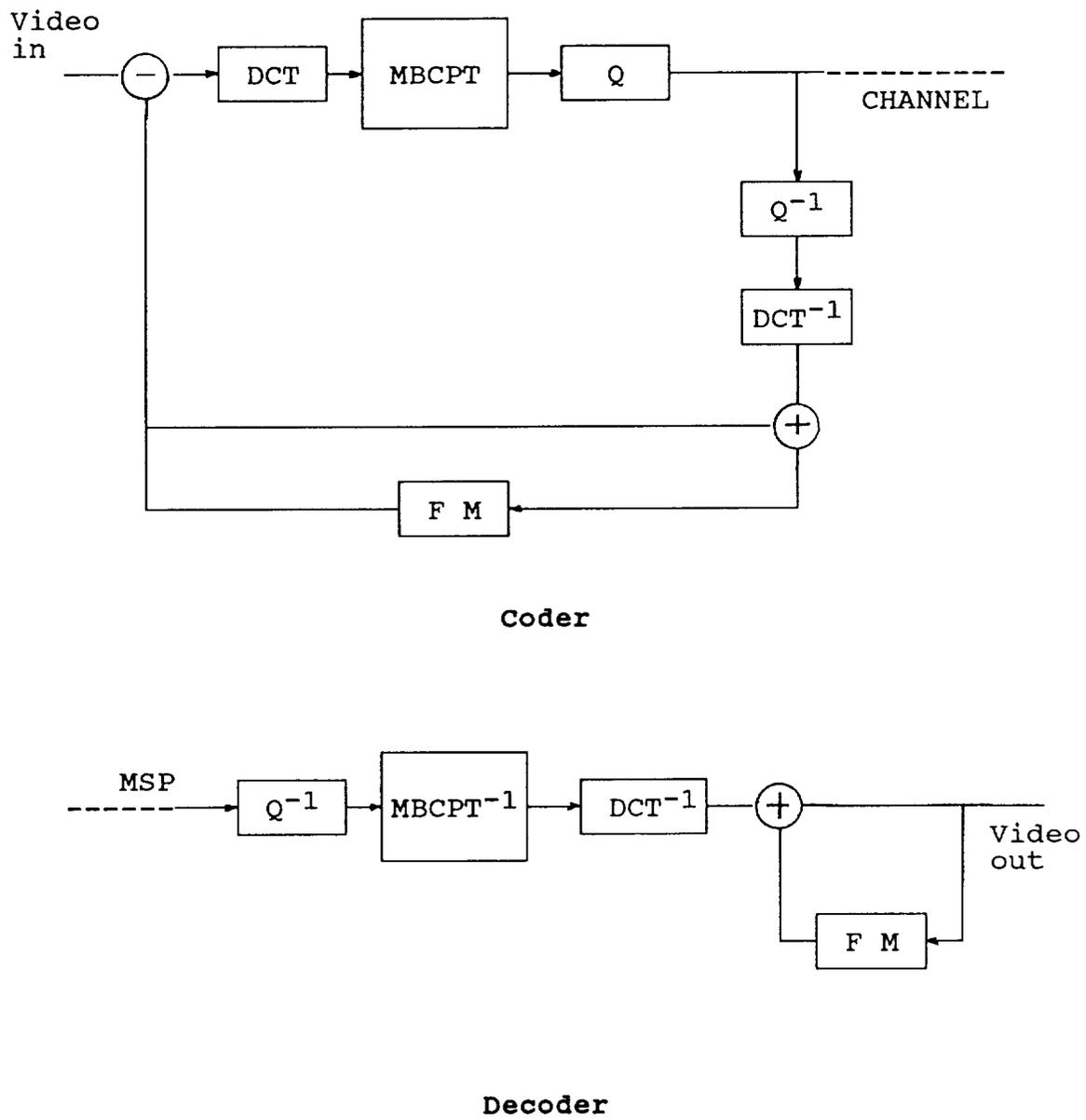


Figure 5.1 Differential MBCPT coding scheme(2)

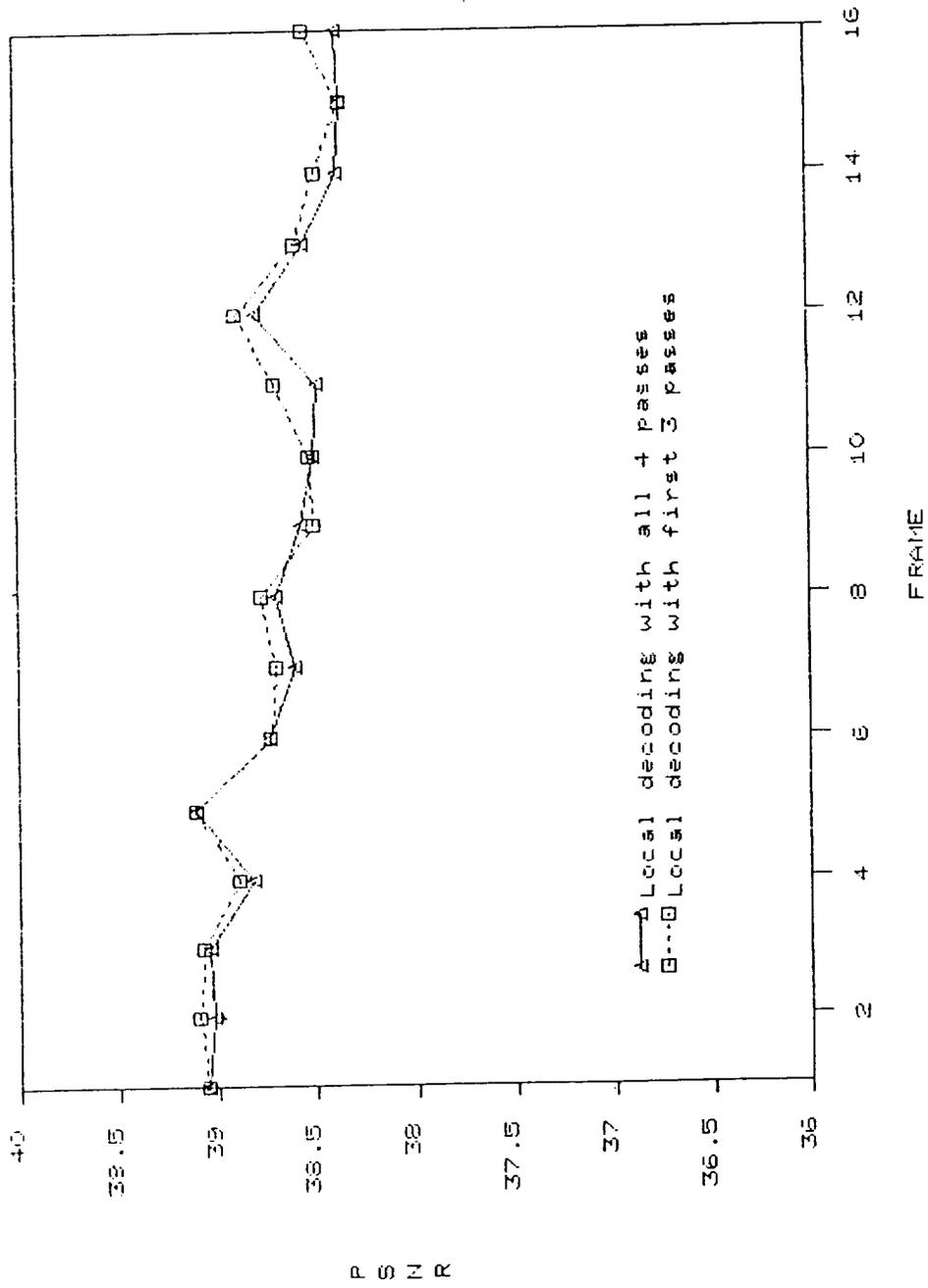


Figure 5.3 Performances without packet loss.

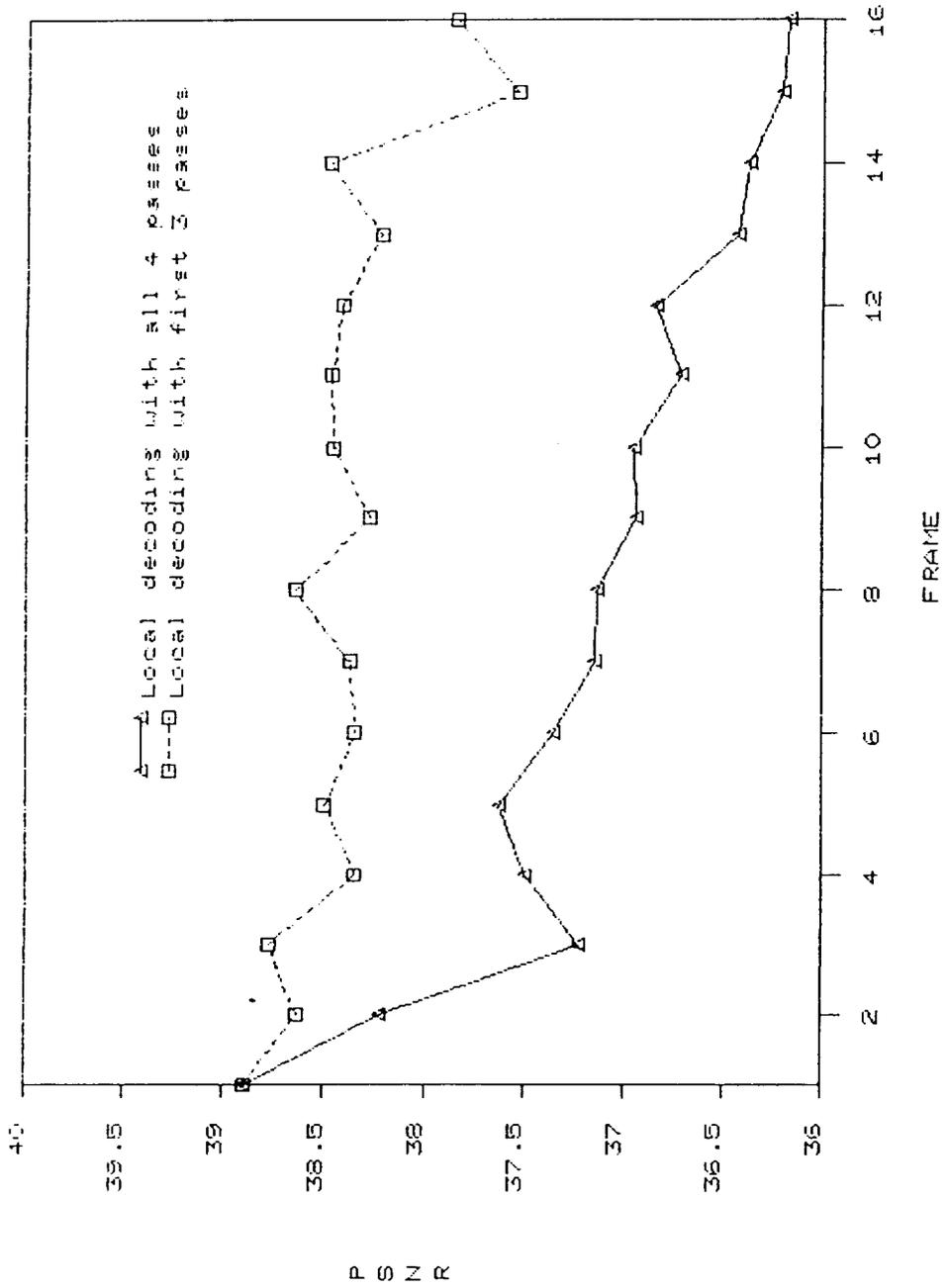


Figure 5.4 Performances with packet losses from pass 4.

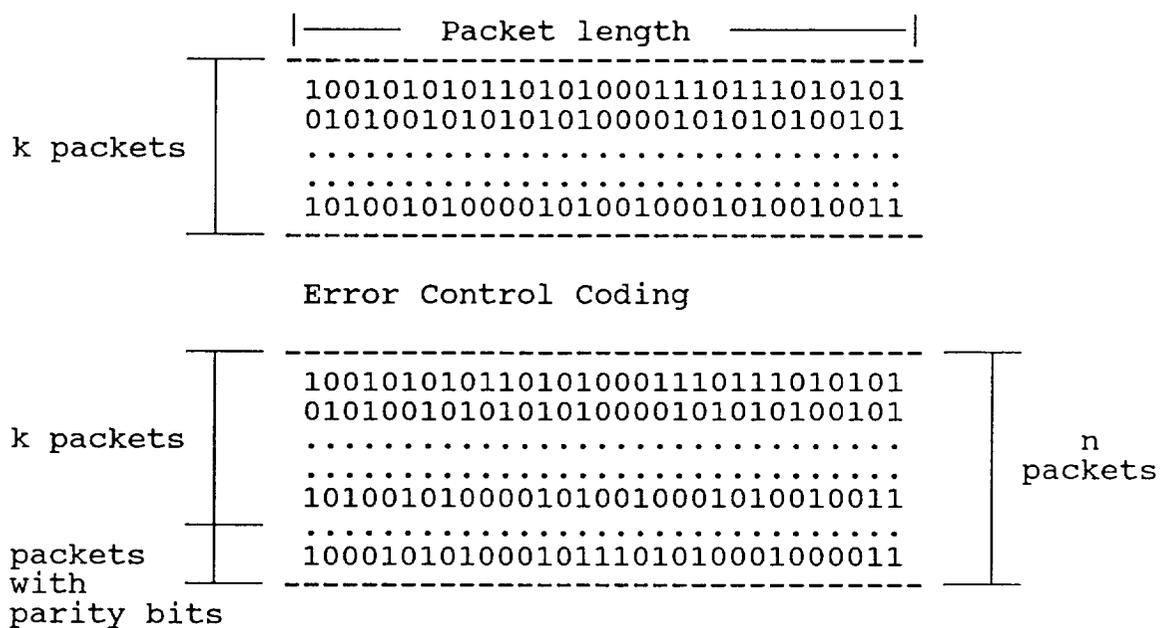


Figure 5.5 Error control coding applied perpendicular to the direction of packetization.

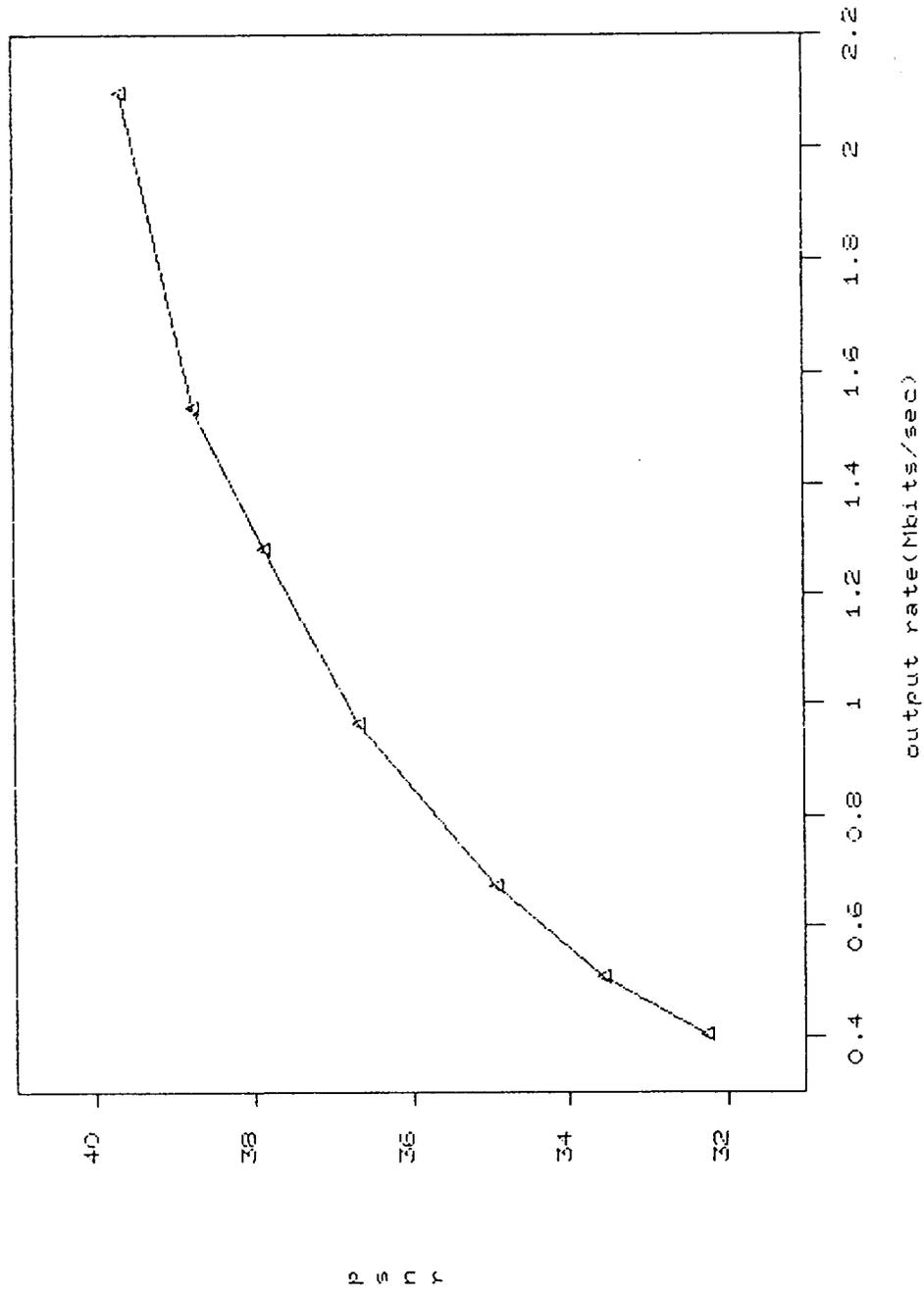


Figure 5.6 PSNR versus video output rate with 30 frames per second.

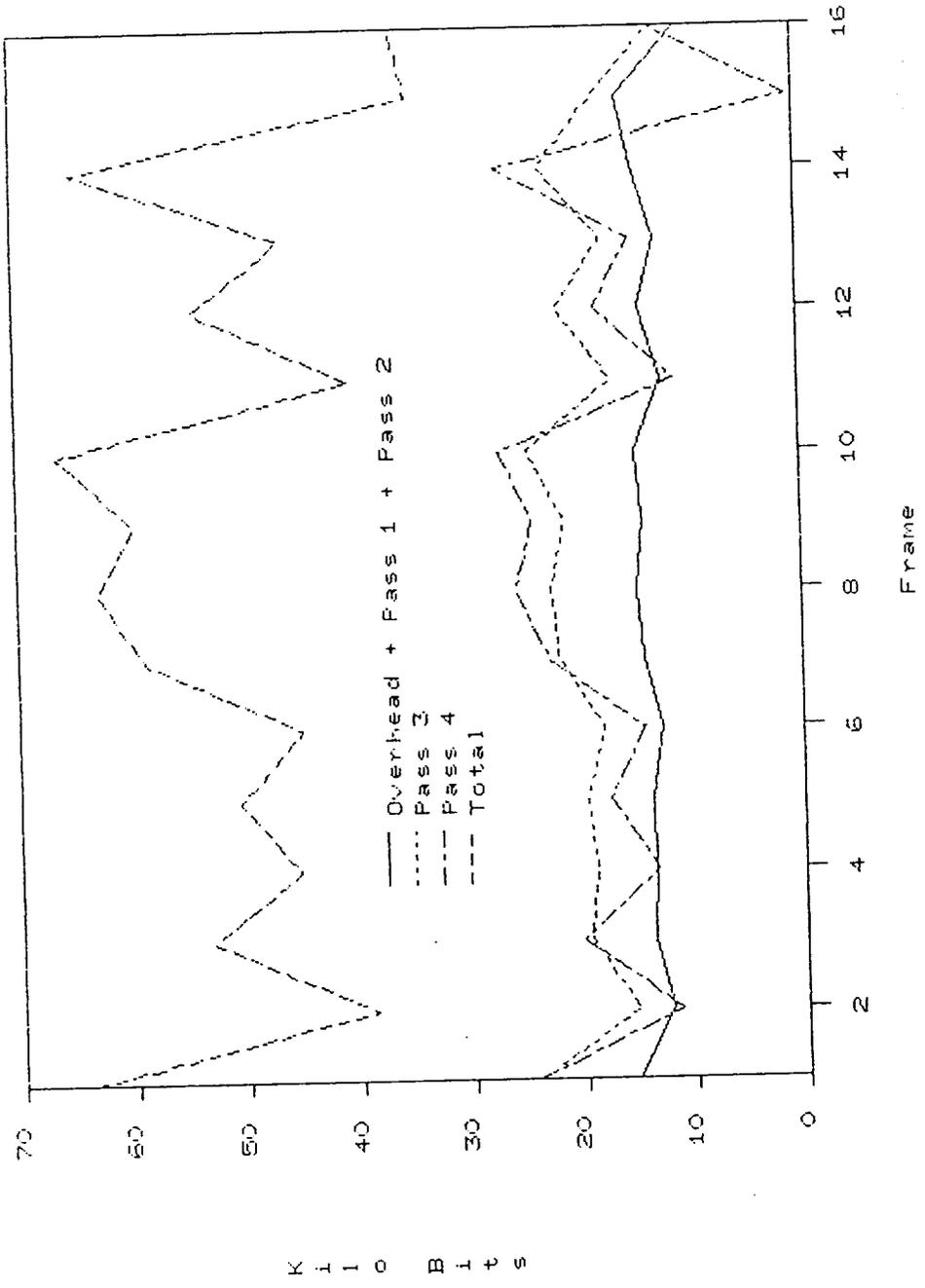


Figure 5.7 Data rate of simulation sequence frames.

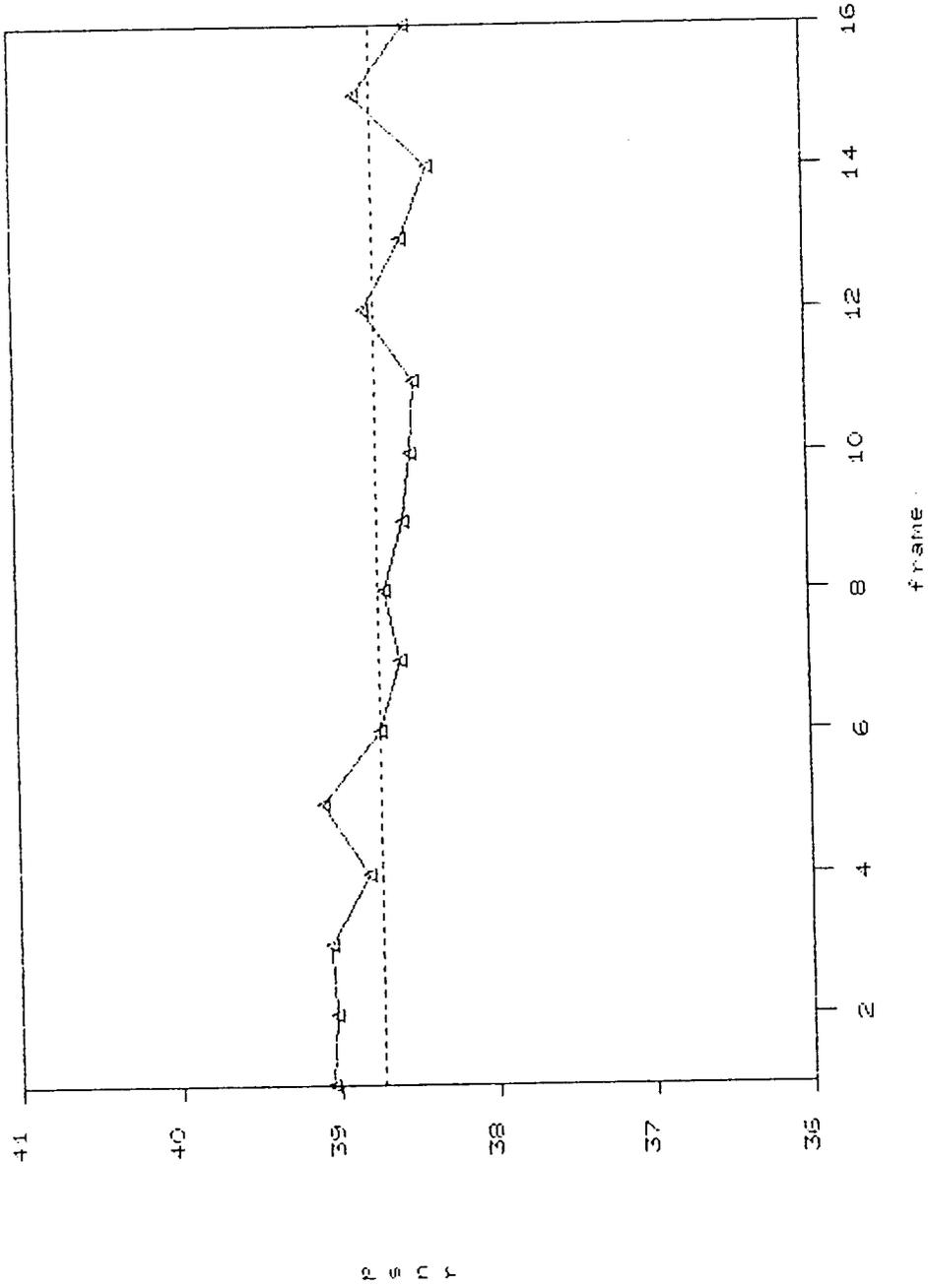


Figure 5.8 PSNR of simulation sequence frames.

2.2

FRAME	OVER- HEAD	PASS1	PASS2	PASS3	PASS4	TOTAL
1	2588	4352	8400	24248	24416	64004
2	1772	4352	5992	15232	11312	38660
3	2156	4352	7168	19432	20104	53212
4	2088	4352	6888	18760	13216	45304
5	2164	4352	7112	19600	17416	50644
6	1988	4352	6328	17920	14336	44924
7	2352	4352	7448	21896	22736	58784
8	2432	4352	7952	22512	25704	62952
9	2316	4352	7504	21336	24136	59644
10	2568	4352	7840	24528	26992	66360
11	1892	4352	6048	16856	11144	40292
12	2352	4352	7616	21728	18200	54248
13	1968	4352	6384	17584	15008	46296
14	2468	4352	7840	23128	26936	64734
15	2216	4352	9352	18088	728	34736
16	1496	4352	4536	12824	12936	36164
TOTAL	34816	69632	114408	315672	287392	820992
MEAN	2176	4352	7150	19729	17962	51312
DEVIATION	290	0	1094	3179	7000	10395

Table 5.1 Output bit rate for each and total pass.
The unit is bits.

	OVER- HEAD	PASS1	PASS2	PASS3	PASS4	TOTAL
MEAN	65.28	130.56	214.50	591.87	538.86	1539.36
DEVIATION	8.70	0.00	32.82	95.37	210.00	311.85
MAXIMUM	77.04	130.56	280.56	735.84	821.52	1990.80
MINIMUM	44.88	130.56	136.08	384.72	21.84	1042.08

Table 5.2 Output bit rate for each and total pass calculated with 30 frames/sec video rate. The maximum and minimum values are the instantaneous rates, which correspond to the respective maximum and minimum number of bits needed to encode a particular frame in the sequence. The unit is kilobits.



Figure 5.9 Frame 3 of simulation sequence.



Figure 5.10 Frame 4 of simulation sequence.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

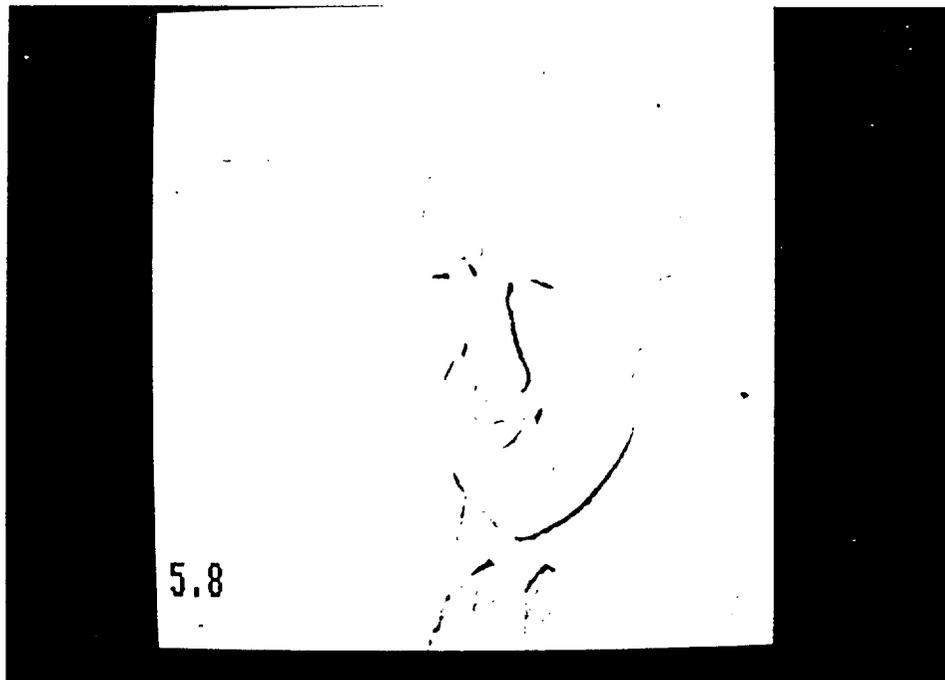


Figure 5.11 Difference image of frame 3 and 4.



Figure 5.12 Frame 9 of simulation sequence.



Figure 5.13 Frame 10 of simulation sequence.



Figure 5.14 Difference image of frame 9 and 10.



Figure 5.15 The effect of pass 4 packet loss for frame 4.



Figure 5.16 The effect of pass 4 packet loss for frame 10.



Figure 5.17 The effect of pass 1 packet loss for frame 3.



Figure 5.18 The effect of pass 1 packet loss for frame 9.

Chapter 6 Conclusions

Chapter 1 and 2 described the environment of the future's telecommunications and proposes some specifications for integrating signal processing into this environment. Chapter 3 introduced the basic materials of data compression and investigated the characteristics of MBCPT. Chapter 4 gave a view of the network simulator used for these tests and the modifications which were required. Chapter 5 proposed the differential scheme of MBCPT as a packet video coder and showed its performance.

The network simulator was used only as a channel in this simulation. In fact, before the real-time processor is built, a lot of statistics can be collected from the network simulator to improve upon the coding scheme. These include transmission delays and losses from various passes under different network loads. For resynchronization, the delay jitter between received packets can also be estimated from this simulation.

The environment for tomorrow's telecommunications has been described and requires a flexibility which is not possible in circuit switching network. MBCPT has the appealing properties like high compression rate with good visual performance, robustness to packet lost, tractable integration with network mechanics and simplicity in parallel implementation. Some more considerations have been

proposed for the whole packet video system like designing protocols, packetization, error recovery and resynchronization. For fast moving scenes, the differential MBCPT scheme seems insufficient. Motion compensation, error concealment or even attaching function commands into the coding scheme are believed to be useful tools for increasing the performance and will be the direction of future research.

References

- [1] K. Smalheiser and C. Hughes, "The Imaging Revolution," FORTUNE, vol.119, No.13, June 1989.
- [2] W. Verbiest and L. Pinnoo, "A Variable Bit Rate Codec for Asynchronous Transfer Mode Networks," IEEE J. Select. Areas Commun., vol.7, pp.801-806, June 1989.
- [3] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks," IEEE J. Select. Areas Commun., vol.7, pp.801-806, June 1989.
- [4] J. Darragh and R. Baker, "Fixed Distortion Subband Coding of Images for Packet-Switched Networks," IEEE J. Select. Areas Commun., vol.7, pp.801-806, June 1989.
- [5] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM networks," IEEE J. Select. Areas Commun., vol.7, pp.801-806, June 1989.
- [6] G. Karlsson and M. Vetterli, "Packet Video and Its Integration into Network Architecture," IEEE J. Select. Areas Commun., vol.7, pp.739-751, June 1989.
- [7] E. Daly and T. R. Hsing, "Variable Bit Rate Vector Quantization of Video Images for Packet-Switched Networks" in Proc. ICASSP, vol. 2, pp. 1160-1163, 1988.
- [8] H. C. Reeve and J. S. Lim, "Reduction of Blocking Effect in Image Coding," in Proc. ICASSP, vol. 3, pp. 1212-1215, 1983.
- [9] P. Carrereau, D. Staelin and G. Jager, "Encoding of Images Based on a Lapped Orthogonal Transform," IEEE Tran. Commun., vol. 37, pp. 189-193, February 1989.
- [10] Anil K. Jain, Fundamentals of Digital Image Processing, Englewood Cliffs, NJ: Prentice Hall, 1989.
- [11] W. Chen, C. Smith and S. Fralick, "A Fast Communicational Algorithm for the Discrete Cosine Transform," IEEE Trans. Commun., vol. COM-26, pp. 934-936, June 1978.
- [12] S. S. Huang, "Source Modelling for Packet Video," IEEE Int. Conf. on Commun., vol. 3, pp. 1262-1267, 1988.
- [13] S. L. Casner, D. C. Cohen and E. R. Cole, "Issues in Satellite Packet Video Communication," IEEE Int. Conf. on Commun., vol. 1, pp. 34-38, 1988.
- [14] K. Sloan, Jr. and S. Tanimoto, "Progressive Refinement of Raster Scan Images," IEEE Trans. Comput., vol. COM-28,

pp. 871-874, Nov. 1979.

[15] H. M. Dreizen, "Content-Driven Progressive Transmission of Grey-Scale Images," IEEE Trans. Commun., vol. COM-35, pp. 289-296, March 1987.

[16] D. Vaisey and A. Gersho, "Variable Block-Size Coding," Proc. ICASSP, pp. 1051-1054, April 1987.

[17] Y. S. Ho and A. Gersho, "Variable-Rate Multi-Stage Vector Quantization for Image Coding" in Proc. ICASSP, vol. 2, pp. 1156-1159, 1988.

[18] K. Tzou, T. Chen, P. Fleischer and M. Liou, "Compatible HDTV Coding for Broadband ISDN," in Proc. GLOBECOM, vol. 2, pp. 743-749, 1988.

[19] L. Chiariglione and L. Contin, "Two-Channel Coding for Packet Video Transmission," IEEE Int. Conf. on Commun., vol. 1, pp. 413-417, 1988

[20] D. Nelson, K. Sayood, G. Ankenman and H. Chang, "Pnetsim System Programmer's Manual," University of Nebraska-Lincoln, Final Report for ARMY Contract DAAB07-85-K-K535, Dec. 1986.

7 Appendix C

Mixture Block Coding with Progressive Transmission in Packet Video

Y. C. Chen, K. Sayood and D. J. Nelson

Department of Electrical Engineering

Center for Communication and Information Science

University of Nebraska-Lincoln

I Introduction

Due to the rapidly evolving field of image processing and networking, video information is promising to be an important part of tomorrow's telecommunication system. Up to now, video transmission has been mainly transported over circuit-switched networks. It is quite likely that packet-switched networks will dominate the communications world in the near future. Asynchronous transfer mode(ATM) techniques in broadband-ISDN can provide a flexible, independent and high performance environment for video communication. Therefore, it is necessary to develop techniques for video transmission over such networks.

The classic approach in circuit switching is to provide a "dedicated path" thus reserving a continuous bandwidth capacity in advance. Any unused bandwidth capacity on the allocated circuit with circuit-switching is therefore wasted. Rapidly varying frequency signals, like video signals, require too much bandwidth to be accommodated by a standard circuit-switching channel. With a certain amount of capacity assigned to a given source, if the output rate of that source is larger than the channel capacity, quality will be degraded. If the generating rate is less than the limit, the excess channel is wasted. Channel sharing protocol between independent sources can improve channel utilization. Another point that strongly favors packet-switched networks is the possibility that the integration of services in a network will be facilitated if all of the signals are separated into packets with the same format.

Some coding schemes which support the packet video idea have been exploited. Verbiest and Pinnoo proposed a DPCM-based system which is comprised of an intrafield/interframe predictor, a nonlinear quantizer, and a variable length coder[1]. Their codec obtains stable picture quality by switching between three different coding modes: intrafield DPCM, interframe DPCM, and no replenishment. Ghanbari has simulated a two-layer conditional replenishment codec with a first layer based on hybrid DCT-DPCM and second layer using DPCM[2]. This scheme generates two type of packets: "guaranteed packets" contain vital information and "enhancement packets" contain "add-on" information. Darragh and Baker presented a sub-band codec which attains a user-prescribed fidelity by allowing the encoder's compression rate to vary[3]. The codec's design is based on an algorithm that allocates distortion among the sub-bands to minimize channel entropy. Kishino et al. describe a layered coding technique using discrete cosine transform coding, which is suitable for packet loss compensation[4]. Karlsson and Vetterli presented a sub-band coder using DPCM with a nonuniform quantizer followed by run-length

coding for baseband and PCM with run-length coding for nonbaseband[5]. In this paper, a different coding scheme called MBCPT is investigated. Unlike those methods mentioned above, MBCPT doesn't use decimation and interpolation filters to separate the signals into sub-bands. But it has the property of sub-band coding by using variable blocksize transform coding. The paper is organized as follows. First, some of the important characteristics and requirements about packet video are discussed. In Section 3, the coding scheme called Mixture Block Coding with Progressive Transmission is presented. In Section 4, a network simulator used in the paper is introduced. In Section 5, the simulation result is discussed. Finally, in Section 6 the paper is summarized.

II. Characteristics of Packet Video

The demand for various services, such as telemetry, terminal and computer connections, voice communications, and full-motion high-resolution video, and the wide range of bit rates and holding times they represent, provide an impetus for building a Broadband Integrated Service Digital Network(B-ISDN). B-ISDN is a projected worldwide public telecommunications network that will service a wide range of user needs. Furthermore, the continuing advances in the technology of optical fiber transmission and integrated circuit fabrication have been the driving forces to realize the B-ISDN. The idea of B-ISDN is to build a complete end-to-end switched digital telecommunication network with broadband channel. Still to be precisely defined by CCITT, with fiber transmission, H4 has an access rate of about 135 Mbps.

Packet-switched networks have the unique characteristics of dynamic bandwidth allocation for transmission and switching resources, and the elimination of channel structure[6]. It acquires and releases bandwidth as it is needed. Because the video signals vary greatly in bandwidth requirement, it is attractive to utilize a packet-switched network for video coded signals. Allowing the transmission rate to vary, video coding based on packet transmission permits the possibility keeping picture quality constant, by implementing "bandwidth on demand". There are three main merits when transmitting video packets over a packet switching network:

1. Improved and consistent image quality: if video signals are transmitted over fixed-rate circuits, there is a need to keep the coded bit rate constant, resulting in image degradation accompanying rapid motion.
2. Multimedia integration: as mentioned above, integrated broadband services can be provided using unified protocols.
3. Improved transmission efficiency: using variable bit-rate coding and channel sharing among multiple video sources, scenes can be transmitted without distortion if other sources, at the same time, are without rapid motion.

However video transmission over packet networks also has the following drawbacks:

1. The time taken to transmit a packet of data may change from time to time.

2. Packets of data may arrive very late or even get lost.
3. Headers of packets may be changed because of errors and delivered to the wrong receiver.

It has to be emphasized that the delay/lost effect can reach very high levels if there are a lot of users accessing the network and may seriously damage the quality of the image.

When the signals transmitted in the network are nonstationary and circuit-switching is applied, a buffer between the coder and the channel is needed to smooth out the varying rate. If the amount of data in the buffer exceeds a certain threshold, the encoder is instructed to switch into a coding mode that has lower rate but worse quality to avoid buffer overflow. In packet-switched network, Asynchronous Time Division Multiplexing (ATDM) can efficiently absorb temporal variations of the bit-rate of individual sources by smoothing out the aggregate of several independent streams in common network buffers.

To deliver packets in a limited time and provide a real time service is a difficult resource allocation and control problem, especially when the source generates a high and greatly varying rate. In packet-switching networks, packet losses are inevitable but use of a packet-switching network yields a better utilization of channel capacity. The video coder will require different channel capacity over time but the network will provide a channel whose capacity changes depending on the traffic in the network.

Therefore, the interactions between the coder and the network have to be considered and be incorporated among the requirements for the coder. These requirements include:

1. Adaptability of the coding scheme: The video source we are dealing with has varying information rate. So it is expected that the encoder should generate different bit rates by removing the redundancy. When the video is still, there is no need to transmit anything.
2. Insensitivity to error: The coding scheme has to be robust to the packet loss so that the quality of the image is never seriously damaged. Remember that retransmission is impossible because of the tight timing requirement.
3. Resynchronization of the video: Because the varying packet-generating rate and the lack of a common clock between the coder and the decoder, we have to find a way to reconstruct the received data which is synchronous to the display terminal.
4. Control of coding rate: Sensing the heavy traffic in the network, the coding scheme is required to adjust the coding rate by itself. In the case of a congested network, the coder could be switched to another mode which generates fewer bits while degrading image quality.
5. Parallel architecture: The coder should preferably be implemented in parallel. That allows the coding procedure to be run at a lower rate in many parallel streams.

In the next section, we investigate a coding scheme to see if it satisfies the above requirements.

III. Mixture Block Coding with Progressive Transmission

Mixture Block Coding (MBC) is a variable-blocksize transform coding algorithm which codes the

image with different block sizes depending upon the complexity of that block area. Low-Complexity areas are coded with a large block size transform coder while high-complexity regions are coded with a small block size one. The complexity of the specific block is determined by the distortion between the coded and original image. A more complex image block has higher distortion. The advantage of using MBC is that it does not process different complex region with the same block size. That means MBC has the ability to choose a finer or coarser coding scheme to deal with different complex parts of the same image. With the same coding source (coding rate), MBC is able to increase the quality of the whole image than a coding scheme which codes different complex region with the same block size coder.

When using MBC, the image is divided into maximum block size blocks. After coding, the distortion between the reconstructed and original block is calculated. The processing block is subdivided into smaller block size blocks if that distortion fails to meet the predetermined threshold. The coding-testing procedure continues until the distortion is small enough or the smallest block size is reached. In this scheme, every block is coded until the reconstructed image is satisfactory then moves to the next block.

As for Mixture Block Coding with progressive transmission (MBCPT), it is a coding scheme which combines MBC and progressive coding. Progressive coding is an approach that allows an initial image to be transmitted at a lower bit rate and to be refined with an additional bit rate[7]. In this way, successive approximations converge to the target image with the first approximation carrying the "most" information and the following approximations enhancing it. The process is like focusing a lens, where the entire image is transformed from low-quality into high-quality[8]. In progressive coding, every pixel value or the information contained in it is possibly coded more than once and the total bit rate may increase due to different coding scheme and quality desired. Because only the gross features of an image are being coded and transmitted in the first pass, the processing time is greatly reduced for the first pass and a coarse version of the image can be displayed without significant delay. It has been shown that it is perceptually useful for perception to get a crude image in a short time, rather than waiting a long time to get a clear complete image[9].

With different stopping criterion, progressive coding is suitable for dynamic channel capacity allocation. If a predetermined distortion threshold is met, processing is stopped and no more refining action is continued. The threshold value can be adjusted according to the traffic condition in the channel. Successive approximations (or iterations) are sent through the channel in progressive coding and lead the receiver to the desired image. If these successive approximations are marked with decreasing priority, then a sudden decrease in channel performance may only cause the received image to suffer from quality degradation rather than total loss of parts of the images[8].

MBCPT is a multipass scheme in which each pass deals with different block sizes. The first pass codes the image with maximum block size and transmits it immediately. Only those blocks which fail to meet the distortion threshold go down to the second pass which processes the difference image block

coming from the original and coded image obtained in the first pass with smaller blocksize blocks. The difference image coding scheme continues until the final pass which deals with the minimum blocksize block. At the receiving end, a crude image is obtained from the first pass in a short time and the data from following passes serve to enhance it. Fig. 1 shows the structure of pass 16x16 for MBCPT. Fig. 2 shows the parallel structure of MBCPT. A coding structure like a quad tree is proposed by Dreizen[10], Vaisey and Gersho[11] which subdivides those busy blocks into four pieces and will be used in this paper. In the quad tree coding structure of this paper, the 16x16 block is coded and the distortion of the block is calculated. If the distortion is greater than the predetermined threshold for 16x16 blocks, the block is divided into four 8x8 blocks for additional coding. This coding-checking procedure is continued until the only image blocks not meeting the threshold are those of size 2x2. Figure 3 shows the algorithm.

Considering the block size, it should be small enough for ease of processing and storage requirements, but large enough to limit the inter-block redundancy[12]. Larger block size results in higher image quality, but it is very difficult to build real-time hardware for block sizes larger than 16x16 because the number of calculations increase exponentially with block size for the DCT transform[8]. So, 16x16 is chosen to be the largest blocksize in here. The minimum blocksize determines the finest visual quality that is achievable in the busy area. If the minimum blocksize is too large, it is likely to observe the blockiness in the coded edge of spherical object because the coding block is square. In order to match the zonal transform coding used in this paper, 2x2 is the smallest blocksize and there are four passes (16x16, 8x8, 4x4, 2x2) in this scheme. Fig. 4-7 show the images from 4 passes individually.

After discrete cosine transform, only four coefficients including the dc and three lowest order frequency coefficients are coded and others are set to zero. The dc coefficient in the first pass is coded with 8-bit uniform quantizer due to the fact that it closely reflects the average gray level for that image block and is hard to predict. It is easy to predict the dc coefficient in the following pass because it is a residual and distributes like a laplacian model. Typically, a 5-bit optimal laplacian nonuniform quantizer is used. The three ac coefficients, as mentioned above, distribute like the laplacian model with a variance greater than that of the dc coefficient. Because different variances are exhibited for different coefficients, the input samples are first normalized so that they have unit variance and therefore can be used with the same 5-bit laplacian quantizer. As an alternative, an LBG vector quantizer with a 512 codebook size is used to quantize the vector which comprises the three ac coefficients. The threshold of each pass has to be selected before the coder is going to work and it is readjustable during the operation according to the channel condition and quality required.

Because only partial blocks which fail to meet the distortion threshold need to be coded, there must be some side information to instruct the receiver how to reconstruct the original image back. One bit of overhead is needed for each block. If a block is to be divided, a 1 is assigned to be its overhead; if not,

a 0 is assigned. A coding process in Fig. 8 has the following overhead: 1,1001,1001,1001,1001,1001.

The interframe coder used in this paper is a differential scheme which is based on MBCPT. This coder processes the difference image coming from the current frame and the previous frame which is locally decoded from the first three pass data. Fig. 9 shows the algorithm of this coder. Fig. 10 shows a different scheme which does the local decoding with all four passes. From Fig. 11, when there is no packet get lost, the performances of these two schemes are quite the same. But when congestion happened in the network, with the priorities assigned to packets, packets from pass 4 are expected to be discarded first. In this case, the performance (from Fig. 12) of scheme in Fig. 9 is much better than the one in Fig. 10. Therefore the coding scheme in Fig. 9 is used in our simulation. In this paper, the Kronkite motion picture with 16 frames is used as the simulation source. Every image is 256x256 pixels with graylevels ranging from 0 to 255. It is similar to a video conferencing type image which has neither rapid motion nor scenes changes. Due to this characteristic, advanced techniques like motion detection or motion compensation are not used but could be implemented when dealing with broadcasting video.

From the datastream output that is listed in the Table 1, we can see that the data in pass 4 which represents 30-40% of the entire data and is deemed as a less significant pass(LSP). This part of the data is going to increase the sharpness of the image and is usually labeled with the lowest priority in network. With a substantial possibility of being discarded due to low priority, those packets from pass 4 won't be used to reconstruct the locally decoded image and be stored in the frame memory. That is supposed to avoid the packet loss error propagating into following frames if the lost packet truly belongs to pass 4.

IV. Simulation Network

The network simulator to be used for this paper would be a modification of an existing simulator developed by Nelson et al.[13]. A brief description of the simulator is provided here.

A. Introduction

As mentioned in section 2, tomorrow's integrated telecommunication network is a very complicated and dynamic structure. Their efficiency requires sophisticated monitoring and control algorithms with communication between nodes reflecting the existing capacity and reliability of system components. The scheme for communicating information regarding the operating status are called the system protocols. Since this communication of system information must flow through the channel, it reduces the overall capacity of the physical layers, but hopefully provides a more efficient system overall. Therefore, the optimal system efficiency depends a lot upon these protocols, in turn, upon the system topology, communication channel properties, nodal memory and component reliability. Most network protocols have been developed around high reliability in topological structures with reasonable high channel reliability.

In order to fit into the purpose of this paper, most modifications which have been made to this simulator are basically in those modules concerning network layers. And this simulator is structured in modules which represent, to some degree, the ISO Model for packet switched networks. Therefore, a more detailed description about the network layer modules will be made next.

B. The Network Layers

Each layer of the simulator module contains a processor and one or more packet queues. The processor is idle before there is a packet coming into its associate queue. The packet and the task that must be performed are entered into "SIM_Q", a queue which drives the simulator, with a completion time. When the task is performed, that means the completion time has arrived, the queue is checked. If there is another task to be performed, then its completion is scheduled. If the queue is empty the processor is marked idle again. The layers in the simulator are quite close in operation to the ISO transport, network and datalink layers. A "partial" session layer exists principally as a reporting layer for end to end statistics.

1) The Session Layer

In the OSI model, the session layer allows users on different machines to establish "sessions" between them. In the simulator, as mentioned above, it is a relatively simple model of the subscribers and an end to end statistics collector. At message arrival time, the session layer generates the message with all of its randomly selected attributes and if flow control or node hold-down are not in effect, submits it to the transport layer and then builds up the next message arrival time. During initialization, a task "SL(Session Layer)_Rcv_Msg" for each node is queued in SIM_Q for the arrival time of the first message at that node. When this task is executed by the simulator, a message packet is generated and placed in the transport queue. The arrival of the next message is then queued in SIM_Q with the same task and an arrival time determined by the random number generator (Poisson distributed). The only other task performed at the session layer is the "SL_Snd_Msg" task which simulates the delivery to the subscriber. In the simulator, this is principally a "bookkeeping task" that records message statistics and "cleans up" the queues containing packets with resolved references.

2) The Transport Layer

The basic function of the transport layer is to receive the message from the session layer, separate it up to smaller units if necessary, pass these to the network layer and make sure these pieces will arrive sequentially at the other end. Furthermore, all this work is expected to be done efficiently, and in a way that isolates the session layer from the future progressive in the hardware technology. In the simulator, the transport layer simulates packetization, reassembly, message acknowledgement and resubmittal in the case that a message acknowledgement is not received in time, transport-layer timeout. There are four tasks simulated by the transport layer. They are "TL(Transport Layer)_Packetize", "TL_Timeout", "TL_Reassemble", and "TL_Ack_Send". It is recognized that in some networks, packetization takes place at the network level, leaving the transport layer responsible

only for message level structures. Reassembly, depending upon the protocol can take place as low as the datalink level. These tasks were both placed in the transport layer for ease of coding, but are separate modules that could be quite easily extracted and placed elsewhere. Also, the system was originally designed for datagram operation and since the packets will not necessarily arrive in order, it is unlikely that assembly would take place at the datalink level.

3) The Network Layer

The network layer is concerned with controlling the operation of the network. A key design issue is determining how packets are routed from source to destination. Another issue is how to avoid the congestion caused in the case if too many packets are presented into the network at the same time. In the simulator, the network layer performs all of the functions related to these two aspects with the exception of flow control which takes place at the session layer, and the recovery protocols which require some service from the datalink layer. It also activates new channels when needed and determines when packets originating at other nodes are to be discarded. The network layer is currently the most dynamic with regard to the coding of modules. Five modules currently comprise the network layer. These include relatively static modules; one module for dialing up new lines when more line capacity is required and releasing them when not needed; one module for the network processor and queue handling and one module for the routines which are common to most routing algorithms. This leaves two modules for the dynamic parts of the routing and flow control algorithms.

4) The Datalink Layer

The main task of the datalink layer is to take a raw transmission facility and transform it into a line that appears free of transmission errors to the network layer. It simulates the sending of the message over the channel and the delivery at the other end. When a packet is received, the datalink acknowledgement is initiated either by the piggy-back acknowledgement or by generating a datalink acknowledgement packet. As mentioned previously, the datalink level also simulates the physical layer on a statistical basis. If correct transmission was indicated (through a random number generator) then acknowledgement was also assumed. Current datalink layer simulation modules include generation of acknowledgement packets and simulation of the piggy-back acknowledgement as well. When a line is "brought up", health packets are used to establish initial connections. Also, when a line "goes down", an active node will immediately issue health check packets to ascertain when the channel is again available.

C. Modifications

A major problem of using this system as a simulation tool for the study of packet video is that the system doesn't actually transmit the data from node to node. While a packet is transmitted, the data field is empty. Therefore modifications had to be made to the simulator to accommodate the video data. In the sending node, a field called "Image" which contains real image data is attached to the record "Packet_Ptr" allocated to the message generated in the session layer. There are three new

modules in this layer. First, "Get_Image" puts the image data into the image field of a message generated at a specific time and node. Second, "Image_Available" checks to see if there is still any image data needed to be transmitted. If that is true, the following message generated at that specific node is still the image message and contains some image data. Third, "Receive_Image" collects the image data in the session layer of the receiving node when the flag "Image_Complete" is on. In module "Session_Msg_Arrive", different priority is assigned to different messages. In module "Session_Msg_Send", some statistics are calculated including the number of lost image packets and the transmission delay for image packets.

Currently, the transport layer simply duplicates the same packet with different assigned sequential packet numbers without actually packetizing the message. The module "Transport_Packetize" is modified to really packetize the image data which resides in the message record queued in "Transport_Q" when it is called. The module "Transport_Reassemble" is called to reassemble these image packets according to their packet number when the flag "Image_Content" defined in "Packet_Ptr" is true. The network layer is responsible for routing and flow-control. This module is already very well developed, so the modifications to be performed here were relatively minor. In the datalink layer, in order to simulate the delivery of packets through the channel, a new packet will be generated at the receiving node and the information including the image data from the transmitted packet (which will still be resident at the sending node) will be copied into it. With the bit-error-rate defined in the program Topology, transmission success rate will be set and bit errors can be inserted in both the data and control bits in the packet. Errors in the control bits are simulated separately as long as the error rates are consistent. If an error in control bits really occurs, the transmission fails and needs to be sent again depending on the threshold of the timeout number. Besides the modifications made in those layer modules, we still have to arrange some new memory elements allocated for image messages and packets. In order to make sure the simulation is run in the steady state, image data is available after some simulation time.

V. Interaction of the Coder and the Network

When the video data is packed and sent into a nonideal network, some problems that emerge and are discussed in the following section.

A. Packetization

The task of the packetizer is to assemble video information, coding mode information, if it exists, and synchronization information into transmission cells. In order to prevent the propagation of the error resulting from the packet loss, packets are made independent of each other and no data from the same block or same frame is separated into different packets. The segmentation process in the transport layer has no information regarding the video format. Avoiding the bit stream being cut randomly, the packetization process has to be integrated with the encoder which is in the presentation layer of user's

premise. Otherwise, some overhead has to be added into the datastream to guide the transport layer doing the correct packetization. In order to limit the delay of packetization, it is necessary to stuff the last cell of a packet video with dummy bits if the cell is not completely full.

Every packet must contain an absolute address which indicates the location of the first block it carries. Because every block in MBCPT has the same number of bits in each pass, there is no need to indicate the relative address of the following blocks contained in the same packet. There always exists a tradeoff between packaging efficiency and error resilience. If error resilience is considerable, one packet should contain a smaller number of blocks. However, since each channel access by a station contains an amount of overhead, the packet should be long for transmission efficiency. Fixed length packetization is used in this paper for simplicity.

B. Error Recovery

There is no way to guarantee that packets won't get lost after being sent into the network. Packet loss can be mainly attributed to two problems. First, bit errors can occur in the address field, leading the packets astray in the network. Second, congestion can exceed the networks management ability and packets are forced to be discarded due to buffer overflow. Effect created by higher pass packet (like pass 4) loss in MBCPT coding will be masked by the basic passes and replaced with zeros. The distortion is almost invisible when viewing at video rates because the lost area is scattered spatially and over time. However, low pass packets (like pass 1) loss, though rare due to high priority, will create erasure effect due to packetization and the effect is very objectionable.

Considering the tight time constraint, retransmission is not feasible in packet video. It may also result in more severe congestion. Thus, error recovery has to be performed by the decoder alone. In our differential MBCPT scheme, the packets from pass 4 are labeled lowest priority and form a great part of the complete data. These packets can be discarded whenever network congestion occurs. That will reduce the network congestion and won't cause too much quality degradation. The erasures caused by basic pass loss is simply covered with the reconstructed values from the corresponding area in the previous frame. This remedy seems insufficient even when there is only small amount of motion in that area. Motion detection and motion compensation could be used to find a best matched area in the previous frame for replacement.

Side information in the MBCPT decoding scheme is very important. So, this vital information is not allowed to get lost. Two methods can be used for protection. First, error control coding, like block codes or convolutional codes, can be applied in both direction along with and perpendicular to the packetization. The former is for bit error in the data field while the latter is for packet loss. The minimum distance that the error control coding should provide depends on the network's probability of packet loss, correlation of such loss and channel bit error rate. Second, from Table 1, we can see that the output rate of side information and pass 1 and even pass 2 is quite steady. It seems feasible to allocate an amount of channel capacity to these outputs to ensure their timely arrival. That means

circuit switching can be used for important and steady data.

C. Flow Control

In order to shield the viewer from severe network congestion, there are some flow control schemes which are considered useful. If there is an interaction between the encoder and the transport layer, then the encoder can be informed about the network condition. Depending on that, the encoder can adjust its coding scheme. In the MBCPT coding scheme, if the buffer is getting full, that means that the bit generating rate is overwhelming the packetization rate and the encoder will switch to a coarse quantizer with fewer steps or loosen the threshold to decrease its output rate. In this way, smooth quality degradation is obtainable. This will also complicate the encoder design.

It is possible to use the congestion control of the network protocols to prevent the drastic quality change by assigning different priorities to packets from different passes. Without identifying the importance of each packet and discarding packets blindly sometimes brings disaster and cause a session shut down, for example if the side information gets lost. In the MBCPT coding scheme, side information and packets from pass 1 are assigned highest priority and higher pass packets are assigned with decreasing priority.

D. Interaction with protocols

In the ISO model, physical, datalink and network layers comprise the lower layers which form a network node. The higher layers have transport, session, presentation and application layers and typically reside in a customer's premises. The lower layers have to do nothing about the signal processing and only work as a "packet pipe". The physical layer requires adequate capacity and low bit-error-rate which are determined only by technology. The datalink layer can only deal with link-management because all the mechanics like requesting retransmission is not feasible in packet video transmission. The network layer has to maintain orderly transmission by deleting the delay jitter with input buffering. Otherwise, it can take care the network congestion by assigning transmission priority.

As the higher layers reside in the customer's premises, it performs all the functions of the packet video coder. The transport layer does the packetization and reassembly. The packet length can be fixed or variable. Fixed packet length simplifies segmentation and packet handling while a variable packet length can keep the packetization delay constant. The session layer supervise set-up and tear-down for sessions which have different types and quality. There is always a tradeoff between quality and cost. The quality of a set-up session can be determined by the threshold in the coding scheme and the priority assignment for transmission. Of course, the better the quality, the higher the cost. Fig. 13 shows the tradeoff between PSNR and video output rate by adjusting thresholds. The presentation layer does most of the signal processing, including separation and compression. Because it knows the video format exactly, if any error concealment is required, it will be performed here. The application layer works as a boundary between the user and the network and deals with all the analog-digital signal conversion.

VI. Results from Packet Video Simulation

Some results were obtained in this packet video simulation and it shows that a pretty high compression and the associated image quality can be obtained using this differential MBCPT scheme. The monochrome sequence used in this simulation contains 16 frames, each of size 256x256 pixels with 8 bits per pixel, corresponds to a bit rate of 15.3 Mbits/s, given a video rate of 30 frames/s. As Table 2 shows, the average data rates of our system is 1.539 Mbits/s. The compression rate is about 10 with a mean PSNR of 38.74 dB. Fig. 14 shows the data rate of the sequence frames with sideinformation, 4 passes and total rate. It is clear that data rate of pass 1 is constant as long as the quantization mode keeps the same. Sideinformation and data from pass 2, even pass 3, is quite steady and is referred as Most Significant Pass (MSP). The data rate of pass 4 is bursty, highly-uncorrelated and is called Less Significant Pass (LSP). Fig. 15 shows the PSNR for each frame in the sequence. The standard deviation is only 0.2 dB. In the simulation, the same threshold is used throughout the sequence. If constant visual quality is desired, a varying threshold can be used for different frames. That will generate a much more varying bit rate and of course motion detection is required. Comparing these two figures, it seems true that a varying bit rate can support constant quality video.

From the difference images of this sequence, frames 1-8 seem quite motionless while frames 9-13 are with substantial motion. We adjust the traffic condition of the network to force some of the packets to get lost and check the robustness of the coding scheme. Heavy traffic is set up in the motionless and motion period separately. The average packet loss percentage is 3.3% which is considered high for most networks. Fig. 16 show the images which suffer the packet loss from pass 4. As can be seen, the effect of lost packets is not at all severe, even if the lost packet rate is unrealistically high. This is because the performance from the first three pass is relatively good. Fig. 17 show the case when packet loss occurs in pass 1. Clearly there are visible defects in the motion period. What's worse is that the error will propagate to the following frames. Apparently the replenishing scheme used here is not sufficient in areas with motion. It is believed that this inconsistency can be eliminated with a motion compensator algorithm which would find the appropriate area for replenishment and error concealment which limits the propagation of error.

VII. Conclusions

The network simulator was used only as a channel in this simulation. In fact, before the real-time processor is built, a lot of statistics can be collected from the network simulator to improve upon the coding scheme. These include transmission delays and losses from various passes under different network loads. For resynchronization, the delay jitter between received packets can also be estimated from this simulation. The environment for tomorrow's telecommunication has been described and requires a flexibility which is not possible in a circuit switching network. With all the requirements

about applying packet video in mind, MBCPT has been investigated. It is found that MBCPT has appealing properties like high compression rate with good visual performance, robustness to packet lost, tractable integration with network mechanics and simplicity in parallel implementation. Some more considerations have been proposed for the whole packet video system like designing protocols, packetization, error recovery and resynchronization. For fast moving scenes, the differential MBCPT scheme seems insufficient. Motion compensation, error concealment or even attaching function commands into the coding scheme are believed to be useful tools to improve the performance and will be the direction of future research.

REFERENCES

- [1] W. Verbiest and L. Pinnoo, "A Variable Bit Rate Codec for Asynchronous Transfer Mode Networks," *IEEE J. Select. Areas Commun.*, vol.7, pp.801-806, June 1989.
- [2] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks," *IEEE J. Select. Areas Commun.*, vol.7, pp.801-806, June 1989.
- [3] J. Darragh and R. Baker, "Fixed Distortion Subband Coding of Images for Packet-Switched Networks," *IEEE J. Select. Areas Commun.*, vol.7, pp.801-806, June 1989.
- [4] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM networks," *IEEE J. Select. Areas Commun.*, vol.7, pp.801-806, June 1989.
- [5] G. Karlsson and M. Vetterli, "Packet Video and Its Integration into Network Architecture," *IEEE J. Select. Areas Commun.*, vol.7, pp.739-751, June 1989.
- [6] E. Daly and T. R. Hsing, "Variable Bit Rate Vector Quantization of Video Images for Packet-Switched Networks" in *Proc. ICASSP*, vol. 2, pp. 1160-1163, 1988.
- [7] S. S. Huang, "Source Modelling for Packet Video," *IEEE Int. Conf. on Commun.*, vol. 3, pp. 1262-1267, 1988.
- [8] S. L. Casner, D. C. Cohen and E. R. Cole, "Issues in Satellite Packet Video Communication," *IEEE Int. Conf. on Commun.*, vol. 1, pp. 34-38, 1988.
- [9] K. Sloan, Jr. and S. Tanimoto, "Progressive Refinement of Raster Scan Images," *IEEE Trans. Comput.*, vol. COM-28, pp. 871-874, Nov. 1979.
- [10] H. M. Dreizen, "Content-Driven Progressive Transmission of Grey-Scale Images," *IEEE Trans. Commun.*, vol. COM-35, pp. 289-296, March 1987.
- [11] D. Vaisey and A. Gersho, "Variable Block-Size Coding," *Proc. ICASSP*, pp. 1051-1054, April 1987.
- [12] Y. S. Ho and A. Gersho, "Variable-Rate Multi-Stage Vector Quantization for Image Coding" in *Proc. ICASSP*, vol. 2, pp. 1156-1159, 1988.
- [13] D. Nelson, K. Sayood, G. Ankenman and H. Chang, "Pnetsim System Programmer's Manual," University of Nebraska-Lincoln, Final Report for ARMY Contract DAAB07-85-K-K535, Dec. 1986.

