THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 55, No. 10, December 1976
Printed in U.S.A.

Ordering Techniques for Coding of Two-Tone Facsimile Pictures

By A. N. NETRAVALI, F. W. MOUNTS, and E. G. BOWEN

(Manuscript received June 23, 1976)

This paper describes some techniques for efficient coding of two-tone (black and white) facsimile pictures. These techniques use the two-dimensional correlation present in spatially close picture elements to change the relative order of transmission of elements in a scan line. This ordering increases the average length of the runs of consecutive black or white elements in the ordered line, making the data more amenable to one-dimensional run-length coding. We describe several variations of the ordering scheme, which differ in complexity and coding efficiency and evaluate their coding efficiency. For a variety of 8-1/2 inch by 11-inch typed documents, road maps, and circuit diagrams scanned with 200 lines/inch, these techniques reduce the bit rate by 30 to 50 percent over and above the one-dimensional run-length coding along a scan line; for single-spaced typed material with 100 lines/inch, this reduction is about 25 percent. We compare one of our techniques with a two-dimensional compression technique recently proposed by Preuss. We show that our technique results in an entropy about 10 to 18 percent lower than that obtainable through Preuss' technique.

I. INTRODUCTION

Transmission and/or storage of two-tone (black and white) pictures, such as weathermaps, printed texts, etc., have been receiving considerable attention for some time. The practical importance of this problem is evidenced by the number of facsimile communication systems that are now available.[1] As the cost of electronics decreases faster than transmission costs, it is becoming advantageous to use sophisticated facsimile terminals to reduce transmission costs and time, and, indeed, many of the recent facsimile communication systems have resorted to various source encoding techniques to utilize the statistical redundancy between the spatially close picture elements to reduce the bit rate required for transmission.[2–5]

The picture elements along a scan line of a facsimile picture consist of runs of white picture elements (pels) separated by runs of black picture elements. Values of the spatially close picture elements are significantly correlated. Source coding techniques, which do not reduce the "information content" of the pictures (i.e., it is possible to construct the original picture exactly without any degradation from the coded picture) use the statistical redundancy either along a single scan line or along many scan lines. One-dimensional run-length coding techniques[6] code the runs of black or white elements along a scan line. Development and performance of many different codes to code the runs have been a subject of many papers.[7–9] Some of these codes are capable of performing close to the entropy of the run-length statistics. Extensions of this basic run-length coding scheme have been made to include line-to-line correlations.[10–12] Two-dimensional correlations have also been used in development of block coding[13] as well as blob coding[14] methods.

In this paper, we describe techniques which use the two-dimensional correlation of the picture signals. Specifically, our techniques consist of changing the relative order of transmission of the picture elements along a scan line in such a way as to increase the average run length of the black and/or white elements. A reference signal is constructed from the previously transmitted data, and the data in the present line is ordered with respect to this reference signal. A memory is used to store the incoming bits of new data in a manner such that the address for storing a particular input bit is derived from the reference signal. At the end of the ordering period, the information stored in the memory is read out in a sequential manner and run-length coded. The receiver decodes the run-length coded information and stores it for a given ordering interval. The original data stream is then reconstructed by reading out the stored data in the same order in which it was stored at the transmitter. Our techniques can be classified into three broad categories. In the first category, described in Section 2.1, we order a line of picture data using the previously transmitted line immediately above it. Thus, the elements of the previous line are taken as the reference signal for ordering. In the second category, described in Section 2.2, we predict an element of a line from the value of a corresponding element from the previously transmitted line, and order the prediction error of the present line using elements of the previous line as the reference signal. The third category, which is described in Section 2.3, uses several already transmitted elements both from the present line and the previous line to define a state. We develop a predictor as a function of the state, as one which minimizes the prediction error conditioned on that state. We then sort the states in terms of probability of the prediction error associated with each state, and order the prediction error (using the state-dependent predictor) according to the *goodness* of the state. During ordering, prediction errors corresponding to "good" (states with low probability of prediction error) and "bad" states are arranged in a sequence in two different parts of a memory. Contents of the memory are then read out and run-length coded for transmission. Since prediction errors corresponding to "good" states are mostly 0, this technique increases the length of the 0 runs and consequently achieves bit-rate reduction.

1.1 Summary of results

Our simulations indicate that, for pictures with 200 lines/inch, the previous-line-ordering technique reduces the entropy by about 20 percent over the entropy using one-dimensional run-length coding. Using previous-line-element-prediction and previous-line-ordering, this reduction is about 30 percent. State-dependent prediction and ordering reduce the entropy by about 33 to 50 percent. This reduction is about 25 percent for a picture with 100 lines/inch. Our results indicate that our state-dependent predictor does not vary significantly with pictures and, therefore, may not have to be transmitted for most pictures. Also, most of the advantage in using ordering based on "good-bad" state-groups is obtained by using only two state-groups. Among the algorithms that we compare ours with, is an algorithm recently proposed by Preuss.[12] We show that our algorithm is about 10 to 18 percent more efficient in terms of entropy.

II. CODING ALGORITHMS

In this section, we describe each of our coding algorithms in detail and present results of our simulations on the computer. The computer simulations were done on pictures with 256 lines and 256 elements per line. The resolution was either 200 lines/inch or 100 lines/inch. The pictures we used included a drawing of a schematic, a map, and the inside part of text material (both single- and double-spaced typing). Sections of pictures used are shown in Fig. 1. Figures 1a and 1b are sections of single-spaced text with 200 lines/inch and 100 lines/inch, respectively. Figure 1c is double-spaced text with 200 lines/inch and Figure 1d is part of a circuit diagram. In addition to these, we used a map which is a section of page 19 from Ref. 15. As a measure of performance, we used the entropy of run-length statistics. We computed the entropy of black and white runs and the average black and white run lengths. Using these and eq. (1), we computed the entropy in bits/pel (assuming that the number of black and white runs are

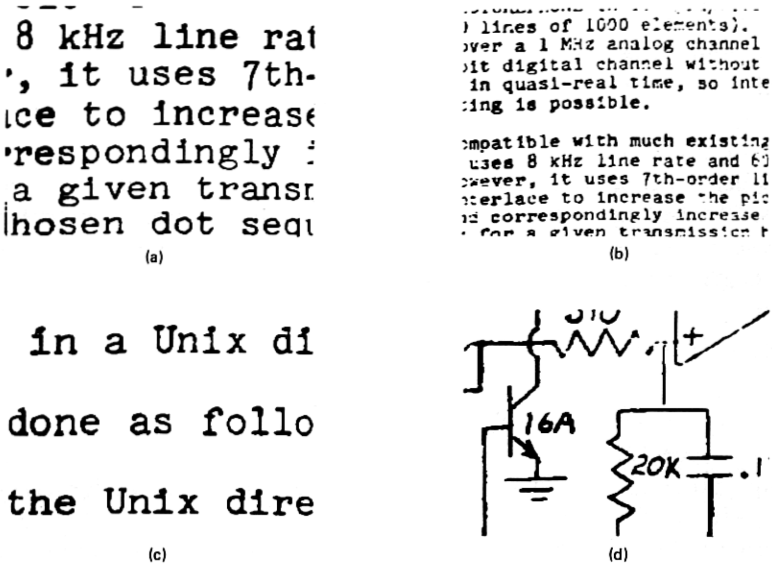


Fig. 1—Sections of figures viewed through a television camera and converted to a two-tone picture by a simple thresholding technique. Figures (a), (c), and (d) had 200 lines/inch and an array of 256 × 256 was used for simulation. Figure (b) was taken at 100 lines/inch.

equal to $N/2$) by:

$$H = \frac{H_w \cdot \frac{N}{2} + H_b \cdot \frac{N}{2}}{r_w \cdot \frac{N}{2} + r_b \cdot \frac{N}{2}} = \frac{H_w + H_b}{r_w + r_b}, \tag{1}$$

where

H_w is the entropy of the white run statistic (bits/run),
H_b is the entropy of the black run statistic (bits/run),
r_w is the average white run length (pels/run),
r_b is the average black run length (pels/run),
H is the entropy in bits/pel.

2.1 Ordering present line with reference to previous line

This technique orders the present line with respect to the previous line. Consider a memory containing 256 cells (equal to the number of elements per line). We store in this memory elements from the present line. Assume for the sake of explanation that the memory is arranged along a line and memory location 1 corresponds to the left-hand side and location 256 corresponds to the right-hand side of the memory. If the first element of the previous line is white (=1), then we put the content of the first element of the present line on the left-hand side of the memory. If, on the other hand, the previous line element is black (=0), we put the first element of the present line on the right-hand side of the memory. We then put the second element on the right-hand or left-hand side of the memory depending on whether the second element of the previous line is black or white. This process is continued until the entire present line is ordered and the memory is filled. The information stored in the memory is coded as runs of black and white elements. It is easy to see that the present line can be uniquely reconstructed from the knowledge of the run lengths of the ordered line since the ordering information is known to the receiver.

The entropies obtained using the ordered run-length statistics are given in Table I. This table also shows, for comparison purposes, the entropy of the picture using the statistics of simple one-dimensional (along a scan line) unordered run lengths. The ordered entropy of the run lengths varies between 0.12 bit/pel to 0.24 bit/pel for 200 lines/inch resolution pictures. The increase in coding efficiency (=decrease in entropy) due to ordering over plain run-length coding is of the order of 20 to 25 percent. This increase in efficiency is decreased to 16 percent for the picture with 100 lines/inch. In Fig. 2, we show the original picture (same as Fig. 1a) and its ordered version. It is interesting to note that the picture elements on the left side of the ordered picture are mostly white and those on the right side of the picture are

Table I—Entropy comparisons for different coding algorithms

Coding Technique	Entropy (bits/pel) Picture				
	1	2	3	4	1 at 100 lines/in.
(1) One-dimensional run-length coding	0.30	0.16	0.21	0.23	0.38
(2) Present line ordered with reference to previous line	0.24	0.12	0.16	0.17	0.32
(3) Finite length ordering of Technique (2) (length = 64)	0.29	0.17	—	—	—
(length = 128)	0.28	0.15	—	—	—
(4) Run-length coding of prediction error using previous line predictor	0.25	0.13	0.15	0.16	0.35
(5) Technique (4) with ordering using previous line	0.21	0.11	0.14	0.15	0.31
(6) Run-length coding of prediction error using state-predictor	0.24	0.13	0.15	0.16	0.34
(7) Technique (6) with state-ordering using 2 state-groups	0.20	0.10	0.11	0.12	0.30
4 state-groups*	0.20	0.10	0.11	0.12	0.30
16 state-groups*	0.19	0.099	0.11	0.119	0.29
Technique of Preuss using 2 state-groups*	0.22	0.11	0.13	0.14	0.34

* These entropy numbers do not include extra bits required to specify number of elements in each state-group.

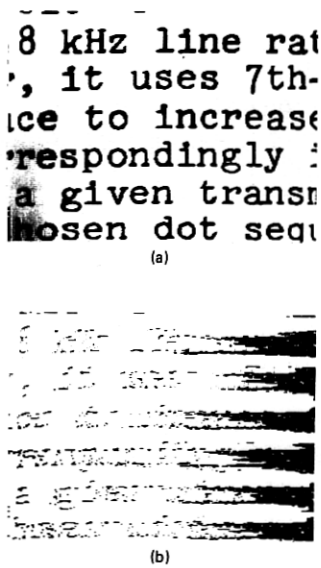


Fig. 2—(a) Picture of original. (b) Present line ordered with respect to previous line.

mostly black. If the vertical correlation between the scan lines was perfect, there would be no scattered black and white elements; all the white elements would be to the left and all the black elements would be to the right. The increase of coding efficiency is intuitively obvious by comparing both the original and the ordered pictures of Fig. 2.

2.1.1 Finite length ordering

To evaluate the effects on coding efficiency of ordering only a part of the line, we simulated finite length ordering. This has the advantage to some extent of localizing along the horizontal direction the effect of transmission errors. However, vertical propagation of transmission errors is still possible. To illustrate this scheme, consider two memories of 128 cells each (half the number of samples/line). We then order the elements of the present line as before for each half of the line. However, in the first memory, we put elements corresponding to black elements of previous line to the left side and elements corresponding to the white elements of the previous line to the right side; whereas, for the second memory we reverse the sides for the black and white elements, i.e., elements of the present line corresponding to the black elements of the previous line are put on the left side. This minimizes the effects of "discontinuity" at the boundary of the two memories. The two memories are now arranged back to back, and their contents are read out sequentially and are run-length coded. The results of simulating this scheme are shown in Table I for different sizes of the memories.

The increase in the entropy by dividing the line into two segments is significant. Thus, there is a considerable loss in coding efficiency due to finite length ordering. This allows us to conclude that by ordering two lines instead of parts of a line and arranging them back to back, as above, there may be further improvement in coding efficiency.

2.2 Ordering present line prediction errors with reference to previous line

This coding technique is similar to the one described in Section 2.1, except that now we order the prediction errors of the present line. We take the predictor to be the corresponding picture element in the previous line. We now order this prediction error as before; i.e., if the previous line has a white element, we put the prediction error of the present line to the left side of the memory and vice versa. As in the previous technique, from the ordered line, it is easy to decode uniquely the contents of the present line.

The entropy of the pictures using the run lengths of the ordered prediction error is between 0.11 bit/pel and 0.21 bit/pel for pictures with 200 lines/inch, as seen from Table I, and this amounts to a 30 to 35 percent decrease in entropy over simple one-dimensional run-length coding. To calculate the advantage of ordering, we also measured the entropy of the picture using the run-length statistics of the prediction error with the previous line element as predictor. The entropy of this, which is shown in Table I, varied between 0.13 bit/pel to 0.25 bit/pel, for pictures with 200 lines/inch. Thus, the reduction in entropy due to ordering was 7 to 16 percent over and above the entropy of the prediction errors. For the picture with 100 lines/inch, ordering the prediction errors brought the entropy down to 0.31 bit/pel, which is a reduction of about 17 percent over the entropy of one-dimensional run-length coding.

2.3 State-dependent prediction and ordering

The technique described in this section differs from the two earlier techniques in its use of more picture elements spatially close to the present element. It uses these elements to define a state of the present picture element. We develop a predictor for each state and use the state also to order the present line. Using this technique, it is possible to *separate* the process of prediction and ordering.

To illustrate the technique, consider the picture element configuration shown in Fig. 3. The state of the present picture element X is defined by elements A, B, C, and D. Thus, state Z is the four-tuple

$$Z = (A, B, C, D). \tag{2}$$

Since each of the elements A, B, C, and D can have two possible values,

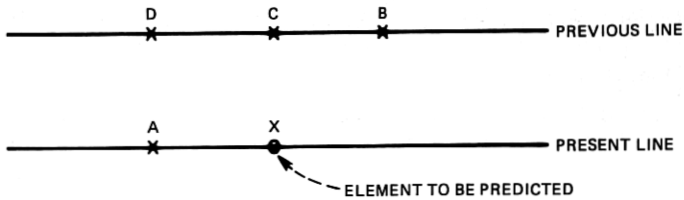


Fig. 3—Configuration for state definition.

there are 16 states, which we denote by the set $\{Z_i\}$, $i = 1, \cdots, 16$. The development of a state-dependent predictor is our next task. Such a development is as old as predictive coding itself.[12,16] The predictor is developed by the following criterion. We first compute $P(X = \text{'Black'} | Z = Z_i)$, the probability of the present picture element X being Black given the state $Z = Z_i$. The predictor $C(Z_i)$ for a given state Z_i, is then,

$$\begin{aligned} C(Z_i) &= \text{'Black'} \quad \text{if} \quad P(X = \text{'Black'} | Z = Z_i) > 0.5 \\ &= \text{'White'} \quad \text{otherwise.} \end{aligned} \tag{3}$$

It is easy to see that this predictor minimizes the probability of making an error given that a particular state has occurred. We have calculated the predictor for each state using several pictures. These are shown in Table II. For most states, the predictors do not depend upon the picture used, except for a few states that are marked with asterisks in Table II; thus, it is not necessary to transmit the predictors for each different picture. We shall evaluate the effects of using the predictors of one picture for other pictures.

Having developed the predictor, we sort the states into two groups. The probability of correct prediction using the state-dependent predictor is shown in Table II. We note that the probability of correct prediction is always higher than 0.5 due to our predictor being a minimum prediction error predictor. States which have high probability of correct prediction will be called "good" states. Our ordering strategy depends upon the goodness of the state. Let the 16 states be divided into two groups: one containing "good" states and one containing "bad" states. Our state-dependent ordering algorithm then works as follows: we first evaluate the prediction error for a particular picture element in the present line by using the state-dependent predictor. Then, if the state is a "good" state, we put the prediction error on the left side of the memory, and if the state is "bad", we put the prediction error on the right side of the memory. In all our simulations, we used a threshold of 0.8 for the probability of prediction error to determine the goodness of a state. Having ordered the prediction error, we then code the run lengths of the prediction error. It is easy to see that the line of data can be uniquely constructed from the coded run lengths of the prediction error. The entropy of run lengths of such ordered prediction errors for different pictures is given in Table I. For pictures with 200 lines/inch resolution, the entropy varies between 0.10 bit/pel to 0.20 bit/pel. This represents a decrease of between 33 percent and 49 percent over the entropy of simple one-dimensional run-length coding. We evaluated, for comparison purposes, the entropy of the picture using the run-length statistics of the prediction error. This varies between 0.13 bit/pel to 0.24 bit/pel for the pictures with 200 lines/inch. It is clear then that state-dependent ordering allows us to decrease the entropy by about 16 to 25 percent over and above the entropy obtained by using the prediction error of the state-dependent predictor.

We described a scheme in which only two groups of states were used for ordering purposes. To evaluate the effect of using more than two groups, we divided the states into 16 groups, and ordered the prediction error as before. In the case of more than two groups of states, it is not possible to decode the original line of picture data from coded run lengths unless extra information about the number of elements in each state-group is specified for each line. The entropy of the run-length statistic using more than two state-dependent groups is shown in Table I. These figures of entropy do not include the extra information that is required to be transmitted about the number of elements in the group. The increase of coding efficiency by using more than two groups is somewhat small and it would be offset completely by the extra information mentioned. Thus, most of the decrease of entropy due to state-dependent ordering is obtained by using only two groups.

In all of the state-dependent coding algorithms, we have not made an effort to optimize several of the coder parameters. Thus, for example, when the state-groups are less than 16, we could optimize the groupings of the states. The groupings that we used were intuitive and somewhat ad hoc. We did vary the groupings in the case of two groups and found that the entropies did not change significantly. It appears that optimization of groupings may not result in any significant entropy reduction.

2.3.1 Sensitivity to picture variation

Picture content affects three parameters of our state-dependent coding algorithms: the state-dependent predictor, the entropy numbers which depend on the statistics of the run lengths, and the definition of "good" and "bad" (or the groupings) states. We studied the sensitivity of our coding algorithms by considering the variation of predic-

Table II — State-dependent predictors

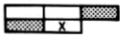
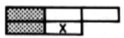
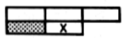
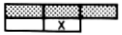
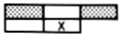
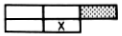
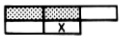
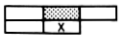
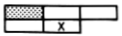
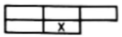
State No.	State Definition	Picture 1 Single-Space Type			Picture 2 Double-Space Type			Picture 3 Circuit Diagram			Picture 4 Road Map			Picture 1 Single-Space Type (100 Lines/In)			Weather Map (Preuss)	
		Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.
1		B	0.96	0.0553	B	0.97	0.0354	B	0.97	0.0366	B	0.97	0.0466	B	0.84	0.0229	B	0.911
2		B	1.00	0.0076	B	1.00	0.0040	B	1.00	0.0043	B	0.96	0.0035	B	0.90	0.0079	B	0.991
3		B	1.00	0.0001	B	0.83	0.0001	B	1.00	0.00003	W*	0.70	0.0014	B	0.51	0.0008	B	0.663
4		B	0.99	0.0033	B	1.00	0.0017	B	0.99	0.0013	B	0.85	0.0013	B	0.99	0.0049	B	0.952
5		B	0.83	0.0291	B	0.86	0.0156	B	0.80	0.0182	B	0.81	0.0152	B	0.68	0.0207	B	0.768
6		B	1.00	0.0008	B	1.00	0.0002	B	0.96	0.0009	B	0.82	0.0014	B	0.87	0.0054	B	0.920
7		W	0.74	0.0266	W	0.76	0.0137	W	0.81	0.0155	W	0.81	0.0175	W	0.74	0.0366	W	0.553
8		B	0.55	0.0169	B	0.54	0.0084	B	0.61	0.0086	B	0.58	0.0089	W*	0.55	0.0259	B	0.501

Table II — Continued

State No.	State Definition	Picture 1 Single-Space Type			Picture 2 Double-Space Type			Picture 3 Circuit Diagram			Picture 4 Road Map			Picture 1 Single-Space Type (100 Lines/In)			Weather Map (Preuss)	
		Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.	Prob. of State	Pred.	Prob. of Correct Pred.
9		W	0.63	0.0169	W	0.60	0.0084	W	0.69	0.0075	W	0.70	0.0071	W	0.71	0.0160	W	0.616
10		B	0.71	0.0260	B	0.72	0.0131	B	0.80	0.0153	B	0.78	0.0145	B	0.52	0.0244	B	0.532
11		W	1.00	0.0003	W	1.00	0.0002	W	1.00	0.00008	W	0.88	0.0004	W	0.97	0.0014	W	0.912
12		W	0.85	0.0320	W	0.84	0.0156	W	0.82	0.0191	W	0.84	0.0208	W	0.83	0.0472	W	0.810
13		W	0.84	0.0035	W	0.97	0.0014	W	0.87	0.0013	B*	0.53	0.0027	W	0.60	0.0114	B	0.609
14		B	0.86	0.0015	B	0.75	0.0001	B	0.67	0.0005	B	0.88	0.0045	B	0.85	0.0166	B	0.885
15		W	1.00	0.0078	W	1.00	0.0034	W	1.00	0.0049	W	0.99	0.0045	W	0.94	0.0151	W	0.988
16		W	0.99	0.7722	W	1.00	0.8789	W	1.00	0.8664	W	1.00	0.8500	W	0.99	0.7429	W	0.996

* Varied with pictures.

 = Black = B.

 = White = W.

tors. The second factor mentioned could be studied by developing a specific code based on some run-length statistics (of one of the pictures, or some "average" picture) and then using it on all the pictures. We did not study this aspect of sensitivity of our algorithm. As mentioned in the previous section, we studied the variation of the entropy with groupings and found that the variation was not too sensitive. Thus, it appears that the groupings-information need not be computed or transmitted for each picture.

From Table II, it is clear that for a resolution of 200 lines/inch, the predictors are identical for all states for pictures of single-spaced typing, double-spaced typing, and schematic. This may be a result of our using a predictor based on the local information surrounding the picture element. In the case of a map, however, there are two states that have a different predictor compared to the first three pictures. Both these states were regarded as "bad" states for the coding of the map. We used the predictor of the first three pictures for the coding of the map; and using two state groups, we found a 3-percent increase in the entropy. Also, in the case of the picture with 100 lines/inch, there is only one state (state number 8) which had a different predictor than the first three pictures. This was again a "bad" state. For this picture, we found the increase in entropy of about 2 percent. This allows us to conclude that it may not be necessary to compute and transmit the state-dependent predictor information for each picture.

2.4 Comparison with the algorithm of Preuss

Most coding algorithms perform differently for different pictures. To compare our results with other two-dimensional coding techniques, we implemented a coding algorithm proposed by Preuss.[7] This also allowed us to bring out certain similarities and dissimilarities between our algorithm and that of Preuss. Preuss has developed a state-dependent predictor analogous to our predictor. This predictor is shown in Table II. It is seen that his predictor differs from our predictor for the first three pictures only for state number 13. Also, it differs from our predictor for the first picture (100 lines/inch) for state numbers 8 and 13. In our simulation of Preuss' scheme, we used the predictor tuned to the particular picture rather than Preuss' predictor. Preuss computes the prediction errors analogous to our scheme, and then codes the run lengths between the prediction errors for each of the state-groups separately, using a different run-length code for each state-group. We, on the other hand, use the state-groups to order the present line and encode the run lengths of the entire ordered line of the prediction errors. Preuss has to specify the number of elements in each state-group;* for K state groups with N elements in each line, this may amount to $(K - 1) \log_2 N$ extra bits/pel. In our algorithm, we do not need transmission of such information for $K = 2$. In Preuss' scheme, the run lengths have to be terminated at the end of each state-group for each line, but in our algorithm, a run may begin in one state-group and extend all the way up to the end of the line, crossing several state-groups. Despite these disadvantages, we thought that Preuss' scheme may result in lower entropy, since his run-length code was matched to the run-length statistics of the prediction errors corresponding to each state-group. We simulated Preuss' scheme using two state-groups that were the same as those used for our algorithm. Results of this simulation are given in Table I. Entropy numbers given for Preuss' scheme do not include the extra information required for the number of elements in each state-group. It is seen from this table that, compared to our scheme using two state-groups, Preuss' scheme results in a 10- to 18-percent increase in entropy. Thus, our scheme appears, at least for the pictures we used, to be more efficient.

III. DISCUSSION AND SUMMARY

We have presented three different algorithms for the coding of two-tone pictures. All three algorithms are "information" preserving, and, therefore, it is possible to decode exactly the original picture with no approximations. We have compared our results (only in terms of entropy) with some standard algorithms such as: (*i*) one-dimensional run-length coding, (*ii*) run-length coding of the prediction errors using several different two-dimensional predictors, and (*iii*) a two-dimensional algorithm of Preuss. We found our algorithm to be 10 to 18 percent more efficient than the algorithm of Preuss. Admittedly, this is not a complete comparison. Other parameters, such as the number of samples per line (we used only 256) and varying picture material, may offset the comparisons. Also, we did not study many other aspects important for a coding system, including the performance in the presence of transmission channel errors.

Our technique can be extended by proper definition of the state to the case of two-tone dithered pictures. This will be reported in a future paper.[17]

IV. ACKNOWLEDGMENTS

This work was performed on the picture processing facility developed mainly by J. D. Beyer and R. C. Brainard. Also, our work

---

* In a private communication, Preuss has shown that his scheme can be modified so that the number of elements in each of the state-groups need not be transmitted.

was made easier by many of the general purpose programs developed by B. G. Haskell.

REFERENCES

1. D. M. Costigan, "FAX: The Principles and Practice of Facsimile Communication," New York: Chilton, 1971.
2. D. Hochman and D. R. Weber, "DACOM Facsimile Data Compression Techniques," 1970 Int. Conf. on Communications, pp. 20.14–20.21.
3. B. M. Rosenheck, "FASTFAX, A second Generation Facsimile System Employing Redundancy Reduction Techniques," IEEE Trans. Commun. Tech., *COM-8* (December 1970), pp. 772–779.
4. C. G. Beaudette, "An Efficient Facsimile System for Weather Graphics," in *Picture Bandwidth Compression*, New York: Gordon Breach, 1971, pp. 217–229.
5. R. B. Arps, "Bibliography on Digital Graphic Image Compression and Quality," IEEE Trans. Inform. Theory, Vol. 20 (January 1974), pp. 120–122.
6. J. Capon, "A Probabilistic Model for Run-Length Coding of Pictures," IRE Trans. Inform. Theory, *IT-5* (1959), pp. 157–163.
7. D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proc. IRE, *40* (September 1952), pp. 1098–1101.
8. W. W. Happ, "Coding Schemes for Run-Length Information Based on Poisson Distribution," National Telemetering Conference Record, April 1969, pp. 257–261.
9. H. Meyer, H. G. Rosdolsky, and T. S. Huang, "Optimum Run Length Codes," IEEE Trans. Commun., *COM-22* (June 1974), pp. 826–835.
10. T. S. Huang, "Run-Length Coding and Its Extensions," in *Picture Bandwidth Compression*, New York: Gordon Breach, 1972, pp. 231–266.
11. P. A. Stern and W. Heinlein, "Facsimile Coding Using Two-Dimensional Run Length Predictions," International Seminar on Digital Communications, Zurich, 1974, pp. 170–176.
12. D. Preuss, "Comparison of Two-Dimensional Facsimile Coding Schemes," International Conference on Communications, June 1975, pp. 7.2–7.16.
13. M. Kunt, "Comparison de Techniques d'Encodage Pour la Reduction de Redondance d'Images Fac-simile a deux Niveaux," Thesis, Ecole Polytechnique Federale de Lausanne, 1974.
14. A. J. Frank, "High Fidelity Encoding of Two-Level, High Resolution Images," International Conference on Communications," June 1973, pp. 26-5–26-1.
15. State Farm Road Atlas: United States, Canada, Mexico, New York: Rand McNally, 1965.
16. J. S. Wholey, "The Coding of Pictorial Data," IRE Trans. Inform. Theory, *IT-7* (April 1961), pp. 99–104.
17. A. N. Netravali, F. W. Mounts, and J. D. Beyer, "Techniques for Compression of Dithered Two Level Pictures," unpublished work.