THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

CCITT Compatible Coding of Multilevel Pictures

By H. GHARAVI* and A. N. NETRAVALI*

(Manuscript received February 25, 1983)

The Comite Consultatif International Telegraphique et Telephonique (CCITT) has recently recommended a code for two-level (black and white) graphics transmission. A large number of pictures in graphics communication contain areas that cannot be represented adequately by only two shades of gray. We describe techniques by which a composite picture, containing an arbitrary mixture of two- and multilevel areas, can be coded by schemes that are compatible with the CCITT code. First, the composite picture is segmented automatically into two types of areas: one requiring only two levels (text, drawings, etc.) and the other requiring multilevels (for example, photos). A Differential Pulse Code Modulation (DPCM) scheme is then used to code the multilevel areas. Code assignment for the outputs of the DPCM quantizer are based on the local conditional statistics, and the bit stream is processed to change the statistics of the run lengths so that the CCITT run-length code becomes efficient. Results of computer simulations are presented in terms of quality of processed pictures and the required bit rate. Simulations show that our CCITT compatible scheme is as efficient as an incompatible but optimum DPCM coding scheme.

I. INTRODUCTION

Simultaneous developments (algorithmic as well as systems) have taken place for many years in coding and transmission of two-level (black and white) document facsimile, and multilevel (many shades of gray) pictures.[1,2] The former type of pictures require very high spatial resolution to preserve the sharpness and have been coded by one-

* Bell Laboratories.

dimensional run-length coding and two-dimensional edge difference coding [Comite Consultatif International Telegraphique et Telephonique (CCITT) one- and two-dimensional codes].[3] On the other hand, multilevel pictures contain gradual luminance transitions, and therefore require lower spatial resolution. They have been coded by Differential Pulse Code Modulation (DPCM) and transform methods. Most pictures used in business facsimile systems and audiographics conferencing contain a mixture of two-level and multilevel segments or subpictures. Coding such pictures using two-level techniques would not be adequate from the point of view of the picture quality, and using multilevel techniques would generate enormous data rates. Thus, it is of interest to devise schemes that automatically divide a picture into segments, each segment with a specified amplitude (gray shades) and spatial resolution and code each segment as best suited for it. Another practical requirement is that of compatibility. A coding scheme that handles a mixture of two-level and multilevel segments should be upwardly compatible with the CCITT standard schemes for two-level pictures. System cost will be reduced if the scheme for two-level multilevel pictures uses hardware blocks that are also used by the two-level picture coder. We present such a scheme below. Principal characteristics of our scheme are:

1. Compatibility with the CCITT schemes for two-level pictures
2. Automatic segmentation of pictures into two-level and multilevel segments
3. High coding efficiency by preprocessing the multilevel segments to fit the CCITT codes
4. Lower spatial resolution for gray-level segments (if desired)
5. Nonlossy (information preserving) coding of two-level segments, and lossy coding of multilevel segments.

II. CODING ALGORITHM

The coding algorithm is explained in the following steps.

2.1 Segmentation

The function of the segmentor is to classify each picture element as one of the three:

1. Black } → Two-Level
2. White }

3. Gray → Multilevel

Figure 1 shows a neighborhood of the current picture element (pel) used for segmentation. We assume that each pel, obtained from the scanner, is specified by many shades of gray (e.g., 8 bits). The size of the neighborhood can be arbitrary. If it is too small, then many

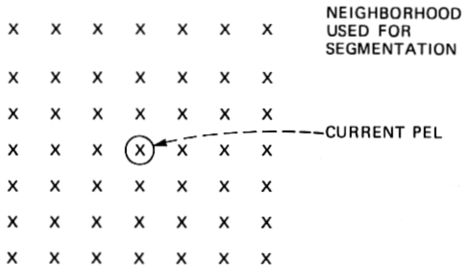

Fig. 1—Picture elements used for segmentation of the current pel. The size of the neighborhood is not necessarily 5 × 5 as suggested in the figure.

discontinuous segments of gray pixels will be generated. On the other hand, if the neighborhood is too large, then the ability to resolve small gray areas is lost. We slide this window of neighborhood over the pels along a scan line and classify each pel. We consider the boundary pels of the picture separately. Two thresholds, t_1 and t_2 ($t_1 < t_2$), are selected. It is hypothesized that most black pels will have intensity less than t_1, white pels will have intensity greater than t_2, and gray pel intensities may lie anywhere. Within the neighborhood let

n_1 = number of pels with intensity value $< t_1$

n_2 = number of pels with intensity value $> t_2$

n_3 = all the rest of the pels.

We define a state, S, consisting of three components: S_1, S_2, and S_3. A picture is segmented on the basis of the value of S. Let

$$S = S_1 + S_2 + S_3, \tag{1}$$

where

$S_1 = 1$, if $n_3 > n_1 + n_2$

$= 0$, otherwise

$S_2 = 1$, if previous pel is gray

$= 0$, otherwise

and

$S_3 = 1$, if $t_1 <$ intensity of present pel $< t_2$

$= 0$, otherwise.

The segmentation rule is then given by

$S \geq 2 => $ current pel gray

$S < 2$ and intensity of current pel $> T =>$ current pel white

$S < 2$ and intensity of current pel $\leq T =>$ current pel black.

T is a threshold used to distinguish black elements from white ones, once they are known to be of the two-level type. If the range $(t_2 - t_1)$ is decreased by increasing t_1 and decreasing t_2, then more elements will be regarded as two-level and the quality of picture may suffer, but this will also decrease the bit rate.

We evaluated the performance of the segmentor, in particular its dependence on the block size and $(t_2 - t_1)$ by computer simulation. Since there are no standard mixtures of two-level and multilevel images, we created our own by taking a 512 × 512 gray-level image (shown in Fig. 2) and superimposing it on the CCITT documents four and five. Since this 512 × 512 original was scanned at low resolution (compared to 200 pels/inch used for CCITT documents), it contains significant sharp transitions that would not be present in a photograph scanned at 200 pels/inch. Also, because the original gray-level picture and the CCITT documents are rather "clean", segmentor works quite

Fig. 2—A 512 × 512 multilevel (8 bits/pel) picture used for simulation.

well. However, this may not be a typical situation if a nonideal scanner was used. We, therefore, added random noise to the entire composite picture. This noise had a variance of 425 (on an 8-bit scale, 0-255). Table I shows the performance of the segmentor with respect to block size for a composite picture made from CCITT document 4. Here $t_1 = 28$, $t_2 = 195$. As we view such a segmented picture we realize that a 5×5 block may be too small. A 9×9 block appears quite adequate even when the added noise variance reaches 758. Higher block sizes result in a larger number of contiguous gray pels, thereby decreasing the number of segments. Figure 3 shows a segmented picture. Due to equipment limitations we show only a 512×512 section of the

Table I—Performance of the segmentor

Block Size	No. of Gray Pels*		No. of Segments	
	Without Noise	With Noise†	Without Noise	With Noise
5 × 5	221,045	257,952	5394	27,910
9 × 9	222,818	226,691	3918	6662
15 × 15	224,595	224,598	3052	3062

* Total number of gray pels is 512 × 512.
† Variance of the noise = 758 (8-bit, 0-255 scale).

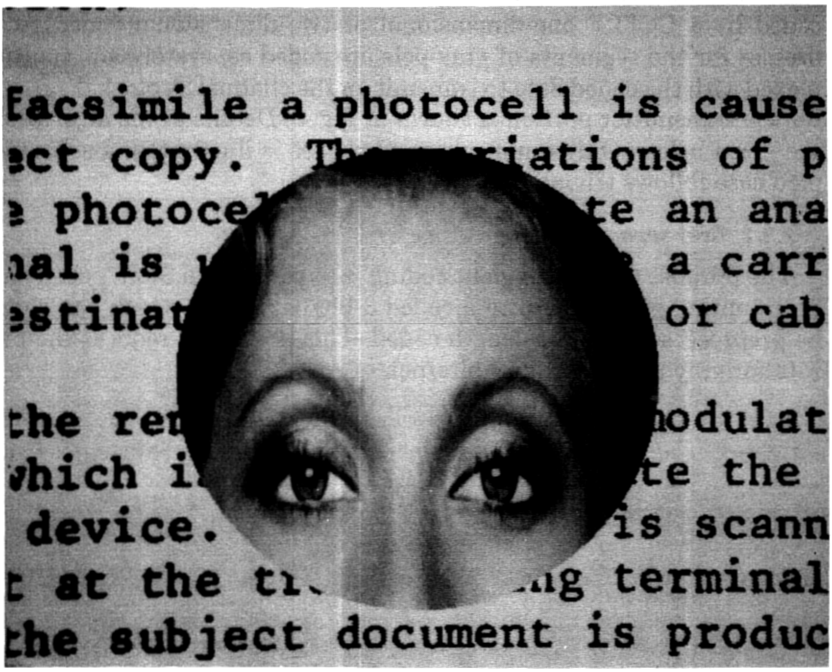

Fig. 3—A segmented picture. Pels classified black and white are reproduced with intensities 30 and 215, respectively. Gray-level pels are reproduced with 8-bit intensities.

composite picture. The segmentor has adequately separated the two-level areas from the multilevel areas.

2.2 Subsampling and interpolation

In most cases, areas of the picture that are segmented to be gray do not need as much spatial resolution as the two-level segments. If the two-level picture is at a very high spatial resolution (e.g., 200 pels/inch), then without any significant loss of quality, spatial resolution can be reduced in gray areas. Following is a scheme for subsampling and interpolation. A subsampling pattern is shown in Fig. 4. Interpolation is performed by averaging four surrounding pels, as in Fig. 4. Although we show only 2:1 subsampling, higher subsampling ratios may be used if the quality requirements are not very high. Also, two-dimensional subsampling may be performed, but this may increase the complexity.

2.3 Coding

After the pels (black, white, or gray) are classified and the gray areas to be transmitted are determined, a DPCM coder is used for gray areas. The resulting bit stream from the DPCM coder is preprocessed, multiplexed with bits from the two-level segments, and then coded by a CCITT one-dimensional or two-dimensional coder. Addresses for the segments of gray pels are coded separately and multiplexed with the coded data to transmit on the channel. A block diagram for the transmitter portion is shown in Fig. 5. Details of the algorithm are given below. Only a nonsubsampled case is illustrated; a subsampled case follows trivially.

2.3.1 Grey segment coding

The purpose of gray segment coding is to convert an 8-bit/pel signal representing gray areas into a coded 3-bit/pel signal, which can then be preprocessed and run-length coded. This procedure reduces the bit rate for gray pels to about 2 bits/pel.

x	o	x^C	o	x
o	x^B	o^A	x^D	o
x	o	x^E	o	x
o	x	o	x	o

INTERPOLATION OF $A = (B + C + D + E)/4$

X: SAMPLE SELECTED FOR TRANS.
C: SAMPLE DROPPED

Fig. 4—Subsampling and interpolation pattern used in gray areas. Only one-dimensional subsampling is considered.

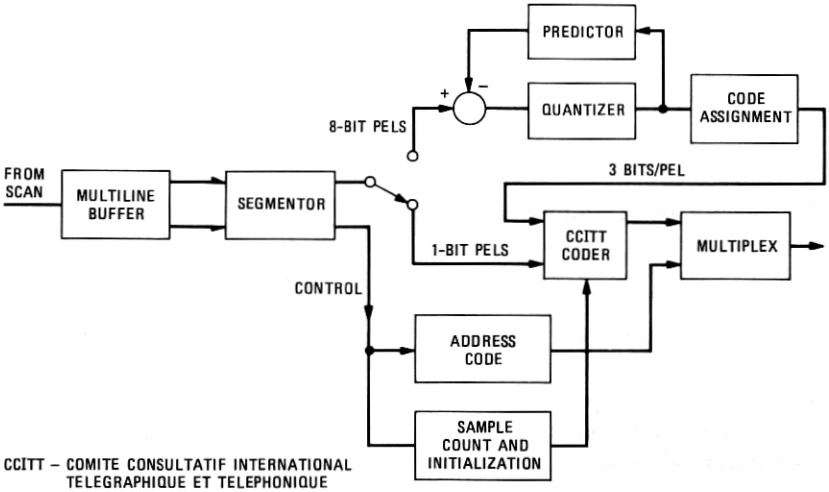

Fig. 5—Block diagram of the transmitter portion of the coder.

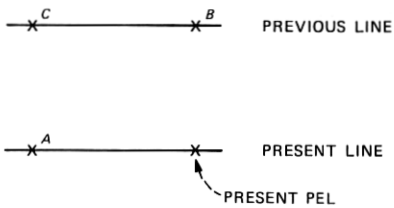

Fig. 6—Configuration of pels used for prediction. Only a nonsubsampled case is shown.

2.3.2 DPCM predictor

On Fig. 6 we see that the present pel is predicted by

$$\begin{aligned}\hat{X} &= \text{prediction of the present pel}\\ &= 0.5A + 0.25(B + C).\end{aligned}$$

It is assumed in this figure that all elements A, B, C are gray elements. Appropriate modification is made if some of these are two-level elements.

2.3.3 DPCM quantizer

The prediction error is quantized by a symmetric seven-level quantizer with the transfer characteristics given in Fig. 7. For most pictures with a resolution of 100 pels/inch, this appears adequate, although in some cases dynamic range may not be sufficient. Subjective studies are needed to optimize the characteristics for a given set of pictures.

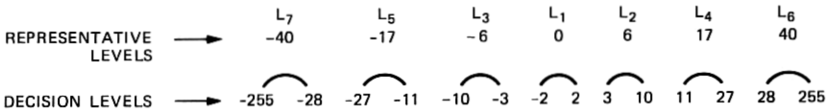

Fig. 7—Transfer characteristics of the quantizer.

Efficiency can be improved further by adapting the prediction and quantization.

2.3.4 Code assignment

To reduce statistical redundancy and create a bit stream that can be coded compatibly with the CCITT code, seven levels of the quantizer output are mapped into a three-bit code. First, a table of 49 states is constructed by looking at the seven outcomes of the quantized prediction values for both elements A and B (in Fig. 6). Given a state, the code words for the present pel are arranged in order of conditional frequency of occurrence. Such statistics are precomputed for a set of pictures. The code word that is most frequent (for a given state) is given the code [000], the next highest is given code [001], etc. In addition, to decrease the probability of occurrence of isolated '1', if the last bit of the code word for A is a '1', then the entire code word for the present pel is complemented (i.e., '0' → '1', and '1' → '0'). The table of 49 states and the corresponding code words are shown in Table II.

2.4 Preprocessing

The code words for various states (e.g., runs) for the CCITT scheme are already defined based on the statistics. The statistics of the states for the gray-level segments are quite different. As an example, Fig. 8 shows histograms of the runs for black and white pels on which the one-dimensional CCITT code is based. The same figure also shows the histograms of the runs of the bits from gray-level picture (only 512×512) with the code assignment of the previous section but without any bit complementing. It is clear that the histograms are not similar in shape, and therefore using the CCITT code for runs of bits from gray segments would not be efficient. Since our experience shows that the two-dimensional CCITT code is not efficient for the gray segments, we give below a method of preprocessing that makes efficient use of the one-dimensional CCITT code. Let $n_c^b(i)$ and $n^b(i)$ be the histograms of the runs of black elements for the CCITT code and the gray-level segments, respectively. Also let $c(i)$ be the code assigned to the ith run by the CCITT coder. Let $j(i)$ be the sequence that is arranged in descending order of the histogram function $h_c^b(i)$, i.e.,

$$h_c^b(j(i)) \le h_c^b[j(i-1)]\,.$$

Similarly, arrange the gray-level histogram $h^b(i)$ with the function $j^*(i)$. Then the code word for a length $j^*(i)$ of the gray segment is the same as $c(j(i))$. Thus, we arrange the two histograms in descending order and choose the code to be the same for entries of both the

Table II—Uncomplemented code words

No.	Quantizer Level for B	Quantizer Level for A	Decreasing Frequency of Occurrence –> 000	001	011	111	110	100	010
1	L_1	L_1	L_1	L_2	L_3	L_4	L_5	L_6	L_7
2	L_1	L_2	L_1	L_2	L_3	L_4	L_5	L_6	L_7
3	L_1	L_3	L_1	L_3	L_2	L_5	L_4	L_6	L_7
4	L_1	L_4	L_2	L_1	L_4	L_3	L_7	L_5	L_6
5	L_1	L_5	L_3	L_5	L_1	L_2	L_6	L_4	L_7
6	L_1	L_6	L_5	L_6	L_3	L_2	L_7	L_1	L_4
7	L_1	L_7	L_7	L_4	L_2	L_6	L_5	L_3	L_1
8	L_2	L_1	L_1	L_3	L_2	L_4	L_5	L_7	L_6
9	L_2	L_2	L_1	L_2	L_3	L_4	L_5	L_7	L_6
10	L_2	L_3	L_3	L_1	L_2	L_5	L_4	L_6	L_7
11	L_2	L_4	L_2	L_4	L_1	L_3	L_7	L_5	L_6
12	L_2	L_5	L_3	L_5	L_1	L_2	L_6	L_4	L_7
13	L_2	L_6	L_6	L_5	L_7	L_3	L_4	L_1	L_2
14	L_2	L_7	L_7	L_5	L_4	L_6	L_2	L_3	L_5
15	L_3	L_1	L_1	L_2	L_3	L_4	L_5	L_6	L_7
16	L_3	L_2	L_1	L_2	L_3	L_4	L_5	L_7	L_6
17	L_3	L_3	L_1	L_3	L_2	L_5	L_4	L_6	L_7
18	L_3	L_4	L_2	L_1	L_4	L_3	L_7	L_5	L_6
19	L_3	L_5	L_3	L_1	L_5	L_2	L_6	L_4	L_7
20	L_3	L_6	L_6	L_5	L_3	L_2	L_1	L_7	L_4
21	L_3	L_7	L_7	L_6	L_5	L_3	L_2	L_4	L_1
22	L_4	L_1	L_3	L_1	L_2	L_5	L_4	L_7	L_6
23	L_4	L_2	L_2	L_1	L_3	L_4	L_5	L_7	L_6
24	L_4	L_3	L_3	L_1	L_5	L_2	L_4	L_7	L_6
25	L_4	L_4	L_4	L_2	L_1	L_7	L_3	L_5	L_6
26	L_4	L_5	L_5	L_3	L_1	L_2	L_6	L_4	L_7
27	L_4	L_6	L_6	L_1	L_5	L_4	L_2	L_3	L_7
28	L_4	L_7	L_7	L_4	L_2	L_3	L_6	L_1	L_5
29	L_5	L_1	L_2	L_1	L_3	L_4	L_5	L_6	L_7
30	L_5	L_2	L_1	L_2	L_4	L_3	L_5	L_6	L_7
31	L_5	L_3	L_3	L_1	L_2	L_5	L_4	L_6	L_7
32	L_5	L_4	L_4	L_2	L_7	L_1	L_3	L_5	L_6
33	L_5	L_5	L_5	L_3	L_6	L_1	L_2	L_4	L_7
34	L_5	L_6	L_6	L_5	L_3	L_1	L_2	L_7	L_4
35	L_5	L_7	L_6	L_7	L_4	L_5	L_2	L_1	L_3
36	L_6	L_1	L_1	L_6	L_7	L_3	L_2	L_5	L_4
37	L_6	L_2	L_6	L_7	L_2	L_4	L_5	L_1	L_3
38	L_6	L_3	L_3	L_6	L_2	L_4	L_1	L_7	L_4
39	L_6	L_4	L_7	L_6	L_4	L_2	L_1	L_3	L_5
40	L_6	L_5	L_5	L_6	L_3	L_1	L_2	L_4	L_7
41	L_6	L_6	L_6	L_5	L_3	L_7	L_1	L_4	L_2
42	L_6	L_7	L_6	L_7	L_5	L_4	L_2	L_3	L_1
43	L_7	L_1	L_7	L_5	L_3	L_4	L_6	L_2	L_1
44	L_7	L_2	L_7	L_2	L_3	L_4	L_1	L_5	L_6
45	L_7	L_3	L_7	L_5	L_6	L_4	L_3	L_1	L_2
46	L_7	L_4	L_4	L_7	L_2	L_3	L_1	L_6	L_5
47	L_7	L_5	L_7	L_6	L_5	L_3	L_4	L_2	L_1
48	L_7	L_6	L_7	L_6	L_4	L_5	L_2	L_3	L_1
49	L_7	L_7	L_7	L_4	L_1	L_2	L_3	L_6	L_5

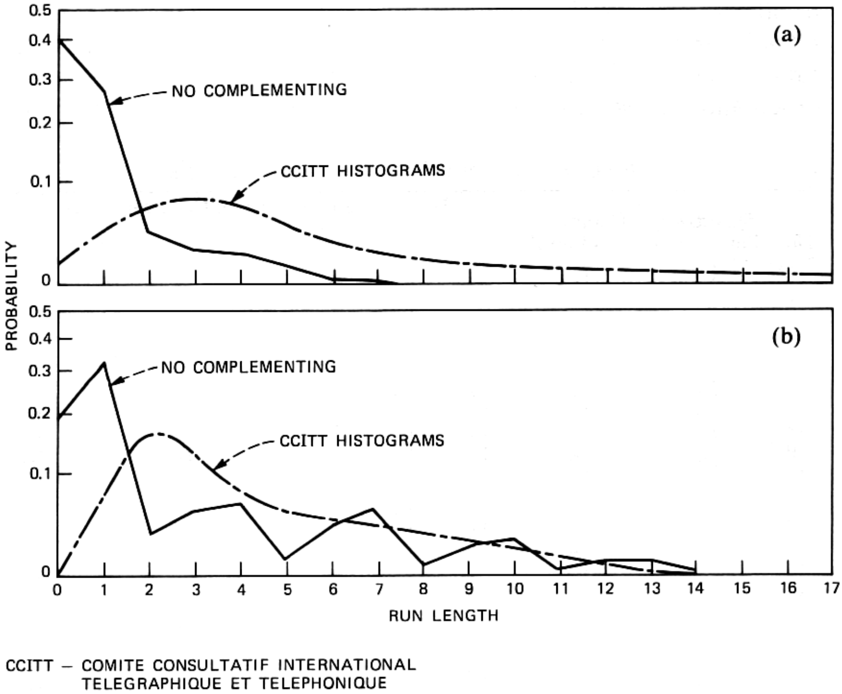

Fig. 8—Histograms of the (a) white and (b) black runs used in the 1D CCITT code and the processed gray-level pictures.

rearranged histograms. We found that in reality much of the gain in coding efficiency can be obtained by exchanging the code words for a few run lengths. This leads to simple preprocessing. Figure 9 shows the results of preprocessing on the histograms. It is clear that, if bit complementing is not used, after the preprocessing the code set is more attached to the histograms and therefore leads to more efficient code. However, if bit complementing is used, the histogram without any preprocessing is not too different from the CCITT histogram. Therefore, when bit complementing is used, the advantages of preprocessing are not large. Although this is the case for the picture we considered, more experiments are needed to evaluate statistics of typical pictures and usefulness of the preprocessing for such statistics.

2.5 Addressing

To encode positional information of the boundaries of a segmented picture, each composite line is considered as a sequence of alternating black and white runs corresponding to the lengths of two-level and gray-level segments, respectively. This is then coded by the two-

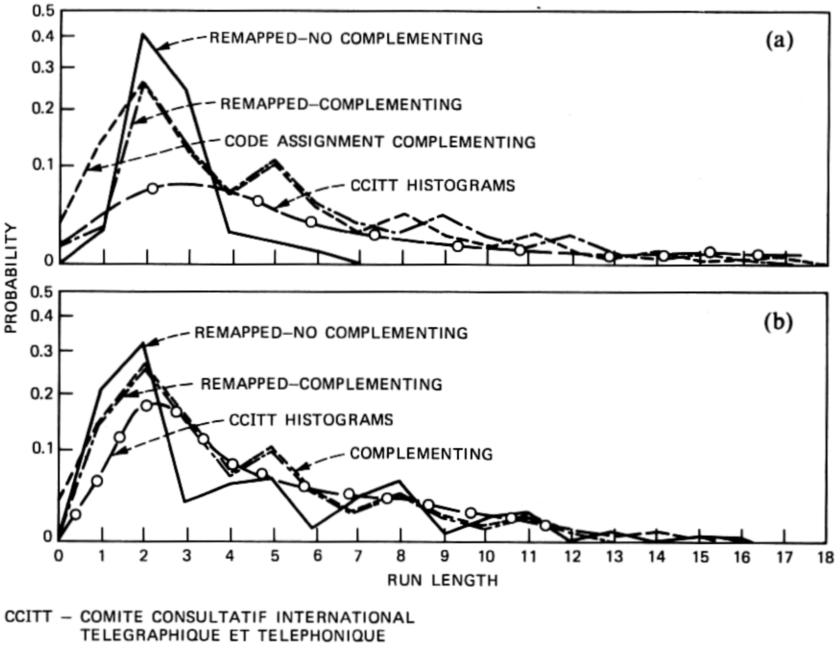

Fig. 9—Effect of preprocessing on the histograms of the (a) white and (b) black runs.

dimensional CCITT code and is transmitted at the beginning of each composite line.

2.6 Multiplexing and CCITT coding

The bits resulting from the above procedure for gray areas are multiplexed pel by pel with those of the two-level pels. Our experiments show that while it is advantageous to encode two-level areas by two-dimensional code, most of the two-dimensional correlation in gray segments is removed by two-dimensional prediction and code assignment. Therefore, gray areas are coded by one-dimensional code. Since the number of gray-level pels may vary from line to line, in order to maintain proper registration of two-level pels (for two-dimensional coding), a sample count is maintained and is used to initialize the two-dimensional coder once it comes out of the gray segment within a line.

III. SIMULATION RESULTS

Results of computer simulations are given in Tables III and IV. Table III shows results for 512 × 512 gray-level picture, and Table IV shows results for composite pictures with CCITT standard documents 4 and 5. It is clear from Table III that for the gray-level picture, without any preprocessing or bit complementing, coding efficiency is

Table III—Performance of coding algorithms for 512 × 512 gray picture

No.	Coding Algorithms	Coded bits (bits/pel)
1.	Entropy of the quantizer output (no subsampling)	1.84
2.	Entropy of the quantizer output (2:1 subsampling)	1.05
3.	Entropy with one-dimensional run-length coding (no complementing, no preprocessing, no subsampling)	2.76
4.	Entropy with one-dimensional run-length coding (no complementing, no preprocessing, 2:1 subsampling)	1.98
5.	One-dimensional run-length coding (no complementing, no preprocessing, CCITT code, no subsampling)	3.68
6.	One-dimensional run-length coding (no complementing, no preprocessing, CCITT code, 2:1 subsampling	2.86
7.	5+ preprocessing	3.14
8.	6+ preprocessing	2.24
9.	Entropy with one-dimensional run-length coding (complementing, no preprocessing)	1.88
10.	Entropy with one-dimensional run-length coding (complementing, no preprocessing, 2:1 subsampling)	1.17
11.	9+ CCITT code	2.05
12.	10+ CCITT code	1.31
13.	11+ preprocessing	1.98
14.	12+ preprocessing	1.19

Table IV—Performance of coding algorithms for composite pictures

No.	Coding algorithms	Coded bits	
		Document 4	Document 5
1.	One-dimensional CCITT code on noncomposite document	870803	547853
2.	Two-dimensional CCITT code on noncomposite document	577527	286911
3.	Two-dimensional code for two-level, one-dimensional code for gray level (no complementing)	1169270	893288
4.	Two-dimensional code for two-level, one-dimensional code for gray level (no complementing), 2:1 subsampling	961210	686777
5.	3+ preprocessing	1086589	811897
6.	4+ preprocessing	879796	601189
7.	(3) + complementing	909876	628626
8.	(4) + complementing	755133	470539
9.	(5) + complementing	894444	616956
10.	(6) + complementing	739874	453229
11.	Two-dimensional code for entire document	999125	714651
12.	Bits for addressing	18147	18153

rather low. This is a result of the mismatch of the run-length statistics. Considerable improvement is obtained by preprocessing the run lengths before applying the CCITT coder. Even higher improvement is obtained by the bit-complementing technique. Much of the mismatch between the statistics is removed by the complementing technique, and therefore additional improvement obtained by preprocessing the complemented output is marginal. The use of complementing makes it possible to achieve bit rates that are close to the entropy of the coded output. Coding of composite pictures shows similar results. Another interesting conclusion from Table IV is that the two-dimensional CCITT code is not very efficient for gray-level segments of the composite picture. This is a result of lack of line-to-line correlation among bits that are outputs of the quantizer. Much of the line-to-line correlation is already removed by the two-dimensional prediction and the bit assignment based on conditional statistics.

IV. CONCLUSIONS

We have presented an algorithm that can automatically segment areas of a picture that require only two shades of gray from those that require many shades of gray. Gray areas are coded in a way that creates a bit stream that subsequently can be efficiently coded by a CCITT coder. We find that, for the gray areas, it is possible to achieve coding efficiencies close to the entropy of the DPCM quantizer output. Therefore, we conclude that it is possible to encode documents that contain an arbitrary mixture of two-level and multilevel areas using a CCITT coder that requires only a preprocessor at the transmitter and a postprocessor at the receiver.

REFERENCES

1. A. N. Netravali, ed., special issue on Digital Encoding of Graphics, Proc. IEEE, *68,* No. 7 (July 1980).
2. A. N. Netravali and J. O. Limb, "Picture Coding—A Review," Proc. IEEE, *68,* No. 3 (March 1980), pp. 366–406.
3. R. Hunter and A. H. Robinson, "International Digital Facsimile Coding Standards," Proc. IEEE, *68,* No. 7 (July 1980), pp. 830–46.

AUTHORS

Arun N. Netravali, B. Tech. (Honors), 1967, Indian Institute of Technology, Bombay, India; M.S., 1969, Ph.D. (Electrical Engineering), 1970, Rice University; Optimal Data Corporation, 1970–1972; Bell Laboratories, 1972—. Mr. Netravali has worked on problems related to filtering, guidance, and control for the space shuttle. At Bell Laboratories, he has worked on various aspects of digital processing and computing. He was a Visiting Professor in the Department of Electrical Engineering at Rutgers University. He is presently Director of the Computer Technology Research Laboratory. Mr. Netravali holds over 20 patents and has had more than 60 papers published. He was the recipient of the Donald Fink Prize Award for the best review paper published in the Proceedings of the IEEE. Editorial board, Proceedings of the IEEE; editor, Transactions on Communications; member, Tau Beta Pi, Sigma Xi; Senior Member, IEEE.

Hamid Gharavi, B.S.E.E., 1970, Tehran Polytechnic; M.S.C. (Digital Communication), Ph.D. (Electrical Engineering), 1979, University of Technology, Loughborough, England; Research fellow at University of Technology, Loughborough, England, 1979–1980; Lecturer at Auckland University, New Zealand, 1980–1981; Bell Laboratories, 1982—. Mr. Gharavi has worked on problems related to digital modulations for satellite communication. He also has worked on bandwidth compression of color television signals, source coding, graphics, and pattern recognition. Member, IEEE.