An Image Compression Technique for Use on Token Ring Networks

B. Gorjala and K. Sayood *

Department of Electrical Engineering

and

Center for Communication and Information Science

University of Nebraska - Lincoln

Lincoln, NE 68588-0511

Tel: (402) 472-6688

Fax: (402) 472-4732

email: ksayood@eecomm.unl.edu

G. Meempat

~~NEC~~ Bellcore

Trenton NJ

August 27, 1992

Abstract

We present a low complexity technique for compression of images for transmission over local area networks. The technique uses the synchronous traffic as a side channel for improving the performance of an ADPCM based coder.

*Supported by the NASA Goddard Space Flight Center(NAG 5-916 and NAG 5-1615).

1 Introduction

The use of local area networks makes it possible to more easily implement algorithms that require the use of a "side channel". In this paper we present an ADPCM (Adaptive Differential Pulse Code Modulation) based codec which can be conveniently implemented on LANs.

Adaptive Differential Pulse Code Modulation (ADPCM) is a very popular compression technique because it is easy to implement, has low processing overhead, and relatively good fidelity. This has made it the algorithm of choice in speech compression applications, and as a second stage for subband coding and transform coding techniques. However, ADPCM image compression is far from ideal. The most obvious drawback is poor edge reconstruction. ADPCM cannot track sudden changes in image statistics, and this can cause substantial edge distortion in the reconstructed image. A modified ADPCM scheme was presented in [1] which relied on the use of side information to prevent edge degradation. The technique is well suited for implementation on token ring networks.

In this paper we describe the implementation of this scheme in a token ring network environment. The paper is organized as follows. The next section gives a brief overview of the aspects of token ring networks that are of interest here. The modified ADPCM scheme is briefly described in the following section. Then we describe the implementation of the proposed algorithm on a token ring network and present simulation results.

2 Token Ring Networks

In a token ring network, nodes are arranged logically in a ring with each node transmitting to the next node around the ring. Each node simply relays the received bit stream from the previous node to the next node with at least one bit delay. The token is defined as a special bit pattern which circulates on the ring whenever all the stations are idle. Whatever node has the token is allowed to transmit a packet. When the packet has been transmitted the token is passed on to the next node. That is, whenever the node that is currently transmitting a packet finishes the transmission, it places the token, for example 01111110, at the end of the packet. When the next node reads this token, it simply passes the token if it has no packets to send. If it does have a packet to send it inverts the last token bit, in our example turning the token to 01111111. The station or node then breaks the interface connection and enters its own data onto the ring.

The token ring supports two classes of traffic;

1. Synchronous Traffic: A class of data transmission service whereby each requester is pre-allocated a maximum bandwidth and guaranteed a response time not to exceed a specific delay.

2. Asynchronous Traffic: A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated ring bandwidth and response time.

A set of timers and several parameters are used to limit the length of time a station may transmit messages before passing the token to the next station, and the duration of information transmission of each class within a station [2]. Each station maintains two timers, the Token_Rotation_Timer (TRT) and the Token_Holding_Timer (THT). The TRT at node j is used to time the interval taken by the token to circulate around the ring starting at node j. When node *j* recaptures the token, the value of TRT is assigned to THT and TRT is reset. When the network is initialized, the stations decide on the value of a target token rotation time (TTRT), so that the requirements for maximum access time are met. The upper bounds on the maximum and average token rotation time have been studied in [3]; the results show that the token rotation time cannot exceed twice the value of TTRT, while the average rotation time is not greater than TTRT. The extension to several priority classes is obtained by introducing a target rotation time for each class, and by using that value to check whether or not the station is allowed to transmit frames of that class.

If a station captures the token before its TRT reaches the value of TTRT, it is called an *early* token. If it captures the token after the TRT has exceeded the value of TTRT, it is called a *late* token. An *early* token may be used to transmit both synchronous and asynchronous traffic , while a *late* token may only be used for synchronous traffic. The difference between TTRT and TRT will be the available bandwidth for the asynchronous information. The amount of time a station can transmit is limited by THT.

In the following section we describe an image compression scheme which takes advantage of lighter loads on the network to provide side information to the receiver as asynchronous traffic. This side information is then used to increase the quality of the reconstructed image.

3 Edge Correcting DPCM

The proposed ADPCM system uses a two-bit Robust Jayant quantizer [4, 5]. This is a uniform quantizer whose step-size $\Delta(k)$ is adapted based on the previous quantizer output level $H(k-1)$ according to the following recursion [6]

$$\Delta(k) = [M(H(k-1))\Delta(k-1)]^{\beta}$$

where $\beta = 1 - \epsilon^2$, $\epsilon \leftarrow 0$, and $M(1) = 0.8, M(2) = 1.6$, $H(k) = 1$ if the output falls into the inner levels of the quantizer and $H(k) = 2$ if the output is one of the outer levels of the quantizer. As the information about which level of the quantizer was used in the previous sample, is available to both the transmitter and the receiver, the adaptation does not require the transmission of any side information.

The Jayant quantizer is designed to track the variance of the quantizer noise by changing the step size $\Delta(k)$. Since edges are regions where the statistics change rapidly, it follows that the step size will expand repeatedly when an edge is encountered. This fact is made use of in the following rule to detect edges:

> An edge is detected when the step size of the Jayant quantizer expands more than P times in succession, P ¿ 1.

The value of P should be small to reduce detection delay; a value of two seems to work well. As both transmitter and receiver have the same information both transmitter and receiver will detect edges at the same time. Once an edge has been detected the proposed scheme uses an embedded quantizer to quantize the quantization error and transmit this value over a side channel. The use of an embedded quantizer was first proposed by Goodman and Sundberg [7] for use over a noisy channel. In [1] the issue of how a side channel could be configured was left open. We address this issue in the context of token ring networks in the following section.

4 ADPCM and the Token Ring Network

As mentioned earlier, the traffic in the token ring network is divided into synchronous and asynchronous traffic. We use the regular ADPCM output as the synchronous traffic and the output of the embedded quantizer as the asynchronous traffic. Thus the side channel simply consists of the asynchronous traffic. The reasoning behind this approach is that the system cannot afford to lose the regular ADPCM output which also has timing constraints. The side information is not as critical, because the image can be reconstructed at the receiver without the side information, albeit with some degradation.

In the analysis of a token protocol, it is generally assumed that the queues of asynchronous messages to be sent are heavily loaded, so that messages are always available for transmission. In our case the asynchronous information queue will not be heavily loaded because the side information needs to be sent only when there is an edge.

The size of the packet for synchronous traffic is fixed. Whenever the node captures an early token, the size of the packet will be increased to match the available capacity and the regular information followed by the side information, if present, will be sent. The most recent side information will be transmitted in the bandwidth available for asynchronous traffic. If there is any side information left after transmission, it will be discarded.

Whenever the receiver receives an increased size packet it takes the bits received after the regular size of the packet as side information. This side information is added to the corresponding most recent "edge" pixels.

5 Simulation of Proposed Scheme

A fifty node token ring network was simulated to test the proposed system. The parameters used in the simulation are given in Table 1.

The system is assumed to work under the following general conditions

- The packet arrival process at each node follows a Poisson distribution.

Number of nodes	50
Bit traveling speed	200,000 met/msec
Distance between nodes	100 meters
Data generation rate	11,000 bits/sec
Packet size for synchronous information	1540 bits
Time taken by node to read the data	$10\mu sec$
Channel capacity of coaxial cable	12,000 bits/msec

Table 1: Simulator parameters

The actual image information is taken at node 1 with regular ADPCM output arriving into one buffer, and the side information into a separate buffer.

- The message transmitted transmitted by each station belong to two classes, i.e. asynchronous and synchronous messages.
- The access mechanism is based on the timed token approach, but different classes of asynchronous messages are not considered.
- The queues of asynchronous messages are not heavily loaded.
- When the network is initialized, the token rotation will only allow the transmission of synchronous messages; the second token rotation will allow both synchronous and asynchronous messages.

Two types of simulations were performed.

1. Messages transmitted at all nodes consisted of both synchronous and asynchronous messages.

2. Only synchronous messages were transmitted at all nodes.

Load versus delay and throughput versus delay characteristics were plotted for both cases and are shown in figures 1 and 2. Load is defined as the inverse of the mean inter-arrival time λ. The graph in Figure 1 shows that at a particular value of the load, the average delay of a packet in the network with both classes of traffic is more compared to when only synchronous messages are transmitted. This is especially true at low loads; as the traffic increases there is not much difference in the delay for the two cases. This is because the network will not have enough bandwidth available for asynchronous traffic when the traffic is busy.

The token ring network transmitting both synchronous and asynchronous messages provides better delay versus throughput characteristics. Here again at large values of throughput, there is not much difference between the curves. The reason for the better throughput versus delay characteristics is that at low loads, the network can utilize the channel more efficiently by transmitting asynchronous messages whenever the bandwidth becomes available.

The two images shown in Figure 3 were used to test the proposed approach and the results obtained at different network loads are shown in Table 3 and Table 2.

The first two entries in these tables were obtained by operating the net-

Run	Load	Delay msec	Throughput	Rate bpp	PSNR dB
1	.226	177.3	0.8033735	2.011	33.33
2	.185	147.8	0.8033297	2.014	33.36
3	.156	123.3	0.8032793	2.057	34.76
4	.136	105.3	0.8032306	2.126	35.69
5	.119	89.19	0.8030134	2.221	36.05
6	.107	73.17	0.8019965	2.223	37.22
7	.097	58.35	0.8000950	2.237	37.59
8	.085	42.82	0.7967703	2.238	37.62
9	.081	38.87	0.7956204	2.238	37.62

Table 2: Results obtained at different network loads for couple image

Run	Load	Delay msec	Throughput	Rate bpp	PSNR dB
1	.254	194.1	0.8033794	2.002	29.13
2	.169	135.3	0.8033345	2.016	29.22
3	.156	123.6	0.8033151	2.039	29.32
4	.145	114.4	0.8033025	2.075	29.78
5	.127	97.5	0.8032005	2.183	30.68
6	.107	70.3	0.8019068	2.227	30.90
7	.092	51.8	0.7989234	2.237	31.02
8	.081	38.4	0.7954938	2.237	31.02

Table 3: Results obtained at different network loads for aerial image

work at high loads which is in the unstable region. At these high loads almost every node will have a packet to send, and there was no bandwidth available for side information. As the load was decreased, more and more side information was transmitted, providing a better reconstructed image at the receiver. In this simulation, at a load of around 0.09, there is enough bandwidth available for node 1 to transmit all the side information. Further reduction of the load did not have any effect on the quality of the reconstructed image.

Error images for the couple image were obtained at four different network loads and are shown in Figure 4. The error image without any side information is shown in Figure 4a for comparison. For the image shown in Figure 3b, side information was sent in the areas of the woman's hands, the woman's left knee and in some portions of the couples heads. In figure 4c the edge errors are corrected in the region of the womans hands, the man's shoulder, the photo frame, and the couple's heads. Some of the edge errors at the man's legs are also corrected. But in this case edge errors are present at the woman's left knee. In the image shown in Figure 4d all edge errors are corrected except a few errors at the intersection of the man's leg and chair. For the image shown in Figure 4e all side information was transmitted.

6 Conclusion

We have presented a low complexity scheme which can be used for transmitting images over local area networks. Because of its low complexity the scheme can be operated at high rates and may be suitable for applications which require low delay.

References

[1] S.M. Schekall and K. Sayood. An Edge Preserving DPCM Scheme for Image Coding. In *Proceedings Midwest Symposium on Circuits and Systems*, pages 904–907. IEEE, 1988.

[2] A. Valenzano, P. Montuschi, and L. Ciminiera. On the Behavior of Control Token Protocols with Asynchronous and Synchronous Traffic. In *Proceedings IEEE INFOCOM*. IEEE, September 1989.

[3] K.C. Sevcik and M.J. Johnson. Cycle Time Properties of the FDDI Token Ring Protocol. *IEEE Transactions on Software Engineering*, SE-13:376–385, March 1987.

[4] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[5] N.S. Jayant. Adaptive Qunatization with One Word Memory. *Bell System Technical Journal*, pages 1119–1144, September 1973.

[6] D.J. Goodman and R.M. Wilkinson. A Robust Adaptive Quantizer. *IEEE Transactions on Communications*, pages 1362–1365, November 1975.

[7] D.J. Goodman and C.E. Sundberg. Combined Source and Channel Coding for Variable Bit-Rate Speech Transmission. *Bell System Technical Journal*, 62:2017–2036, September 1983.

PRECEDING PAGE BLANK NOT FILMED

LOAD VS. DELAY

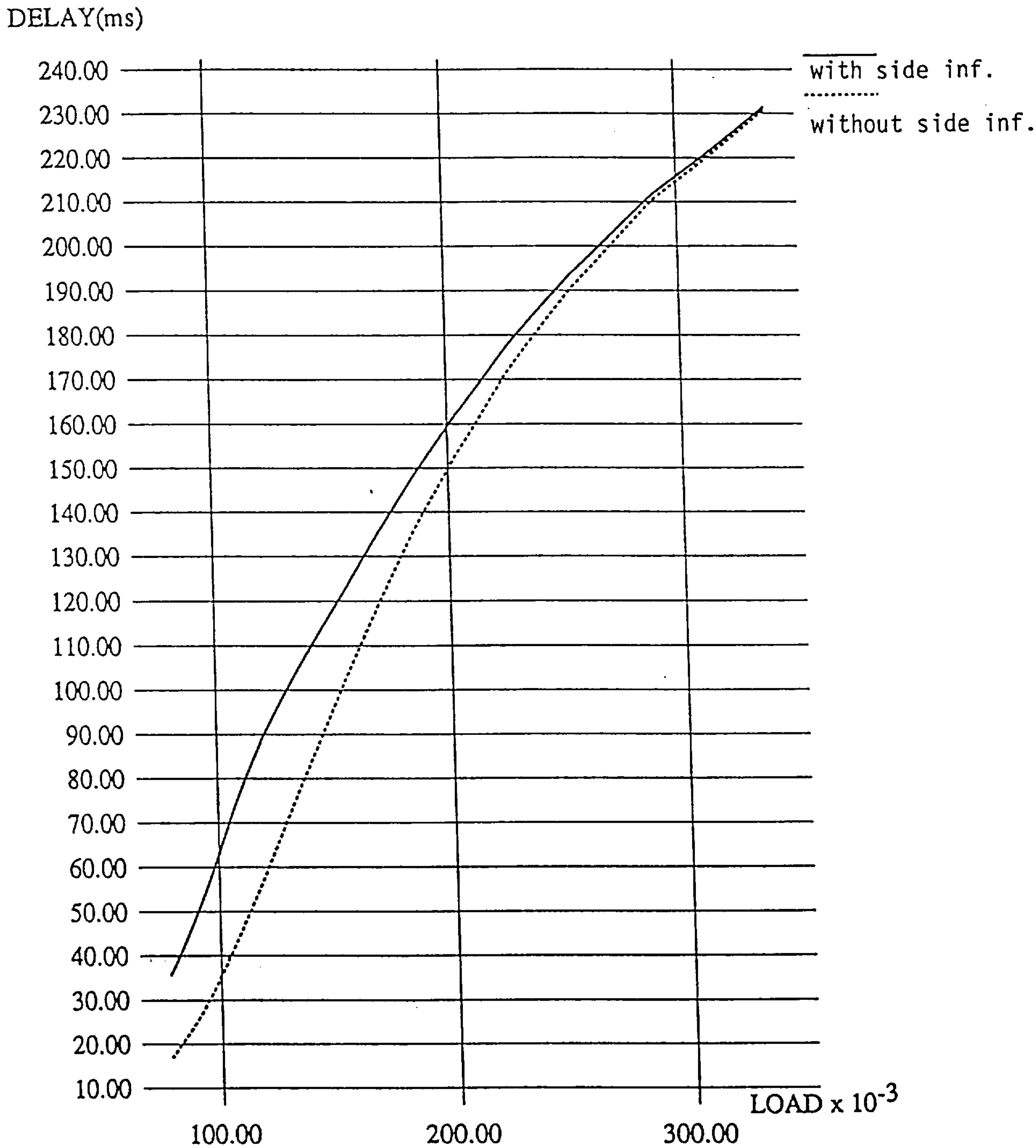

Figure 1 Load versus Delay for the Token Ring

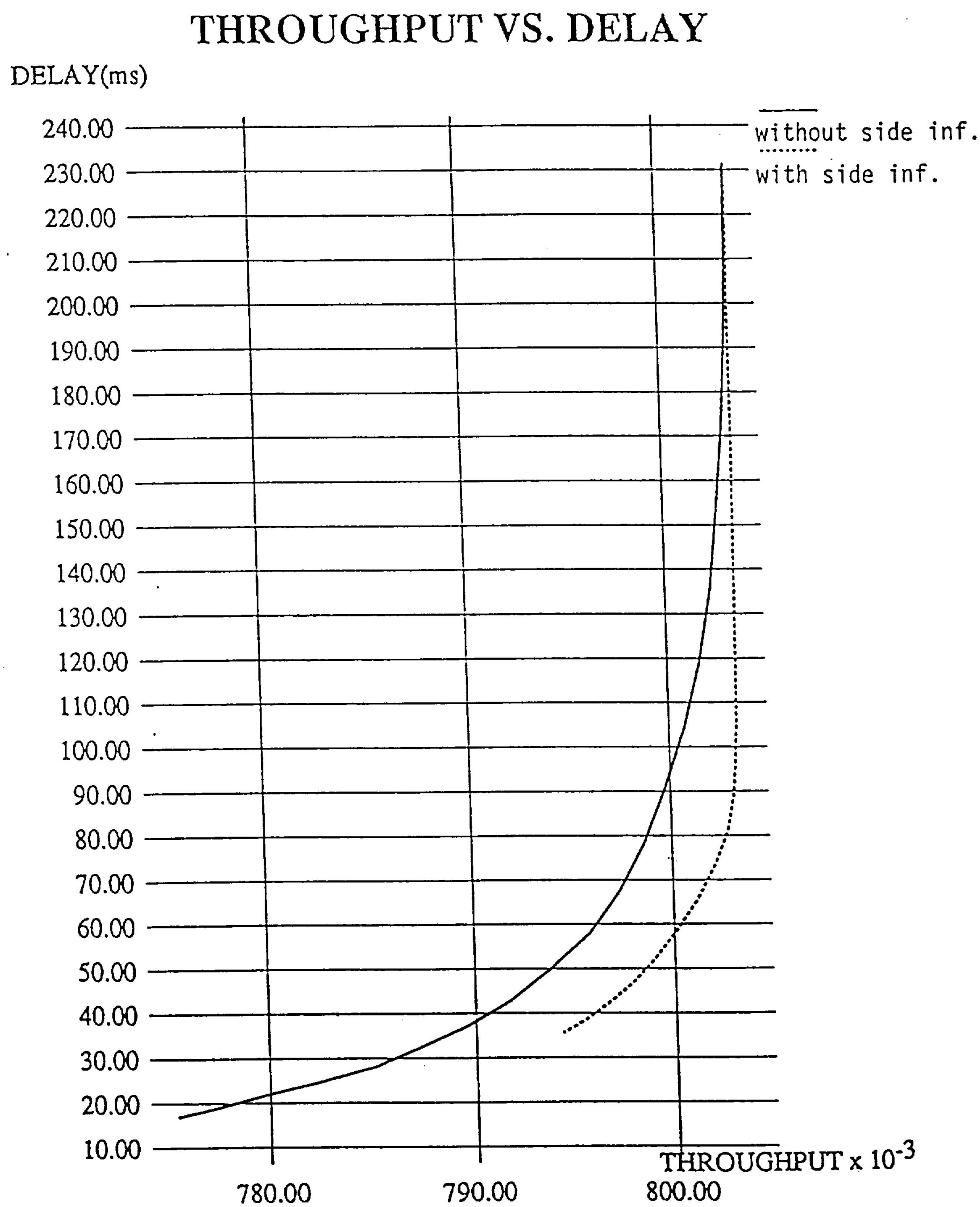

Figure 2 Throughput versus Delay for the Token Ring

C-2

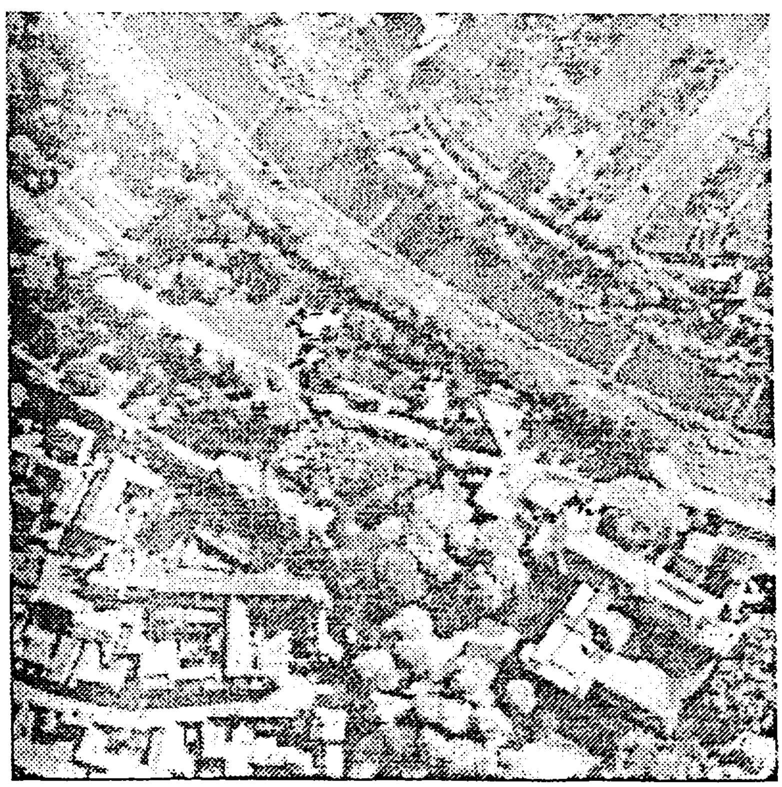

Figure 3. Images used in simulations

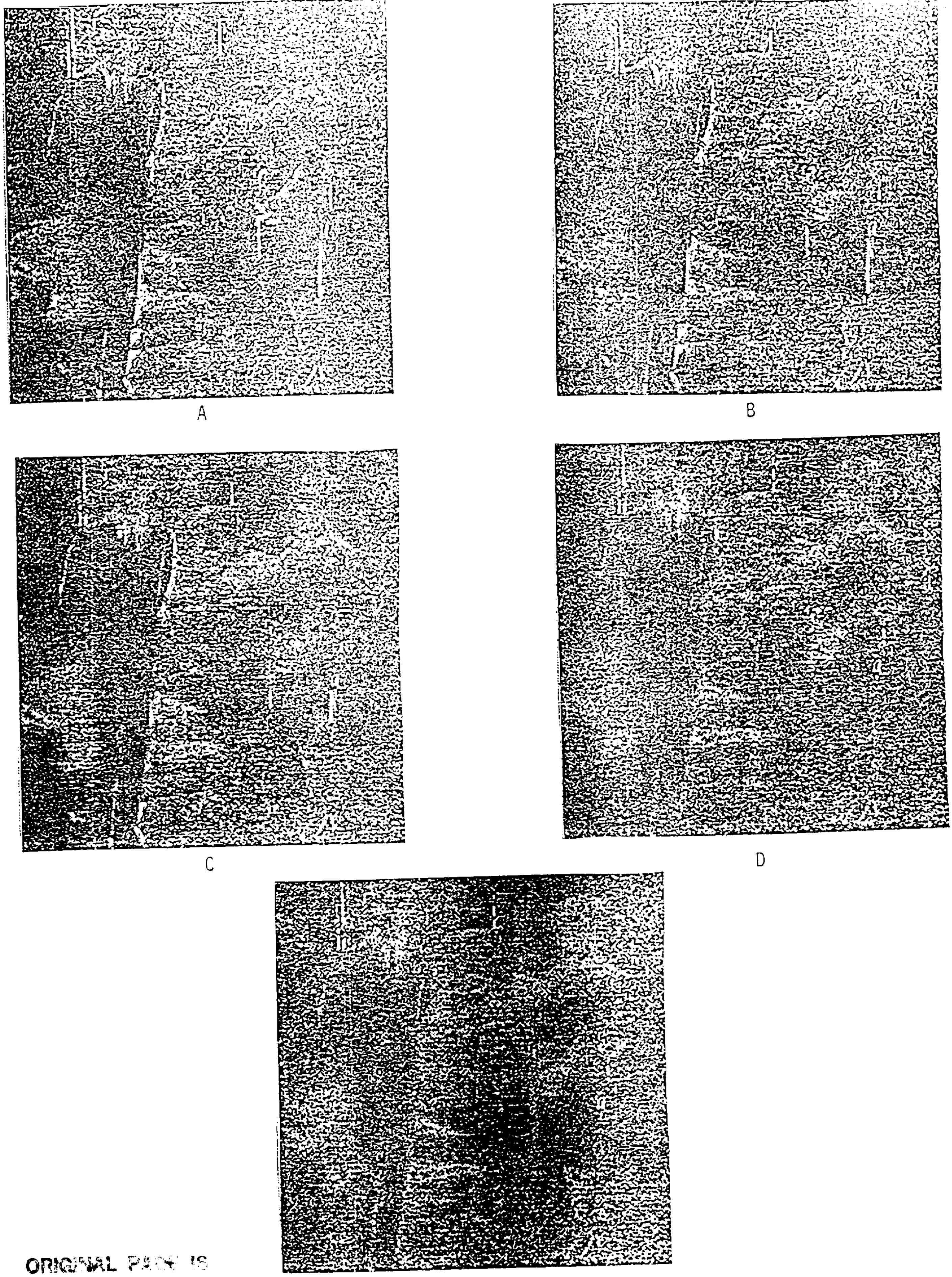

Figure 4. Error image for couple image at
a) rate=2.0, b) rate=2.08, c) rate=2.13, d) rate=2.23 e) rate=2.24

Figure 7. Lena image coded at two bits per pixel using DCT with fixed bit allocation

Figure 8. Lena image coded at (a) two bits per pixel and (b) one bit per pixel using JPEG algorithm

Figure 10. Lena image coded at (a) 2 bits per pixel and (b) 1.35 bits per pixel using EPDPCM