Regulations for the Use of Theses

Unpublished theses submitted for the Master's and Doctor's degrees and deposited in the Georgia Institute of Technology Library are open for inspection and consultation, but must be used with due regard for the rights of the authors. Passages may be copied only with permission of the authors, and proper credit must be given in subsequent written or published work. Extensive copying or publication of the thesis in whole or in part requires the consent of the Dean of the Graduate Division of the Georgia Institute of Technology.

This thesis by STANLEY WARREN JOHNSTON has been used by the following persons, whose signatures attest their acceptance of the above restrictions.

A library which borrows this thesis for use by its patrons is expected to secure the signature of each user.

NAME AND ADDRESS OF USER BORROWING LIBRARY DATE
In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

3/17/65
A METHOD OF PROCESSING AND CODING PICTORIAL INFORMATION TO REDUCE REDUNDANCY

A THESIS
Presented to
The Faculty of the Graduate Division
by
Stanley Warren Johnston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

Georgia Institute of Technology
August, 1966
A METHOD OF PROCESSING AND CODING PICTORIAL INFORMATION TO REDUCE REDUNDANCY

Approved:

Chairman

Date approved by Chairman 11-1-66
ACKNOWLEDGMENTS

The author wishes to express his gratitude to Dr. John B. Peatman, his thesis advisor, for valuable advice and guidance. He also wishes to thank Dr. Edward E. David of Bell Telephone Laboratories for providing the digitized pictures on magnetic tape used in this research. The help of Drs. Benson Perry and Mortimer Mendelsohn of the University of Pennsylvania in furnishing details of their method of using a line printer as a pictorial computer output is also appreciated. Extensive use was made of the services of the Rich Electronic Computer Center in the work described herein.

The writer was supported by a National Science Foundation Training Grant during this research.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>v</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>vi</td>
</tr>
<tr>
<td><strong>Chapter</strong></td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>The Problem</td>
<td></td>
</tr>
<tr>
<td>The Statistical Approach</td>
<td></td>
</tr>
<tr>
<td>The Psychophysical Approach</td>
<td></td>
</tr>
<tr>
<td>The Research</td>
<td></td>
</tr>
<tr>
<td>II. HISTORY OF THE PROBLEM</td>
<td>5</td>
</tr>
<tr>
<td>III. THE EXPERIMENTAL ARRANGEMENT</td>
<td>8</td>
</tr>
<tr>
<td>IV. A NEW CODING METHOD</td>
<td>11</td>
</tr>
<tr>
<td>The Basis of the Process</td>
<td></td>
</tr>
<tr>
<td>Coding the Picture</td>
<td></td>
</tr>
<tr>
<td>Some Details of the Process</td>
<td></td>
</tr>
<tr>
<td>Hysteresis Smoothing</td>
<td></td>
</tr>
<tr>
<td>Edge Sharpening</td>
<td></td>
</tr>
<tr>
<td>The Averaging Process</td>
<td></td>
</tr>
<tr>
<td>The Boundaries</td>
<td></td>
</tr>
<tr>
<td>The Statistical Areas</td>
<td></td>
</tr>
<tr>
<td>Decoding the Picture</td>
<td></td>
</tr>
<tr>
<td>V. RESULTS</td>
<td>53</td>
</tr>
<tr>
<td>VI. CONCLUSIONS</td>
<td>62</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>64</td>
</tr>
<tr>
<td>LITERATURE CITED</td>
<td>90</td>
</tr>
<tr>
<td>OTHER REFERENCES</td>
<td>92</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Gray Scale Code Set</td>
<td>9</td>
</tr>
<tr>
<td>2. Encoding Boundary Heights</td>
<td>34</td>
</tr>
<tr>
<td>3. Transmitting the Variance</td>
<td>49</td>
</tr>
<tr>
<td>4. Efficiency of the Code</td>
<td>58</td>
</tr>
<tr>
<td>5. Effect of Preprocessing and Statistical Areas on Picture Complexity</td>
<td>60</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Three Components of a Picture Illustrated by a One-Dimensional Example</td>
<td>14</td>
</tr>
<tr>
<td>2.</td>
<td>Flow Diagram for Picture Processing</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>Hysteresis Quantization</td>
<td>19</td>
</tr>
<tr>
<td>4.</td>
<td>Edge Sharpening</td>
<td>21</td>
</tr>
<tr>
<td>5.</td>
<td>The Averaging Process</td>
<td>26</td>
</tr>
<tr>
<td>6.</td>
<td>Flow Chart for Tracing Boundaries</td>
<td>35</td>
</tr>
<tr>
<td>7.</td>
<td>State Diagram for Code Word Sequence in Boundary Encoding</td>
<td>40</td>
</tr>
<tr>
<td>8.</td>
<td>Margin of Error in Piecewise Linear Approximation</td>
<td>44</td>
</tr>
<tr>
<td>9.</td>
<td>Flow Chart for Translating Boundary Trace into Piecewise Linear Code</td>
<td>45</td>
</tr>
<tr>
<td>10.</td>
<td>Decoding Procedure for Boundaries</td>
<td>50</td>
</tr>
<tr>
<td>11.</td>
<td>Original and Decoded Pictures</td>
<td>55</td>
</tr>
<tr>
<td>12.</td>
<td>Intermediate Steps in Processing Picture &quot;B&quot;</td>
<td>56</td>
</tr>
</tbody>
</table>
SUMMARY

The problem of efficiently encoding pictorial information is attacked. After a survey of the literature, a complete method of specifying digitized still black-and-white photographs to some accuracy with relatively few bits of information is designed and tested by computer simulation. The decoded pictures are of tolerable quality, while the number of bits used is cut by about 5:1 from straightforward point-by-point encoding.

The entire process is simulated by a computer program and pictures are coded and decoded. Digitized pictures are used as input data to the computer, and pictorial output is handled by a procedure for generating pictures with a line printer.

The method of coding developed is complex to implement. It deals with the picture in two dimensions without scanning.

The basic principle of the code is the separation of the picture into three components. The first one is a smoothly varying low-space-frequency part to be encoded by sampling. The second part is the sharp edges where the reflectivity of the picture changes suddenly. These are coded using piecewise linear approximations. Finally, the third component is a stochastic part consisting of those features of great complexity to be specified only by their statistics.

The receiver regenerates the low frequency component and adds synthetic edges as determined by the second component. These artificial edges sharpen up changes in the smoothly varying part. Then random
features with the required statistics are added to reconstruct the third component, and complete the picture.

The implementation involves the solution of numerous problems. Before coding, the picture is subjected to two preprocessing operations which smooth the small variations in the picture and sharpen the edges. The smoothing operation requires the development of the concept of "hysteresis quantization" as a method of preventing spurious fluctuations. This data is used again at another step in the coding.

For the generation of the low space frequency component and for various smoothing and averaging tasks, an averaging routine is developed which becomes a work-horse of the procedure. An analysis of the process allows theoretical prediction of its effects.

In the course of encoding, significant problems arise and are solved in the areas of piecewise linear approximation, run-length coding, and adaptive sampling.

Although the code developed is hardly a panacea for the difficulties of efficient picture coding, it does illuminate some approaches to this problem.
CHAPTER I

INTRODUCTION

The Problem

The communication of pictorial data typically involves transmitting a burdensome amount of information. The coding process which is almost invariably used is to scan a picture, sending the reflectivity of each point in turn. In a digital system, the picture is quantized in both dimensions in space, and in reflectivity. If there are $m$ vertical quanta, $n$ horizontal quanta, and $r$ quanta on the reflectivity scale, there are $r^{mn}$ possible pictures. The usual scheme assigns a code word of $mn \log_2 r$ bits to each picture.

Clearly, if code words shorter than this are to be assigned to the pictures one expects to transmit, one must either assign code words longer than this to some possible pictures, or transmit some pictures incorrectly. There are hence two approaches to the problem: the statistical and the psychophysical. The statistical approach attempts to assign short code words to the pictures most likely to be sent, at the expense of longer code words for pictures less likely to be transmitted. The psychophysical approach transmits pictures incorrectly, but attempts to make the distortions of such a nature that they are not noticeable to the human sense of vision. Of course, it is possible to combine these two methods.
The Statistical Approach

Imagine that a picture is formed by selecting at random one of the reflectivity levels for each space quantum, independent of the other picture elements. After looking at several such random patterns, one has the subjective feeling that the probability of forming a meaningful picture by this process is fantastically small. This would indicate that the set of pictures that any individual might want to transmit is a very small subset of the set of all possible pictures. Hopefully, this means that shorter code words could suffice to describe realistic pictures.

It is obvious that the class of pictures which might be sent and the class of those which would never be transmitted are not well-defined, disjoint sets. Rather, the probability that a given picture be sent varies from picture to picture. It is furthermore obvious that the number of pictures involved precludes the empirical determination of the probabilities. Hence, the success of the statistical approach to picture coding hinges upon discovering some criterion for determining which pictures will be "common".

In essence, the major fact known about "realistic" pictures as opposed to random patterns is this: large areas of a picture often have nearly the same reflectivity. Or, conversely, most of the information in a picture is in the boundaries between sections of different reflectivities. This is illustrated by the ease with which one recognizes cartoons, in which only the boundaries are shown.

The Psychophysical Approach

There are two psychophysical properties of human sight which indicate what errors of picture transmission may go unnoticed. The first is
that the human eye is sensitive to changes in reflectivity (i.e. the boundaries between regions of different reflectivities), but relatively insensitive to the magnitude of such changes\(^2\). In other words, the exact shape and amplitude of changes in brightness is much less important than the existence and position of these edges, or boundaries, and the density or reflectivity in the areas enclosed by them\(^3\).

The other piece of information about the phenomenon of vision which may be used to advantage in designing codes for pictorial data is that the appearance of a complex pattern is largely determined by the first and second statistical moments of the reflectivity\(^4\). Equivalently, the mean and variance of the reflectivity of the picture elements in an area are sufficient to describe its appearance if the pattern is very complex and essentially random. It is common experience that if each picture element occupies a sufficiently small segment of the field of vision, the mean alone is an adequate description.

**The Research**

With these intellectual tools, the problem of coding pictorial information efficiently was attacked. A method of processing and coding digitized still black-and-white pictures was designed and tested by computer simulation. The objective, to specify typical photographs to reasonable accuracy with relatively few bits of information, has been sought by others, whose work is discussed in the next chapter.

Another class of codes utilizes psychophysical properties of the observer by sending either small changes accurately or large changes roughly. They send the same number of bits for every picture element,
suggest transmitting either the three most significant bits or the three least significant bits of the six bit reflectivity. The most significant bits would be sent only if they change from picture element to picture element, as the picture is scanned. David attained similar results by transmitting the difference between the reflectivity of each point on the picture and the previously transmitted one, using logarithmic quantization. Thus, small changes are sent accurately, while large ones are more coarsely quantized.

A purely statistical approach is predictive coding. The procedure is to estimate the reflectivity of the point in question, and then send the error of the estimate. With a good estimate, the error is likely to be small. The most likely values are assigned a short code word, while the less likely ones are represented by longer code words which do not contain a shorter one as a prefix. The problem is the choice of a good estimate.

The simplest one, the previously transmitted point, is discussed by Knight, et al. Youngblood tried using the average of several surrounding points, while Corradetti predicted points by extending a plane through three previously transmitted points. Wholey used an empirical determination of probabilities to estimate points on two-level (black and white) images.
CHAPTER II

HISTORY OF THE PROBLEM

The most obvious way to reduce the length of a code word representing a picture from its original value of $mn \log_2 r$ is to reduce $m$, $n$, or $r$. Reducing $m$ or $n$ seems to cause irreparable loss of resolution, but $r$, the number of levels of reflectivity to be represented, is not so rigidly fixed.

Good quality pictures require about six bits of reflectivity information per picture point, that is, about $6^4$ levels between black and white. Finer division of the brightness scale produces no noticeable results, but a much coarser quantization impairs picture quality. The worst effect is "contouring", the creation of apparent boundaries due to quantization noise. A region of the unquantized picture may vary smoothly over a range of densities, but quantization breaks this up into discrete changes in density. If the quantization is coarse, these jumps are visible and undesirable.

At least three ways of overcoming this difficulty have been tried. R. S. Marcus tried having the receiver smooth the picture. Roberts added pseudo-random noise to his pictures before quantizing so that the average reflectivity in a region is portrayed accurately. A similar result was achieved by Bisignani, et al., who stored up quantization error until it was sufficient to change one picture element by a whole quantum of reflectivity.
If the estimator is very good, and the error is often zero, it may be expedient to encode the distance between errors. This is run-length coding, as discussed by Wholey\textsuperscript{12}.

A final class of picture coding procedures separates the picture into two components: the slowly varying part corresponding to low space frequencies, and the rapidly changing part consisting of edges. The low frequency component needs to be transmitted accurately, but it can be sampled coarsely in space. The high frequency part, on the other hand, need not be reproduced with great accuracy, as long as edge positions are preserved.

Cunningham\textsuperscript{13} worked on a method for sending a low frequency signal plus roughly quantized correction points as needed. Julesz\textsuperscript{2} wrote of a procedure for encoding pictures based on the detection of edges. Not unlike this is Schreibner's "synthetic highs" scheme, a semi-digital approach. The low frequencies are sent as an analog signal, while the position and amplitude of edges is sent by digital code. The receiver manufactures a "synthetic" high frequency component based on this edge information. Pan\textsuperscript{11} has written about experiments with piece-wise linear approximations of the edges in two dimensions.

Most of these procedures are either explicitly or implicitly one-dimensional, and thus require scanning the picture to transform the two dimensional picture into a function of one dimension. This is often convenient for implementation, but it throws away the redundancy of the picture in one direction. In many cases, a two-dimensional extension of these methods can be imagined.

Some of the simpler methods of coding, such as those of Bisignani,
Roberts, David, Knight, and Schreibner reduce the information content of a picture by about 2:1 with very little deleterious effect on the image. Several of the more complex procedures, such as those described by Pan and Cunningham, cut the number of bits per picture by figures on the order of 6:1 to 10:1, but only with significant deterioration of picture quality.
CHAPTER III

THE EXPERIMENTAL ARRANGEMENT

Picture processing was simulated on a large general purpose digital computer, the Burroughs B-5500, at the Rich Electronic Computer Center. Sampled, quantized pictures were fed to the computer, which first coded, then decoded them. The original and reconstructed pictures were produced by the machine, along with the calculated number of bits in the code for each picture.

The input pictures were fed to the computer on digital magnetic tape. Four pictures were sampled, quantized, and recorded on tape by special equipment at Bell Telephone Laboratories, and a copy of this tape was made available for this research by Dr. Edward E. David of Bell Laboratories.

Output from the computer was by means of the line printer. Pictures were produced by printing various characters, including some overprints, to produce different shades of gray on white paper. The procedure used for this was a slight modification of that developed by Drs. Benson Perry and Mortimer Mendelsohn at the University of Pennsylvania\textsuperscript{15}. They graciously provided the details of their method.

Two printing positions were used for every picture element. The combinations of characters shown in Table 1 were used to produce 32 different shades of gray.

Since there are ten printing positions per inch horizontally, and
Table 1. Gray Scale Code Set

<table>
<thead>
<tr>
<th>Print</th>
<th>Overprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>:</td>
</tr>
<tr>
<td>4</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>L</td>
</tr>
<tr>
<td>6</td>
<td>V</td>
</tr>
<tr>
<td>7</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
</tr>
<tr>
<td>9</td>
<td>H</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>W</td>
</tr>
<tr>
<td>12</td>
<td>/</td>
</tr>
<tr>
<td>13</td>
<td>//</td>
</tr>
<tr>
<td>14</td>
<td>L/</td>
</tr>
<tr>
<td>15</td>
<td>V/</td>
</tr>
<tr>
<td>16</td>
<td>LL</td>
</tr>
<tr>
<td>17</td>
<td>VL</td>
</tr>
<tr>
<td>18</td>
<td>VV</td>
</tr>
<tr>
<td>19</td>
<td>YV</td>
</tr>
<tr>
<td>20</td>
<td>XV</td>
</tr>
<tr>
<td>21</td>
<td>XY</td>
</tr>
<tr>
<td>22</td>
<td>XX</td>
</tr>
<tr>
<td>23</td>
<td>NX</td>
</tr>
<tr>
<td>24</td>
<td>WX</td>
</tr>
<tr>
<td>25</td>
<td>WW</td>
</tr>
<tr>
<td>26</td>
<td>XX</td>
</tr>
<tr>
<td>27</td>
<td>N</td>
</tr>
<tr>
<td>28</td>
<td>XX</td>
</tr>
<tr>
<td>29</td>
<td>N</td>
</tr>
<tr>
<td>30</td>
<td>VW</td>
</tr>
<tr>
<td>31</td>
<td>AA</td>
</tr>
<tr>
<td>32</td>
<td>WW</td>
</tr>
<tr>
<td></td>
<td>$$$</td>
</tr>
</tbody>
</table>
six vertically, each picture element is $1/5$ by $1/6$ inch. The picture elements in the originals are square, so some distortion is introduced.

In order to view the output conveniently, four pages of printing were combined to form a picture which is photographed. A very high contrast film, Kodak Kodalith Ortho 3, was used to increase the contrast from the low level of the printout to normal. Prints made from the negatives were defocused slightly to blend the density of the picture over a whole picture element. The resulting pictures are reproduced elsewhere in this volume.
CHAPTER IV

A NEW CODING METHOD

The Basis of the Process

An attempt was made to design a sophisticated picture processor which would drastically reduce the information required to specify pictures. Simplicity of implementation was not considered to be of overriding importance. Consideration was not restricted to local, as opposed to global operators, nor was scanning assumed. Many of the processes developed seem most suitable for implementation by a special purpose computer like that discussed by Unger\textsuperscript{16} or Fuller and Bird\textsuperscript{17}. These machines do parallel processing of pictures by being spatially organized into a two-dimensional array containing the picture. The computer operates on all picture elements simultaneously. This organization is useful for performing local operations, which combine information from adjacent picture elements only. Of course, the process was simulated on a general purpose serial computer.

The new picture code recognizes three types of information in pictures. First there are the boundaries. Their position and the approximate size of the discontinuity in reflectivity occurring there are the most important aspects of the picture. Second, there are smooth changes in reflectivity. This data is necessary for a natural appearance of the transmitted image. Third, there are complex patterns best described by their statistics. These could be treated as many boundaries, but it
may be prohibitively inefficient to do this.

This last type of information is probably most important for very high resolution pictures. If a picture shows a person's hair, it is neither necessary nor desirable to code the position of each hair; some statistics of the region should allow the reconstruction of an image that looks "harry". For the present work, it is assumed that the mean and variance of the reflectivity is sufficient, as Rosenfeld's paper indicates it should be. Since these functions are defined as averages, they are meaningful only for many points taken as a group. Hence, some sort of coarse sampling is indicated for the specification of this part of the image.

In a quantized picture, there are, of course, no continuous changes from one level of reflectivity to another. However, small changes, such as one quantum, can be considered separately from boundaries with large discontinuities, since the small ones may be introduced by quantization noise. The reflectivity can then be viewed as a continuous function over a domain limited by boundaries (in the restricted sense of multi-quantum boundaries). If these boundaries are removed by the substitution of continuous variations for the discontinuities, the reflectivity will be smooth and slowly varying over the whole of the picture. Such continuous functions are amenable to characterization by low frequency sampling, for example.

Finally, there are the boundaries. Generally, these will be continuous curves in the plane. They could be encoded in several ways, but one thinks immediately of tracing them point by point, and of piecewise linear approximations. From the position and amplitude of these boundaries,
the receiver can generate edges, as in the "synthetic highs" method, but in two dimensions.

The three components of the picture can be thought of this way: the low frequency part (smooth variations), the high frequency part (edges), and a stochastic component to account for some complex phenomena which are not to be included in the first two parts. The use of the first two categories is a two-dimensional extension of "synthetic highs". The third part, however, has no known analog in previous work. The separation of these three components is illustrated in Figure 1.

Coding the Picture

The diagram in Figure 2 illustrates the new method of coding pictures. The working of the individual steps in the process is explained in the next section. Here, the objective is to clarify the process as a whole.

The digitized input picture is first subjected to two preliminary processes. These two preprocessors operate on the picture to remove undesirable effects of quantization by smoothing small changes and sharpening edges.

The low frequency component is formed from the picture by an averaging process. The low frequency part is sampled and transmitted as shown on the left.

On the right side of the diagram, those areas of the picture to be encoded statistically are selected. An averaging process called Intavg averages the reflectivities within those stochastic areas. This average is not transmitted, since it is believed that the two deterministic
Figure 1. Three Components of a Picture Illustrated by a One Dimensional Example.
Figure 2. Flow Diagram for Picture Processing.
components of the picture will form the correct mean in the statistical areas. Only the variance is transmitted. The variance is formed from the original and averaged pictures in the statistical areas, and is then smoothed by Intavg. The smoothed variance is transmitted by selecting suitable sampling points, then sending the positions of these points, and the variance there.

Finally, in the center of the chart, the boundaries are formed. Boundaries are not detected in the statistical areas, but edges between stochastic and non-stochastic areas are allowed. The very short boundaries are removed, and the rest are traced out, producing a string of words describing the tracing process. The magnitude of the discontinuities along the boundaries is smoothed, and the boundaries are encoded by a piece-wise linear approximation. This process takes the previous string of words and translates it into another string, which is then transmitted.

At this point, the simulation program counts the number of bits being transmitted, then goes on to decode the picture from these bits.

The boundaries are decoded and a synthetic high frequency component is generated. The low frequency component is reproduced from the samples. The variance is also reproduced from its samples and smoothed between samples by an averaging process called Limavg. This process avoids smearing things across major boundaries of the picture. Of course, the variance is zero outside the stochastic areas. Random numbers with zero mean and the appropriate variance are generated for all the points on the picture to produce a statistical component. The addition of these three components produces the final, decoded output picture.
Some Details of the Process

Hysteresis Smoothing

The first preprocessor operates on a principle which is used again later in the coding process. This principle is called hysteresis smoothing or hysteresis quantizing for reasons which will become obvious. It is a way of quantizing an analog signal (or more coarsely quantizing a finely quantized digital signal) in a way that minimizes changes between quantization levels.

The general situation is that a function of one independent variable (often time) is to be sampled and quantized. The usual way to do this is to treat each sample independently. Between any two adjacent nominal levels of quantization there is a sharp dividing line. All samples above the line are assigned to the upper nominal level, while all those below are assigned to the lower level. A problem arises when the value of the function lies very near this threshold line. In that case, very small changes in the function from sample to sample cause the quantized signal to jump back and forth between the two nominal levels. Such quantization noise may be undesirable.

A solution to this problem is hysteresis quantization. Instead of a sharp threshold line between the range of inputs to be represented by particular quantized outputs, there is a "twilight zone". If the value of the function at a sample point lies in this zone, it may be assigned to either of the two nearest nominal values. The decision as to whether to round up or down is made so that the sample is assigned to the same level as the previous sample, if possible. Therefore, once the function is quantized to the lower of two nominal levels, it must rise
above the twilight zone before being assigned to the larger level. After that, the signal must fall completely below the twilight zone before again receiving the lower value. This hysteresis effect eliminates changes in the output signal caused by small changes in the input, and results in a smoother signal. This is illustrated by Figure 3.

When the function changes drastically between samples and happens to land in a twilight zone, the quantizer must arbitrarily assign the sample to one of the two nearby levels. If the choice is inexpedient, an unnecessary change is introduced. For example, the first sample may be assigned to the lower of the two possibilities, and the subsequent samples may rise monotonically out of the twilight zone. This effect may be eliminated by leaving these samples in doubt until all are quantized. Then the quantizer requantizes all the samples in reverse order. This will minimize the number of changes between adjacent samples subject to the constraint of the width of the twilight zone, which is imposed for the sake of accuracy.

The first preprocessor in the coding method invokes hysteresis smoothing on the picture. The reflectivity of the original picture is quantized to 11 bits, i.e., 2048 levels. It is requantized to 32 levels, or 5 bits, with the hysteresis quantizer. The twilight zones are so large that they touch and cover the entire range of inputs. That is, the twilight zones stretch from one nominal level to the next. Initially, all samples are in doubt, but are tentatively assigned to the lower of the two levels. Later they may be increased by one level in order to make them agree with their neighbors.

The picture is a function of two independent space coordinates
Figure 3. Hysteresis Quantization.
whose value is reflectivity. In doing this hysteresis quantization, the picture is scanned. But the previous sample (the one to the left) is no more important than the point above on the previous scan line. Therefore, the doubtful sample level is increased by one if this causes agreement with either of these two adjacent points. As discussed above, the quantizer operates on the samples twice—once in each direction. Care is taken to notice on the second pass which samples remain in doubt and may be altered.

The computer program which simulates the processing is presented in the appendix. Comments have been added to this and other parts of the program to help make it self-explanatory.

The effect of this preprocessor is to eliminate, as much as possible, one-quantum boundaries. It creates larger areas of equal reflectivity. It may, however, produce some two-quantum boundaries from the original one-quantum ones.

**Edge Sharpening**

The edge sharpening preprocessor acts to sharpen up edges that are somewhat smooth changes, or, what is the same thing, to combine nearby parallel boundaries into one large boundary. This is necessary because of the effect of space quantization, as illustrated in Figure 4. The smoothing occurs because the reflectivity of a picture element is the average of the reflectivity of the original over its area. The original sharp boundaries may pass through some picture elements, causing them to have intermediate values between the levels on the two sides of the boundary.

It would be desirable to sharpen up the boundary to attain the
Figure 4. Edge Sharpening.
condition shown in Figure 4c. This result is attained by a one-dimensional operator working on the scanned picture. The operator is used twice—once while scanning horizontally, and once vertically.

In Figure 4d, the operation of the one-dimensional edge sharpener is illustrated. For every space quantum, the process ascertains whether the reflectivity is locally monotonic. This will be the case if one of its neighbors is greater than the point in question and the other lower, as for point C in Figure 4d (1). Because of this fact, point C is replaced by its neighbor on the left, as shown in (2). Going on to D, the situation is again monotonic. However, replacing by the left neighbor in consecutive samples is forbidden, so D is replaced by E instead, to produce the sharpened edge in (3).

Consecutive replacement by neighbors on the left must be prohibited to prevent streaking one value of reflectivity for indefinite distances across the picture. Doing so once, however, is necessary to combine triple edges, such as shown, into a single edge. Triple edges are the worst case to be contended with, because for any orientation of a boundary either the vertical or horizontal scan will contain three or fewer edges. In the case of the example of Figure 3, both dimensions have exactly three edges before sharpening.

The Averaging Process

The averaging process used to produce the low frequency component of the picture is fundamental to the coding method. This process, and its close relatives, Intavg and Limavg, are used repeatedly at several points in the procedure, whenever a function of the two space coordinates needs to be smoothed, averaged, or low-pass filtered.
The basic operation of the process called Average is the replacement of the value of the function at each space quantum by the average of the values of the function at the four nearest space quanta. This operation is repeated several times to produce the final result. When forming the low frequency component of the picture, the function is reflectivity.

It is clear that repetitive application of this operation acts to smooth, or average out, the local peculiarities of the function. In order to obtain, at least approximately, a quantitative relation between the number of repetitions and the amount of smoothing, the digital system is approximated by a continuous one which is then analyzed. This derivation is carried out in the following paragraphs.

Suppose the function is called $P$ and the space coordinates are $x$ and $y$. The distance between space quanta is $\Delta x$ and $\Delta y$, respectively. Then the basic operation is represented by

$$P(x, y) \equiv \frac{P(x + \Delta x, y) + P(x - \Delta x, y) + P(x, y + \Delta y) + P(x, y - \Delta y)}{4}.$$ 

If $x$ and $y$ are small, we may write:

$$\frac{\partial P}{\partial x} \approx \frac{P(x + \Delta x, y) - P(x, y)}{\Delta x} = \frac{P(x, y) - P(x - \Delta x, y)}{\Delta x}$$

$$\frac{\partial^2 P}{\partial x^2} \approx \frac{\frac{P(x + \Delta x, y) - P(x, y)}{\Delta x} - \frac{P(x, y) - P(x - \Delta x, y)}{\Delta x}}{\Delta x}$$

$$\approx \frac{P(x + x, y) + P(x - x, y) - 2P(x, y)}{\Delta x^2}.$$
Similarly,

\[ \frac{\partial^2 P}{\partial y^2} \sim \frac{P(x, y + \Delta y) + P(x, y - \Delta y) - 2P(x, y)}{\Delta y^2} \]

And the Laplacian of \( P \) is

\[ L(P) = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \]

\[ \sim \frac{P(x + \Delta x, y) + P(x - \Delta x, y) + P(x, y + \Delta y) + P(x, y - \Delta y) - 4P(x, y)}{\Delta x^2 + \Delta y^2} \]

if \( \Delta x = \Delta y \).

But this is just four times the change in \( P \) in one averaging operation, divided by \( \Delta x^2 \). If \( n \) is the number of averaging operations performed,

\[ \frac{\Delta P}{\Delta n} \sim \frac{\Delta x^2}{4} \left( \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right) \]

If the function is averaged \( \frac{4}{\Delta x} \) times per second,

\[ n = \frac{4}{\Delta x^2} t ; \quad \frac{\Delta n}{\Delta t} = \frac{4}{\Delta x^2} ; \quad \text{and} \quad \frac{\partial P}{\partial t} \sim \frac{\Delta P}{\Delta t} \sim \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \]

Here \( P \) is considered to be a function of \( xy, y, \) and \( t \). If \( \Delta x (= \Delta y) \) is small and \( n \) is large, it is possible to investigate the operation of the digital averaging process by studying the continuous differential equation

\[ \frac{\partial P}{\partial t} = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \].
The Laplacian is known to be symmetric with respect to space directions, so without loss of generality it is possible to consider changes in the x direction, and assume $\frac{\partial P}{\partial y} = 0$. Then

$$\frac{\partial P}{\partial t} = \frac{\partial^2 P}{\partial x^2}.$$

Observe that one solution to this equation is $P = e^{-\omega^2 t} \sin \omega x$ for any $\omega$. This indicates that after time $t$, space frequencies of $\omega$ are reduced by a factor of $e^{-\omega^2 t}$ without phase change.

Consider the operation of changing one function into another by averaging for a fixed time $t$. The Fourier transfer function for this process is $e^{-\omega^2 t}$, where $\omega$ is the space frequency in any direction. This can be written $e^{-(\Delta x^2 t \ n \ \omega^2)}$. As a function of $\omega$, this is the famous "bell shaped curve", and is roughly a low pass filter with bandwidth proportional to $1/n$. A graph of the function is drawn in Figure 5a.

To find the impulse response of this filter, the inverse transform is performed:

Let $b = \frac{\Delta x^2}{4} n$.

\[
P(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-bw^2} e^{iwx} \, dw = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-bw^2} \cos \omega x \, dw
\]

\[
= \frac{1}{\pi} \int_{0}^{\infty} e^{-bw^2} \cos \omega x \, dw = \frac{1}{\pi} \sqrt{\frac{\pi}{2b}} e^{-x^2/4b}
\]

\[
= \frac{1}{\Delta x \sqrt{n \pi}} e^{-x^2/n \Delta x^2}.
\]

*If we allow changes in both dimensions, a similar argument shows that the two-dimensional Fourier transfer function is $e^{-(\omega_x^2 + \omega_y^2) t}$. 


Figure 5. The Averaging Process.
This is again a "bell shaped curve" as a function of \( x \), as shown in Figure 5b. The step response, which indicates what the filter does to an edge in the reflectivity function, is the integral of the impulse response. It is

\[
P(x) = \int_{-\infty}^{\infty} \frac{1}{\Delta x \sqrt{n\pi}} e^{-u^2/\Delta x^2} du = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du
\]

As is well known, this integral cannot be reduced to elementary functions, but it is widely tabulated and is illustrated in Figure 5c.

It is reassuring to verify that this function does indeed satisfy the original differential equation. It is asserted that the function produced by averaging a step function is

\[
P(x, y, t) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi t}} e^{-u^2/4t} du
\]

\[
\frac{\partial P}{\partial x} = \frac{1}{2\sqrt{\pi t}} e^{-x^2/4t} ; \quad \frac{\partial^2 P}{\partial x^2} = -\frac{x}{4t\sqrt{\pi t}} e^{-x^2/4t} = -\frac{x}{4\sqrt{\pi t}(3/2)} e^{-x^2/4t}
\]

\[
\frac{\partial P}{\partial y} = \frac{\partial^2 P}{\partial y^2} = 0
\]

\[
\frac{\partial P}{\partial t} = \frac{1}{\sqrt{\pi t}} \int_{-\infty}^{\infty} e^{-u^2/4t} du = \int_{-\infty}^{\infty} \left( \frac{1}{2\sqrt{\pi t}} e^{-u^2/4t} \right) du
\]
\[ x = \int_{-\infty}^{\infty} \frac{1}{2\sqrt{\pi}} \left( -\frac{1}{2}t - \frac{3}{2} \right) e^{-u^2/4t} + t^{1/2} e^{-u^2/4t} \frac{u^2}{4t^2} \, du \]

\[ = -\frac{1}{4\sqrt{\pi t^{3/2}}} \int_{-\infty}^{x} \left( 1 - \frac{u^2}{2t} \right) e^{-u^2/4t} \, du . \]

It will be true that \( \frac{\partial P}{\partial t} = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \) if

\[ \int_{-\infty}^{x} \left( 1 - \frac{u^2}{2t} \right) e^{-u^2/4t} \, du = x e^{-x^2/4t} . \]

If both sides of this equation are differentiated with respect to \( x \), the result is

\[ (1 - \frac{x^2}{2t}) e^{-x^2/4t} \]

for each side. Hence, they differ by a function of \( t \) only. However, as \( x \) approaches minus infinity, both sides vanish for all \( t \), so this function is identically zero, and equality is proved. This completes the demonstration that the result obtained for averaging a step function does satisfy the original differential equation.

The foregoing analysis produced equations which hold exactly for a continuous system, to which the digital system in question is an approximation. The frequency response of the system illustrates that it is basically a low-pass filter, as claimed. The impulse response shows that a sharp spike, or impulse, is transformed into a wider, shorter, smoother peak. Finally, the output produced by a step input is plausible; the
sharp edge is smoothed as shown in Figure 5c. The equation of this curve was developed with the number of repetitions of the averaging operation as an explicit parameter. This makes it possible to decide how many averagings are necessary to produce the desired effect.

This analysis is used in designing the process which forms the low frequency component of the picture. The original picture is composed of a 100 by 100 array of picture elements. It is decided to transmit the slowly changing part by sampling every fifth point on every fifth row, making $20 \times 20 = 400$ samples. At 5 bits per point, 2000 bits are transmitted for this part of the picture. For comparison, the original picture would require 50,000 bits if transmitted point by point.

It is necessary to average enough so that no big changes occur between samples. In particular, sharp edges should be smoothed out over an interval of at least one sample. It would be desirable to have the .05 and .95 points on the unit rise be at least five picture elements apart. This requirement calls for about seven or eight averagings, according to equation (1). Seven averages are actually used.

There is one pragmatic difficulty in implementing the averaging procedure. If the picture elements are arranged in a perfect checkerboard pattern, half black and half white, every white element is surrounded by black ones and vice versa. Averaging will change white elements to black and black to white, but never mix the two. Thus, there are two different sets of picture elements, those on "even" positions and those on "odd" ones. Averaging mixes the reflectivities within each group, but fails to mix between the groups. To secure this intergroup mixing, the average used in replacing a point contains the point itself as well as
its neighbors. What is done in implementation is to weigh the initial value of a point equally with the average of all its neighbors. This modification is done on the first and last averages only, on the theory that this is sufficient for smoothing and is less time consuming than doing so every time.

This alteration does not basically affect the previous analysis. Averaging in the initial value of each element merely halves that change in its value. Therefore, the same things happen only more slowly. To account for this, the averaging operation is performed one extra time to compensate for the two performances with half the normal effect.

The Boundaries

Detection. The boundaries of the picture consist of interfaces between picture elements and the changes in reflectivity occurring there. They occur between, not in, picture elements, and they begin and end at corners of picture elements. It is expected that one quantum changes will be common, and the hysteresis quantization produces many two quantum changes. So only changes greater than two will be called boundaries. The smaller changes will have to be represented by the low frequency component.

No boundaries are detected in statistical areas since it is their function to avoid boundaries there. However, boundaries between statistical and normal areas are needed. A smoothed version of the reflectivity inside statistical areas is used in determining the existence and magnitude of such boundaries.

Very short sections of boundary add very little to picture quality while adding a good deal to the difficulty of encoding. Therefore, the
next step is to erase all short boundaries. The unit of length for boundaries is the length of one side of a picture element.

Specifically, boundaries of length three or less are removed if they are not attached to any other boundaries, as are sections of boundary of length two or less which are attached on one end to another boundary.

**Tracing.** The boundaries are to be encoded by describing a process for tracing them. A string of bits is to be transmitted from which the receiver can deduce a way to trace the boundaries and determine their magnitude. The things to be transmitted are starting positions, magnitudes of boundaries, directions of travel along the boundary, and some additional information on how later bits are to be interpreted. The final version sent is a piecewise linear approximation to the boundaries, but a set of bits describing point by point tracing of the boundaries is generated as an intermediate step.

For this intermediate step, efficient coding is of negligible importance. One code word is produced for every unit length of boundary, but this code word is divided into seven parts. These parts tell:

1. if a new starting point is required for this segment, or if it continues from where the previous segment ended;
2. the new starting position, if applicable;
3. if there is a branch in the boundary before the current segment begins;
4. the magnitude of the boundary;
5. the encoded magnitude of the boundary;
6. the sign of this magnitude; and,
(7) the direction of the segment from the starting point (up, down, left, or right).

The specification of starting points is expensive in terms of bits transmitted, so this is avoided as much as possible. Once a starting point is established all those boundaries which can be reached from there through a continuous path of boundaries are traced. As they are traced, they are erased from the memory of the encoding machine.

It is possible to trace all connected boundaries by this process: tracing continues point by point until a branch is reached. This condition is then indicated in part (3) of the next word. One of the branches is selected and the position of the branch point is entered into a push-down list. The receiver can make its own list which will be identical to that of the transmitter.

When the trace ends at a dead end, this condition is recorded in part (1). Then a new position is extracted from the push-down list, unless it is empty. If there are no old branch points to which to return, a new starting point is entered in part (2). The receiver, of course, will know whether or not to interpret the next bits it receives as a new position by looking at its copy of the push-down list.

A problem may arise with this system. The position of a branch point may be entered in the list, but the remaining branch may be encoded from the opposite direction before the position is removed from the list. Then the tracer goes back to this point to pick up the other boundary and finds nothing there. If this happens, the tracer searches its records to locate the code word indicating a branch there, and changes that indication. The transmitted signal then will not contain this
problem, and the receiver will be spared this frustration.

Some information transmission can be avoided if the sign of the change in reflectivity does not need to be transmitted. In many cases, the boundaries can be traced with the darker side to the right, for example, thus indirectly indicating the sign of the boundary. However, this policy sometimes interferes with the more valuable objective of tracing an entire group of connected boundaries without new starting positions, and therefore must be compromised.

The settlement reached is to begin tracing from starting points with the darker side on the right, and continue as far as possible. When tracing begins anew from branch points out of the push-down list, a bit is sent to indicate whether the tracing from there will be with the dark side on the right or left. No change in this sign should be necessary until a new starting point is obtained with one of the two methods.

It is more efficient to enter a group of connected boundaries through an isolated end, if one can be found with the dark side on the right. In most cases it is possible to do so. The tracer first encodes all boundaries that can be done in this way. There may remain others, however, so a second pass is made in which the tracer will start at the beginning of any section of boundary, i.e., at any branch point with a branch leading out with the darker side on the right. After this pass, there may remain still others, namely those boundaries without ends or branches, the simple closed curves. On the third and final pass, the tracer will begin even in the middle of a continuous section of boundary and trace out the remaining edges. This completes the tracing proper.
Figure 6 is a flow chart showing the operation of the tracer in detail. The labels Lab 1, Lab 2, etc., in this and subsequent flow charts refer to labels in the computer program in the appendix.

After the three passes, the three lists of code words are sorted so that the original starting points are in order from the top of the picture. Then the magnitude of the boundaries is recoded. It is now coarsely quantized using a hysteresis quantizer, as discussed above. The quantization is logarithmic, as the work of David and Schreibner indicate that it should be. Four different heights are recognized as shown in Table 2. There is a tendency to overestimate the height and thus increase the high frequency component of the picture. Schreibner says that such slight augmentation of the edges produces pictures subjectively preferred by most subjects even to faithful reproduction.

Coding the Boundaries. The code described above is translated into a code designed for efficient transmission. The first word of this code is a position. Next is sent the magnitude of the boundary starting there. Then there is a word describing a line segment from that point by giving the change along both coordinate axes from the beginning to the end of the line segment. The receiver may then draw the boundary there.

At this point in the code, a word appears which merely tells the receiver how to interpret the next bits it will receive. There are four possible messages:

1. The boundary continues from this point with the same magnitude. The next word to be sent will describe the next line segment.

2. The magnitude has changed. The next words give the new magnitude, followed by a line segment.
START

For 3 passes DO:

For each point on the picture DO:

LAB1: Is this an initial starting point?

no

Record branch.

LAB2: Record magnitude and direction of boundary. Go on to next point on the boundary.

yes

no

Is it a branch?

no

Is this the end of the boundary?

LAB4: Are there points in the push-down list?

no

yes

no

Get a point from list and go there. Is there a boundary there?

yes

Record the new sign of the boundary.

no

Erase original indication of branch.

yes

Figure 6. Flow Chart for Tracing Boundaries
Table 2. Encoding Boundary Height

<table>
<thead>
<tr>
<th>Actual Height</th>
<th>Nominal Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4-5</td>
<td>twilight zone</td>
</tr>
<tr>
<td>6-7</td>
<td>8</td>
</tr>
<tr>
<td>8-10</td>
<td>twilight zone</td>
</tr>
<tr>
<td>11-14</td>
<td>16</td>
</tr>
<tr>
<td>15-18</td>
<td>twilight zone</td>
</tr>
<tr>
<td>19-31</td>
<td>32</td>
</tr>
</tbody>
</table>
(3) This is a branch point and should be entered in the push-down list. The next words give a magnitude and a line segment which starts here.

(4) This is the end of this branch of the boundary. A new starting position is required. If the push-down list is empty, the next transmission will be a position. Otherwise, take a point from the list and go from there. In this case, the next word indicates whether the next boundary is to be traced with the darker side on the right or left. The following bits, in either case, give a magnitude and a line segment.

The sign word is only one bit long, of course, and the instruction or decision word described above requires two. The magnitude has four possibilities and so requires two bits. The line segment description was empirically chosen as eight bits, four bits for each dimension. The change in either coordinate in one line segment is an integer between -8 and +7 inclusive.

The starting positions which are transmitted are relative positions—differences between the new position and the position last sent. If the positions are sent in order from the top of the picture, this difference will usually be a small positive number. To convert two dimensional positions to one number, the possible starting points are numbered by scanning the picture. The top row is numbered 0 through 100, the second row 101 through 201, the third 202 through 302, etc. Notice that if there are 100 picture elements per dimension, there are 101 places in each dimension where a boundary may begin (one place before each of the 100 elements and one after the last element).

The difference between consecutive starting points may, in rare
cases, be zero. That is, two boundaries may start from the same place. There is no provision in the code for entering the initial starting point into a push-down list.

If \( n \) bits are used in the position word, the allocation is as follows. The \( n \) bits represent a binary number between 0 and \( 2^n - 1 \) inclusive. The number \( 2^n - 1 \) will be reserved to indicate the end of boundary transmission. After the last boundary has been sent, a new position will be needed, and \( 2^n - 1 \) will be sent.

If the distance from the previous point is less than \( 2^{n-2} \), it will be transmitted and the receiver will have the information it needs. But if the distance is \( 2^{n-2} \) or more, another device is used. First, \( 2^{n-2} \) is transmitted. The receiver interprets this to mean the distance between points is too large to be sent directly, and that the excess of this difference over \( 2^{n-2} \) will be sent as the next \( n \) bit word. Of course, if the distance is as large as \( 2(2^{n-2}) \), the process will have to be repeated with yet another \( n \) bits.

The problem remains of selecting \( n \) to optimize this run-length code. Experience shows that the number of starting points on one picture will be around 100. Since there are about 10,000 points, an obvious approximation is that each point is a starting point with probability \( 0.01 \), independent of the other points. The best choice for \( n \) can then be made with the aid of the following derivation.

Let \( p \) be the probability that any individual point is a starting point. \( q = 1-p \). The probability of a run of length \( x \) or more = \( q^x \).

Then:
<table>
<thead>
<tr>
<th>Code Length</th>
<th>Run Length</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>0 to ((2^n-2)-1)</td>
<td>(1-q^{2^n-2})</td>
</tr>
<tr>
<td>2n</td>
<td>((2^n-2)) to (2(2^n-2)-1)</td>
<td>(q(2^n-2) - q^{2(2^n-2)})</td>
</tr>
<tr>
<td>3n</td>
<td>(2(2^n-2)) to (3(2^n-2)-1)</td>
<td>(q^{2(2^n-2)} - q^{3(2^n-2)})</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The average code length is

\[
\bar{L} = \sum_{j=1}^{\infty} L_j P(L) = \sum_{j=1}^{\infty} j n \left[ q^{(j-1)(2^n-2)} - q^{j(2^n-2)} \right]
\]

\[
= \sum_{j=1}^{\infty} j n q^{(j-1)(2^n-2)} - \sum_{j=1}^{\infty} j n q^{j(2^n-2)}
\]

\[
= \sum_{j=0}^{\infty} (j+1) n q^{j(2^n-2)} - \sum_{j=0}^{\infty} j n q^{j(2^n-2)}
\]

\[
= \sum_{j=0}^{\infty} n q^{j(2^n-2)} = n \sum_{j=0}^{\infty} q^{j(2^n-2)}
\]

\[
= \frac{n}{1-q^{2^n-2}}
\]

If \(q = .99\), trial and error shows that the minimum \(\bar{L}\) is attained for an \(n\) of eight. Therefore, eight bits are used for position words in the boundary encoding.

This completes the description of the code used to send the edges. Figure 7 shows a state diagram illustrating the sequence of code words in
Figure 7. State Diagram for Code Word Sequence in Boundary Encoding.
the boundary encoding.

There are five possible reasons for terminating a line segment during the encoding process:

1. The end of the boundary is reached.
2. A branch is reached.
3. The magnitude changes.
4. The length of the segment exceeds the maximum allowable.
5. The boundary is too crooked to be approximated further by a straight line.

Detection of the last case requires some measure of how straight the boundary is. The one chosen is simple to implement and measures straightness without assuming anything about the slope of the boundary. It is discussed below.

The curve may be viewed as a pair of parametric equations, that is, the two space coordinates, say $x$ and $y$, can be thought of as functions of one independent variable, $t$. Suppose the curve begins at the origin at $t = 0$ and meanders across the $x$-$y$ plane. One can compute two Stieltjes integrals as the curve progresses:

$$
\int_{t=0}^{T} x(t) \, dy(t) \quad \text{and} \quad \int_{t=0}^{T} y(t) \, dx(t)
$$

They represent geometrically the area between the curve and the $y$ and $x$ axes respectively. It is clear that if the curve is a straight line, say $y = kx$, the two integrals are equal. It is also a fact that if the two integrals are equal for all $T$, then the curve must be a straight line. This is demonstrated below.
\[
\int_{t=0}^{T} x(t) \, dy(t) = \int_{t=0}^{T} y(t) \, dx(t) \quad \text{for all } T.
\]

Rewriting as ordinary Riemann integrals,
\[
\int_{0}^{T} x(t) \, Dy(t) \, dt = \int_{0}^{T} y(t) \, Dx(t) \, dt.
\]

Differentiating with respect to \( T \),
\[
x(T) \, Dy(T) = y(T) \, Dx(T)
\]
\[
\frac{Dy(T)}{y(T)} = \frac{Dx(T)}{x(T)}
\]
\[
\int_{a}^{b} \frac{Dy(T)}{y(T)} \, dT = \int_{a}^{b} \frac{Dx(T)}{x(T)} \, dT \quad \text{for any } a \text{ and } b.
\]

Converting to Steiltjes integrals,
\[
\int_{a}^{b} \frac{1}{y(T)} \, dy(T) = \int_{a}^{b} \frac{1}{x(T)} \, dx(T)
\]
\[
\ln y = \ln x + C
\]
\[
y = kx
\]

Thus, equality of the two integrals guarantees straightness of the curve.

The method of piecewise linear approximation is to follow the curve, calculating the two integrals. When they differ significantly, the curve has
be begun to deviate from a straight line, and a new linear segment must be started.

The computation of these two Steiltjes integrals in the digital system is very simple. They reduce to sums

\[ \int x \, dy = \sum x \Delta y \quad \int y \, dx = \sum y \Delta x \]

Of course, \( \Delta x \) and \( \Delta y \) are always \( \pm 1 \), so the integrals can be formed by performing one addition or subtraction for every step along the boundary.

In the digital case, one cannot expect straight lines, but rather a jagged curve made up of horizontal and vertical segments. So the integrals will not be perfectly equal. A question arises as to how much deviation from equality should be tolerated. One answer is illustrated in Figure 8. It requires the curve to stay within one quantum of a straight line. If the curve does so, it will remain inside the cross-hatched region, and the two integrals will differ by no more than the area of this region.

It is a simple problem in geometry to verify that the area is \( x + y - 1 \). In coding the boundaries, the procedure generously allows the integrals to differ by \( |x| + |y| \) before terminating the line segment.

This concludes the discussion of the procedure for encoding the boundaries. A flow chart for this process is shown in Figure 9.

The Statistical Areas

**Detection.** There are several conceivable criteria for deciding what parts of a picture are sufficiently complex to warrant being described statistically. The one chosen selects all those areas of smoothly varying
Figure 8. Margin of Error in Piece-wise Linear Approximation.
Figure 9. Flow Chart for Translating Boundary Trace into Piecewise Linear Code.
reflectivity which are sufficiently small.

Complex areas are composed of many small areas of constant or slowly changing reflectivity. Isolated small areas of constant reflectivity are also detected as being statistical, but since the whole area is one area of constant reflectivity, the mean is that reflectivity and the variance is zero. So no harm is done by including such isolated small areas.

A picture element which differs by more than a certain amount, empirically chosen as two quanta, from at least one of its neighbors is said to be on a boundary. Those elements not on boundaries are said to be interior points because they are inside an area of constant or slowly varying reflectivity.

Areas are considered to be small and therefore selected for statistical treatment if they are so small that they have no interior points. Areas are picked which are composed exclusively of boundary points. A point is included if it is on a boundary and so are all its neighbors. If the point is part of an area with interior points, the test will fail; otherwise it is a statistical point.

**Intavg.** It is now desired to form the mean of the reflectivities in the statistical areas. The mean must be an average, but since the mean can change from place to place, all of the points must not be averaged together. The solution is local averaging like the Average process does. However, it is undesirable to include in the average points outside the statistical areas. Therefore, a modified averaging routine, called Intavg for internal averaging, is used.

Intavg differs from average only in that it disallows use of points outside the statistical areas. Whenever a point on the edge of a statisti-
cal area calls for a point outside the area for use in an average, the value of the point in the area is substituted. In this way the mean is made to depend only on the statistical points.

The Variance. The variance is formed by averaging the square of the difference between individual points and the mean computed as described above. All points outside the statistical areas are assigned a variance of zero. Averaging is again by the same procedure, Intavg. As in the case of the low frequency component, transmission is by sampling. In this case, however, the samples are not uniformly spaced but are chosen to put samples in the statistical areas.

In order to select appropriate points for sampling, all statistical picture elements which border on non-statistical ones are first removed. There remain only the cores of the large statistical regions. It is desired to pick one of these points from each distinct region.

Every point selected represents points in a five by five quantum square centered on it. After selection of a sample point, all other candidates in this square are erased. Every candidate must be represented by such a square.

For implementation, the picture is scanned. Whenever a candidate is encountered, a sample point is selected which will represent this point and as many other candidates as possible. Of course, the point selected may be the one first encountered, if there are no other candidates nearby.

Because of the nature of the variance, it is believed that coarse quantization is permissible. With the reflectivity scale ranging from 0 to 31, four non-zero levels of variance are recognized. The quantization
levels are chosen to give linear quantization to the standard deviation (which is the square root of the variance), as shown in Table 3.

The adaptive sampling method involves selecting a few key points to sample, then transmitting the position of each point and the variance at the point. The points are transmitted in the order they occur as the picture is scanned. The positions are specified by sending the distance between samples, i.e., the number of non-sampled points between samples. This run length coding is the same phenomenon analyzed above in the section on Coding the Boundaries. Since there are about 50 samples out of the 10,000 possible, the average number of bits as given by Equation (2) is minimized for a word length of nine bits.

Decoding the Picture

The Low Frequency Part. The receiver must reconstruct a picture from the bits it receives through the communication channel. The reproduction of the low frequency part is very simple. Each of the sample values received is spread until the entire picture is covered. Then the Average routine is used five times to smooth the values between sample points. The result is used as the low frequency component by the decoding machine.

The Boundaries. By following the tracing directions transmitted, the boundaries can be faithfully reproduced. Figure 10 shows in detail how the decoding is done.

Linear interpolation is used to reproduce the line segments. The procedure followed is a standard linear interpolation routine which never allows a boundary to deviate more than one quantum from the straight line it is following, and allows no errors in locating the end points.
Table 3. Transmitting the Variance

<table>
<thead>
<tr>
<th>Actual Variance</th>
<th>Nominal Variance</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>3-6</td>
<td>4</td>
<td>01</td>
</tr>
<tr>
<td>7-12</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>13-</td>
<td>16</td>
<td>11</td>
</tr>
</tbody>
</table>
Figure 10. Decoding Procedure for Boundaries
Once the boundaries are mapped into the memory of the decoder, a synthetic high frequency component must be generated. These synthetic edges are to be functions which have a discontinuity at the boundary but fall smoothly to zero away from the boundary. The function selected is a negative exponential, because it closely resembles the high frequency components removed by the averaging process and is easy to generate. Thus, if the boundary height is to be 32, the picture element on one side of the boundary becomes 16, and the one on the other side becomes -16. The elements on the far sides of these have values of 8 (and -8), 4, 2, and 1, with zeros elsewhere.

These artificial edges are smoothed by an averaging routine called Limavg, which smooths them without destroying the sharp edges. Four averagings are used.

Limavg (for limited averaging) is the third member of the family of averaging processes. Here, the smoothing action is limited by the major boundaries of the picture, as previously established. The mechanism for stopping information flow across these boundaries is the same used in Intavg.

The Statistical Part. The variance function is first reproduced from the sampled transmission. The samples are extended to cover large areas of the picture, but care is taken to avoid extending them across boundaries. Those parts of the picture which are isolated by boundaries from variance samples or are distant from all samples, are assigned zero variance. Finally, the variance function is smoothed by Limavg four times.

The motive for insuring that no smoothing of the variance across the boundaries takes place is the belief that very detrimental effects
would be produced if a positive variance were created in the non-statistical areas. This would appear in the output as noise. The boundaries are taken as the limiting factor rather than the edges of the statistical areas because the receiver already has these boundaries available when attempting to decode the variance.

It is now desired to produce a function whose value on each picture element is an independent random variable with zero mean and the variance previously calculated. To generate a random variable with zero mean and variance \( n \), the decoder forms the sum of \( n \) identically distributed independent random variables, each having zero mean and a variance of 1. Such a random variable is one which is 1 with probability \( \frac{1}{2} \) and \(-1\) otherwise. This gives an easy way to generate any integral variance from a string of random bits.

Rather than forming pseudo-random numbers in the computer for the simulation, random numbers are fed to the computer as data. One thousand twenty-three random bits representing the evenness or oddness of consecutive digits from the Rand Corporation's *One Million Random Digits* were given to the machine. They were reused as needed. Thus, a stochastic function was produced for the statistical component of the picture.

Reconstructing the Picture. To obtain the decoded picture, these three components are added together. The result is limited in amplitude to the original range of reflectivities, and displayed as the completed output picture.
CHAPTER V

RESULTS

The results of the development of this new method of coding pictures fall into two categories. There are the theoretical results which are products of the design of the code, and the experimental results of testing the code. The former have been discussed in the previous chapter but deserve a summary presentation here.

The primary theoretical innovation in the overall structure of the picture code is the concept of a statistical component in a picture. But for this, the code is a fairly straightforward extension of the synthetic-highs method. This innovation is of particular significance because it attempts to utilize a psychophysical phenomenon which has gone practically unnoticed in previous work. The criterion for separating the statistical part of the picture is not wholly satisfactory, as will be seen later, but the idea of coding part of a picture statistically remains potent.

A procedure of wide possible application which was developed for use in the picture code is hysteresis quantization. This method could be useful in any application where it is desired to trade off accuracy of quantization for a reduction in quantization noise.

The edge sharpening routine is of interest only for picture processing, but it seems to be a valuable preprocessing operation for almost any coding method. It combats a universal problem of spatially quantized
pictures.

The averaging procedure was defined and analyzed. This process is useful in a wide variety of situations calling for smoothing or averaging. It has the virtues of simple implementation and a mathematical description of its effect.

The code used to transmit boundaries is not limited to this picture code, but should be serviceable in any curve encoding application. Intermediate results in developing this code include formulae for optimizing run-length code word sizes and a particularly simple way of doing piecewise linear approximations.

A final theoretical result is the method of adaptive sampling devised for transmitting the variance in the picture code. This method may find application in other similar situations.

In order to evaluate the code experimentally, the four available pictures were processed by the new method. A computer program (reproduced in the appendix) simulated the encoding and decoding of the pictures. Original, reconstructed, and intermediate pictures were produced by the computer output techniques described in Chapter III.

Figure 11 shows the four original pictures and the corresponding decoded ones. It is clear that picture quality is lost in the process. In order to determine the causes of the degradation in image quality, intermediate results of the processing of one of the pictures is shown in Figure 12.

The first picture in Figure 12 is the output of the preprocessors. It seems that the hysteresis smoothing and edge sharpening routines cause little distortion of the picture. That these routines do a great deal,
Figure 11.
After Pre-Processing

Low-Frequency Component

Statistical Areas and Statistical Sample Points

Boundaries Detected

Boundaries Removal of Short Boundary Sections

Decoded Boundaries after Piece-wise Linear Coding

Synthetic High Frequency Component

Low and High Frequency Components without Statistical Part

Figure 12. Intermediate Steps in Processing Picture "B".
however, to reduce the complexity of the picture is demonstrated by Table 4.

The criterion of complexity for pictures was chosen to be the number of interfaces between picture elements of different reflectivity. In a 100 x 100 array of picture elements, there are 19,800 interfaces. From Table 4 it can be seen that around 10,000 of these are between picture elements of different reflectivities in the original picture. Hysteresis smoothing brings this number down to about 8,000, and subsequent edge sharpening reduces the index of complexity by another 2,000. This is a measure of the effectiveness of these preprocessing operations.

Referring back to Figure 12, the second picture, the low frequency component, presents a plausible appearance. The sharp edges have averaged into smooth changes in density.

The next picture shows the statistical areas (in gray), and their sample points (in black). The selection of statistical areas is not ideal, since there are complex areas not included, and yet some needed detail is lost by being included in the stochastic regions. The criterion used for this selection is not able to completely distinguish areas which should be statistical (like the hair) from those which shouldn't (like the eyes and nose). However, the removal of the statistical part does, according to Table 4, reduce the index of complexity from around 6,000 to 4,000. Given the areas, the sample points appear reasonably well placed within them.

The next three pictures show the boundaries at different phases of processing. First are the originally detected boundaries, followed by edges after the removal of short segments. Some loss of detail occurs
<table>
<thead>
<tr>
<th>Picture:</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of boundaries:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>before preprocessing</td>
<td>8,581</td>
<td>10,979</td>
<td>10,637</td>
<td>12,111</td>
</tr>
<tr>
<td>after hysteresis smoothing</td>
<td>6,637</td>
<td>7,740</td>
<td>8,017</td>
<td>9,231</td>
</tr>
<tr>
<td>after edge sharpening</td>
<td>5,036</td>
<td>6,347</td>
<td>6,444</td>
<td>7,176</td>
</tr>
<tr>
<td>after exclusion of statistical areas</td>
<td>2,613</td>
<td>4,156</td>
<td>4,548</td>
<td>3,452</td>
</tr>
</tbody>
</table>
here, of course. Lastly are the decoded edges, including piece-wise linear approximations. These approximations are clearly noticeable. Also shown as intermediate results are the synthetic high frequency component produced from the edges, and the combined low and high frequency part. This last picture needs only the statistical part to be the complete decoded picture, as shown in Figure 11. It looks as if quantization of boundary height causes some additional distortions in the picture.

Table 5 shows the number of bits used in different parts of the code for various pictures. Since there are 10,000 picture elements, the standard technique would require 50,000 bits to specify the picture with five bit accuracy in reflectivity. The new code requires about 10,000 bits total, according to Table 5. Therefore, the information required to send pictures is reduced by about 5:1. Of course the efficiency varies with different pictures.

It is interesting to notice the division of the total of bits among the three components of the picture. There is wide variation here, but on the whole, the low frequency part requires 20 percent of the total, the edge information 70 percent, and the statistical part 10 percent. Within the statistical part, the adaptive sampling method causes the lion's share of the bits to go to specifying the positions of the samples, rather than the values of the variance at these samples. More than half of the 70 percent belonging to edge information is used to specify the line segments themselves. Other large portions go for decision words, starting points and magnitudes, with only a few bits left for the sign of the boundaries.

In this presentation of the results of the new picture code, the
### Table 5. Efficiency of the Code

<table>
<thead>
<tr>
<th>Picture:</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of bits in:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>6,083</td>
<td>9,236</td>
<td>9,611</td>
<td>11,780</td>
</tr>
<tr>
<td>low frequency part</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td>statistical part (total)</td>
<td>1,261</td>
<td>681</td>
<td>620</td>
<td>1,448</td>
</tr>
<tr>
<td>position of variance samples</td>
<td>1,035</td>
<td>567</td>
<td>522</td>
<td>1,188</td>
</tr>
<tr>
<td>sampled variance</td>
<td>226</td>
<td>114</td>
<td>98</td>
<td>260</td>
</tr>
<tr>
<td>boundaries (total)</td>
<td>3,048</td>
<td>6,555</td>
<td>6,991</td>
<td>8,332</td>
</tr>
<tr>
<td>starting points for boundaries</td>
<td>448</td>
<td>880</td>
<td>832</td>
<td>904</td>
</tr>
<tr>
<td>magnitude of boundaries</td>
<td>278</td>
<td>644</td>
<td>694</td>
<td>832</td>
</tr>
<tr>
<td>sign of boundaries</td>
<td>22</td>
<td>51</td>
<td>55</td>
<td>76</td>
</tr>
<tr>
<td>line segments of boundaries</td>
<td>1,840</td>
<td>3,984</td>
<td>4,328</td>
<td>5,216</td>
</tr>
<tr>
<td>decision words for boundaries</td>
<td>460</td>
<td>996</td>
<td>1,082</td>
<td>1,304</td>
</tr>
</tbody>
</table>
theoretical and numerical data are objective facts which must be weighed against a subjective evaluation of picture quality, in order to appraise the worth of the methods described above.
CHAPTER VI

CONCLUSIONS.

The above described coding procedure produces recognizable decoded pictures with about one bit per picture point. Inasmuch as picture quality is not subject to numerical evaluation, it is impossible to express in meaningful terms the exact desirability of a particular compromise between picture quality and efficiency of transmission. In considering efficiency it is interesting to note Pan's experience of achieving greater efficiency with more finely quantized pictures than with more coarsely quantized ones such as the ones used in this research. However, the great complexity of implementing the new code would be a serious liability in most applications even if the picture quality and efficiency were outstanding.

The experiments indicate that the obstacles to obtaining greater picture quality are an inadequate criterion for selecting statistical areas, and the approximations made in encoding the boundaries. Yet, the greatest contribution to the number of bits in the code for a picture comes from the edge information, thus discouraging more accurate specification of the boundaries.

The organization of the new code extends Schreibner's "synthetic-highs" technique into two dimensions, and adds a significant innovation, a statistical component. The component attempts to exploit a heretofore ignored psychophysical property of the human observer, the equation of
essentially random patterns with similar statistics.

Two preprocessing operations were defined which reduce picture complexity with little subjective harm to the image. The hysteresis quantization and edge sharpening routines succeed in making unnoticeable changes in pictures to make them better fit one's expectations. They therefore seem to be useful preprocessors for any picture processor.

Conceptual and mathematical tools for designing picture codes were forged during the development of the new method. These include the averaging process and its analysis, a contour coding method, a procedure for piecewise linear encoding curves, an analysis of run-length codes, and an adaptive sampling process. These results can be intellectual building blocks for other picture codes, and possibly have other applications.

Thus, while the new code is not exciting as a final solution to the difficulties of transmitting pictures efficiently, its development and testing have shed light on some approaches to this problem.
APPENDIX

On the following pages is the computer program used for the simulation of the picture coding method discussed above. The program is written in Extended Algol for the Burroughs B-5500, which is an extended version of ALGOL 60. Comments have been added to the program to help make it self-explanatory. This program requires approximately 15 minutes of processor time and 20 minutes of input/output time to run on the A-5500.
BEGIN %

COMMENT STANLEY W. JOHNSTON %

COMMENT 1 THIS PROGRAM WILL CODE AND DECODE ONE PICTURE %
DEFINE FOR I = FOR 1 + 1 STEP 1 UNTIL 100 DO FOR J + 1 STEP 1 UNTIL 100 DO #
DEFINE FOR A = FOR 1 + 0 STEP 1 UNTIL 101 DO FOR J + 0 STEP 1 UNTIL 101 DO %
DEFINE SUM = [1, J] #
DEFINE E1K = E1 * DIV 1000 + K MOD 1000 # %
DEFINE EPL = E2 * DIV 1000 + L MOD 1000 # %

COMMENT 2 PAPER IS THE LINE PRINTED OUTPUT FILE, PICDATA IS THE INPUT MAGNETIC TAPE FILE, AND CARDS IS THE INPUT PUNCHED CARDS FILE %
FILE OUT PAPER 6 (7, 15) %
SAVE FILE IN PICDATA (1, 1000) %
FILE IN CARDS (3, 10) %
FORMAT IN RAND (90, L1) %
FORMAT OUT NUM ("THE TOTAL NUMBER OF BITS IN THE CODE FOR THIS PICTURE IS", T6, " THE DECODED PICTURE FOLLOWS.") %
FORMAT OUT LIT ("THE ORIGINAL PICTURE NUMBER ", 11; " "; "") %
FORMAT IN INT (11) %
LABLE PO, P1, P2, P3, P4, P5 %
INTEGER H, I, J, X, L, M, N, I1, J1, A, B, I2, I3, I4, I5, I6, I7, I8, L1, J1, J2, J3, X1, Y1, X2, Y2, R0, R1, MAG, BITS %
BOOLEAN F %
INTEGER ARRAY A, B1, D2, D3, Q1, Q2 [0 + 101, 0 + 101], F1, E2 [0, 9, 0 + 101], P1, P2, PK [0 + 101] %
BOOLEAN ARRAY A1 [0 + 101, 0 + 101], G [0 + 100], R [0 + 1022] %
PROCEDURE FULLOUT (A) %
COMMENT 3 THIS FUNCTION PRODUCES A PRINTED OUTPUT OF THE PICTURE %
ARRAY A %
VALUE A %
INTEGER ARRAY A [0, 0] %
BEGIN %
  INTEGER H, I, J %
  INTEGER H, I, J %
  INTEGER H, I, J %
  INTEGER H, I, J %
  INTEGER H, I, J %
  INTEGER H, I, J %
  ALPHA ARRAY C [0, 1, 11] %
  FORMAT OUT ALFA (50 A2) %
FORALL R1 SUR + B2 SUB
END CONTRACT

PROCEDURE AVERAGE (A1; N) \% 
COMMENT 5. THIS PROCEDURE AVERAGES THE PICTURE IN ARRAY A1 N TIMES.
THE ELEMENTS OF THE ARRAY ARE MULTIPLIED BY 8 BEFORE THE OPERATION
AND DIVIDED BY 8 AFTERWARDS TO INCREASE ACCURACY SINCE THE ARRAY ELEM
ENTS ARE INTEGERS \% 
VALUE N \% 
INTEGER N \% 
INTEGER ARRAY A1 [0, 0] \% 
BEGIN \%
INTEGER I, J, 12 \% 
INTEGER ARRAY A? [0, 101, 0, 101] \% 
FORALLX A1 SUR = A1 SUB * 8 J \% 
FOR I2 = 0 STEP 1 UNTIL N DO \%
BEGIN \%
FORALL A1 SUR + IF 12 = 0 OR 12 = N THEN (A1 SUB + A2 SUB) / \%
? ELSE A2 SUB \%
END \%
FORALLX A1 SUB = A1 SUB / 8 \%
END AVERAGE \%

PROCEDURE INTAVG (A1, B, N) \% 
COMMENT 6. THIS IS A MODIFICATION OF AVERAGE, WHICH AVERAGES ONLY WITHIN THE AREAS WHERE THE BOOLEAN ARRAY R1 IS TRUE \% 
VALUE N \% 
INTEGER N \% 
INTEGER ARRAY A1 [0, 0] \% 
BOOLEAN ARRAY B [0, 0] \% 
BEGIN \%
INTEGER I, J, 12 \%
INTEGER ARRAY A2 [0 : 101, 0 : 101] ; %
FORALX A1 SUB + A1 SUB / 8 ; %
FOR I2 = 0 STEP 1 UNTIL N DO
REGIN %
FORALL IF R SUB THEN A1 SUB + IF I2 = 0 OR I2 = N THEN (A1 SUB + A2 SUB) / 2 ELSE A2 SUB
END %
FORALX A1 SUB + A1 SUB / 8
END INTAVG I %
PROCEDURE LINTAVG (A1, N) I %
COMMENT 7 THIS IS A MODIFICATION OF AVERAGE WHICH DOES NOT AVERAGE ACROSS BOUNDARIES. THE BOUNDARIES ARE STORED IN ARRAYS Q1 AND Q2.
Q1[I, J] REPRESENTS THE DIFFERENCE BETWEEN THE PICTURE ELEMENT IN THE I TH ROW AND J TH COLUMN AND ITS NEIGHBOR ON THE RIGHT; Q2 THE NEIGHBOR BELOW I %
VALUE N ; %
INTEGER N ; %
INTEGER ARRAY A1 [0, 0] ; %
REGIN %
INTEGER I, J, I2 ; %
INTEGER ARRAY A2 [0 : 101, 0 : 101] ; %
FORALX A1 SUB + A1 SUB / 8 ; %
FOR I2 = 0 STEP 1 UNTIL N DO
REGIN %
FORALL A1 SUB + IF I2 = 0 OR I2 = N THEN (A1 SUB + A2 SUB) /
2 ELSE A2 SUB
END J %
FORALX A1 SUB + A1 SUB / 8
END LIMAVG I %
WRITE (PAPER I (NOT)) J %
COMMENT 8 THE STATEMENT ABOVE OPENS THE LINE PRINTER FILE. BELOW,
THERANDOM BITS TO BE USED LATER IN THE PROGRAM ARE READ IN FROM CARDS;
READ (CARDS, RAND, FOR K = 0, STEP 1, UNTIL 1022 DO R (K)) J %
COMMENT 9 A CARD IS READ TO DETERMINE WHICH PICTURE TO CODE, THEN T
HE CARD FILE IS CLOSED AND THE CARD READER IS RELEASED TO THE COMPUTER.
THE N TH PICTURE IS TO BE USED. A SPACE STATEMENT POSITIONS THE
INPUT TAPE TO THAT PICTURE J. %
READ (CARDS, INT, N) J %
CLOSE (CARDS, RELEASE) J %
SPACE (PICDATA, 10 X (N - 1)) J %
BEGIN %

LABEL DUMMY1 J %
COMMENT 10 DUMMY LABELS ARE USED TO BREAK THE PROGRAM INTO BLOCKS
OF 51023 WORDS AS REQUIRED BY THE COMPILER. BELOW THE INPUT PICT
URE IS READ AND THE INPUT TAPE FILE LOCKED J %
FOR K = 1, STEP 10, UNTIL 91 DO READ (PICDATA, *, FOR I = K STEP
1 UNTIL K + 9 DO FOR J = 1, STEP 1, UNTIL 100 DO A SUB) J %
LOCK (PICDATA, RELEASE) J %
COMMENT 11 NEXT THE PICTURE, NOW STORED IN ARRAY A, IS LIMITED TO
THE NOMINAL UPPER LIMIT, AND DIVIDED BY 64 TO EFFECT COARSER QUANTI
ZATION. NO Rounding IS DONE SINCE THIS IS INTEGER ARITHMETIC J %
FORALL
BEGIN %

IF A SUB ≤ 2047 THEN A SUB = 2047 J %
A SUB = A SUB DIV 64 J %
END J %
COMMENT 12 THE ORIGINAL PICTURE IS NOW PRINTED OUT WITH A MESSAGE
IDENTIFYING IT \% 
WRITE (PAPER, LIT, N) \% 
FULLOUT (A) \% 

COMMENT 13 THE PICTURE OCCUPIES CELLS 1 THROUGH 100 IN EACH DIMENSION. THE ADDITIONAL CELLS AROUND THE BORDER ARE FOR CONVENIENCE IN DOING SOME OPERATIONS. THEY ARE NOW FILLED \% 
FOR I = 1 STEP 1 UNTIL 100 DO 
BEGIN \% 
   A [0, 0] + A [1, 1] \% 
   A [100, 0] + A [100, 1] \% 
   A [0, 100] + A [1, 100] \% 
END \% 

COMMENT 14 THE PICTURE IS SUBJECTED TO HYSTERESIS QUANTIZATION. THE BOOLEAN ARRAY B1 SERVES TO REMEMBER WHICH PICTURE ELEMENTS HAVE BEEN INCREASED ON THE FORWARD PASS \% 
FOR I = 2 STEP 1 UNTIL 101 DO FOR J = 2 STEP 1 UNTIL 100 DO 
BEGIN \% 
   B1 SUB \% 
   IF B1 SUB THEN A SUB = A SUB + 1 \% 
END \% 
FOR I = 99 STEP -1 UNTIL 0 DO FOR J = 99 STEP -1 UNTIL 1 DO 
IF NOT B1 SUB AND (A SUB + 1 = A [I, J] OR A SUB + 1 = A [I, J + 1]) THEN A SUB = A SUB + 1 \%

COMMENT 15 NOW THE PICTURE IS SUBJECTED TO THE EDGE SHARPENING PROCESS. IN THE HORIZONTAL SHARPENING THE BOOLEAN VARIABLE F RECORDS WHETHER THE PREVIOUS ELEMENT HAS BEEN FILLED FROM THE LEFT. THE BOOLEAN ARRAY G PERFORMS A SIMILAR FUNCTION IN THE VERTICAL SHARPENING \% 
FOR ALL IF (A [I, J + 1] = A SUB) \% 
   (A SUB - A [I, J + 1]) > 0 \% 
   THEN
BEGIN %
A SUB + IF F THEN A (I, J + 1) ELSE A (I, J + 11) %
F + NOT F
END ELSE F + FALSE J %
FOR ALL IF (A (I + 1, J) = A SUB) x (A SUB + A (I + 1, J)) > 0
THEN
BEGIN %
A SUB + IF G (J) THEN A (I + 1, J) ELSE A (I - 1, J) %
G (J) + NOT G (J)
END ELSE G (J) + FALSE J %
COMMENT 16 THE STATISTICAL AREAS ARE NOW DETECTED AND RECORDED IN
BOOLEAN ARRAY B1 J %
FOR ALL R1 SUR + ABS (A SUB - A (I, J + 1)) > 2 OR ABS (A SUB -
A (I + 1, J)) > 2 OR ABS (A SUB - A (I, J - 1)) > 2 OR ABS (A
SUR - A (I - 1, J)) > 2 I %
I %
FOR I = 1 STEP 1 UNTIL 100 DO 81 (I, 1) + B1 (101, 11) + B1 (111 + B1 (1, 0) + FALSE J %
CONTRACT (B1) J %
COMMENT 17 A COPY IS MADE OF THE PICTURE AND THE ORIGINAL IS AVERA
GED INSIDE THE STATISTICAL AREAS I %
FOR ALL D1 SUB + A SUB J %
INTAVG (A, B1) J %
COMMENT 18 THE BOUNDARIES ARE DETECTED AND STORED IN Q1 AND Q2, TH
E CODE IS EXPLAINED IN COMMENT 7 J %
FOR ALL
BEGIN %
Q1 SUB + A SUB - A (I, J + 1) %
IF ABS (Q1 SUR) ≤ 2 THEN Q1 SUB + B %
IF B1 SUR AND B1 (I, J + 1) THEN Q1 SUB + 0 %
Q2 SUB + A (I + 1, J) - A SUB %
IF ABS (Q2 SUR) ≤ 2 THEN Q2 SUR + 0 %
IF B1 SUB AND R1 [I + 1, JJ] THEN Q2 SUB = 0 J %
END J %
FOR I = 0 STEP 1 UNTIL 101 DO Q1 [I, I] = Q1 (I, I) + Q1 [I, I] + Q2 (I, I) + Q2 (I, 0) + Q2 (I, 101) + 0 J %
COMMENT 19 ADDED INTERNAL AVERAGING IS DONE ON A J %
INTAVG (A, R1, 2) J %
COMMENT 20 THE VARIANCE IS FORMED AND AVERAGED. THE STATISTICAL POINTS WITH ZERO VARIANCE ARE REMOVED FROM THE ARRAY OF STATISTICAL POINTS J %
FORALX D1 SUB = (D1 SUB - A SUB) * 2 J %
INTAVG (D1, R1, 5) J %
FORALL IF D1 SUB = 0 THEN R1 SUB = FALSE J %
DUHHY!
END J %
COMMENT 21 THE VARIANCE SAMPLES ARE SELECTED AND PUT IN D2. COUNTS THE NUMBER OF SAMPLES J %
CONTRACT (B1) J %
19 + 0 J %
FORALX D2 SUB = 0 J %
FORALL IF R1 SUB THEN
REGIN%
19 + (19 + 1) J %
FOR II = I + 2 STEP = 1 UNTIL I DO FOR JJ = J + 2 STEP = 1
UNTIL J = 2 DO IF II > 0 AND II < 101 AND JJ > 0 AND JJ < 101 THEN IF B1 [II, JJ] THEN GO TO PO 1 %
PO1 D2 [II, JJ] = D1 [II, JJ] + 1 %
FOR II = II + 2 STEP 1 UNTIL II + 2 = 0 FOR J1 = JJ + 2 STEP 1
UNTIL JJ + 2 DO IF II > 0 AND II < 101 AND J1 > 0 AND J1 < 101 THEN B1 [II, J1] = FALSE
END J %
BITS = 2000 + 2 x 19 J %
COMMENT 22 BITS KEEPS A RECORD OF THE NUMBER OF BITS TO BE TRANSMIT
EO. 2000 ARE REQUIRED FOR THE LOW FREQUENCY PART, AND 2 BITS ARE REQUIRED FOR EVERY VARIANCE SAMPL. Below the variance is quantized. 

FOR ALL D2 SUB + IF D2 SUB = 0 THEN 0 ELSE IF D2 SUB < 3 THEN 1 ELSE IF D2 SUB < 7 THEN 4 ELSE IF D2 SUB < 13 THEN 9 ELSE 16 J %

COMMENT 23 NEXT TO COUNTS THE NUMBER OF POSITION WORDS NEEDED TO SPECIFY THE POSITIONS OF THE VARIANCE SAMPLES. 9 BITS ARE NEEDED FOR EACH, AND THIS IS ADDED TO BITS 1 %

FOR ALL IF D2 SUB # 0 THEN

BEGIN %

18 + 101 × I + J %

10 + (TA - POS) DIV 510 + I %

POS + 1A

END %

BITS = BITS + 9 × 19 %

COMMENT 24 THE ORIGINAL PICTURE IS NOW AVERAGED TO PRODUCE THE LOW FREQUENCY PART. IT IS SAMPLED AND THE SAMPLES USED TO FILL IN THE PICTURE. THEN THIS PICTURE IS AVERAGED TO PRODUCE THE RECONSTRUCTED LOW FREQUENCY PART %

AVERAGE (A, 7) %

FOR II + 3 STEP 5 UNTIL 98 DO FOR JJ + 3 STEP 5 UNTIL 98 DO

BEGIN %

IO + A [II, JJ] %

FOR I + II - 2 STEP 1 UNTIL II + 2 DO FOR J + JJ - 2 STEP 1

UNTIL JJ + 2 DO A SUB + IO %

END %

AVERAGE (A, 7) %

BEGIN %

LABEL DUMMY2 %

COMMENT 25 THE FOLLOWING PART OF THE PROGRAM REMOVES THE SHORT BOUNDARY SEGMENTS. IT SCANS THE PICTURE, STOPING AT EVERY BOUNDARY END IT TRACES THE BOUNDARY FOR THREE STEPS AND ERASES THE WHOLE TH
ING IF THE BOUNDARY TURNS OUT TO BE SHORT, I AND J ARE THE STARTING
POSITION FOR TRACING. I1 AND J1 ARE THE FIRST STEP UP THE BOUNDARY,
I2 AND J2 THE SECOND, ETC. SINCE THE BOUNDARIES END ON CORNERS
OF PICTURE ELEMENTS, THE I,J TH CORNER IS TAKEN TO BE THE LOWER
RIGHT-HAND CORNER OF THE I,J TH PICTURE ELEMENT I X
FOR I = 0 STEP 1 UNTIL 100 DO FOR J = 0 STEP 1 UNTIL 100 DO
BEGIN %
I0 + 0;
IF Q1 SUB # 0 THEN I0 + I0 + 1 I %
IF Q2 SUB # 0 THEN I0 + I0 + 1 I %
IF Q1 [I + 1, J] # 0 THEN I0 + I0 + 1 I %
IF Q2 [I, J + 1] # 0 THEN I0 + I0 + 1 I %
IF I0 = 1 AND Q1 SUB + Q2 SUB + Q1 [I + 1, J] + Q2 [I, J + 1]
= 64 THEN
BEGIN %
I1 + 1 I %
J1 + J I %
IF Q1 SUB # 0 THEN I1 + I1 = 1 ELSE IF Q2 SUB # 0 THEN J1 +
J1 = 1 ELSE IF Q1 [I + 1, J] # 0 THEN I1 + I1 + 1 ELSE J1 +
J1 = 1 I %
I0 + 0 I %
IF Q1 [I1, J1] # 0 THEN I0 + I0 + 1 I %
IF Q2 [I1, J1] # 0 THEN I0 + I0 + 1 I %
IF Q1 [I1 + 1, J1] # 0 THEN I0 + I0 + 1 I %
IF Q2 [I1, J1 + 1] # 0 THEN I0 + I0 + 1 I %
IF I0 # 2 THEN Q1 SUB + Q2 SUB + Q1 [I + 1, J] + Q2 [I, J +
I1 + 0 ELSE
BEGIN %
I2 + I1 I %
J2 + J1 I %
IF Q1 [I1, J1] # 0 AND I # I2 = 1 THEN I2 + I2 = 1 ELSE
IF Q2 [I1, J1] # 0 AND J # J2 = 1 THEN J2 + J2 = 1 ELSE
IF Q1 [I1 + 1, J1] # 0 AND I # I2 + 1 THEN I2 + I2 + 1
ELSE J2 + J2 + 1 J %
10 + 0 J %
IF Q1 (I2, J2) # 0 THEN I0 + I0 + 1 J %
IF Q2 (I2, J2) # 0 THEN I0 + I0 + 1 J %
IF Q1 (I2 - 1, J2) # 0 THEN I0 + I0 + 1 J %
IF Q2 (I2, J2 - 1) # 0 THEN I0 + I0 + 1 J %
IF I0 + 2 THEN Q1 (I1, J1) + Q2 (I1, J1) + Q1 (I1, J1 + 1) + Q2 (I1, J1 + 1) + 0 ELSE
BEGIN %
I3 + I2 J %
J3 + J2 J %
IF Q1 (I2, J2) # 0 AND I1 + I3 - 1 THEN I3 + I3 - 1
ELSE IF Q2 (I2, J2) # 0 AND J1 # J3 - 1 THEN J3 + J3 - 1
ELSE IF Q1 (I2 + 1, J2) # 0 AND J1 # I3 + 1 THEN I3 + 1 ELSE J3 + J3 + 1 J %
I0 + 0 J %
IF Q1 (I3, J3) # 0 THEN I0 + I0 + 1 J %
IF Q2 (I3, J3) # 0 THEN I0 + I0 + 1 J %
IF Q1 (I3 - 1, J3) # 0 THEN I0 + I0 + 1 J %
IF Q2 (I3, J3 + 1) # 0 THEN I0 + I0 + 1 J %
IF I0 + 1 THEN Q1 (I1, J1) + Q2 (I1, J1) + Q1 (I1 + 1, J1) + Q2 (I1 + 1, J1) + Q1 (I2, J2) + Q2 (I2, J2 + 1, J1) + Q1 (I2, J2 + 1) + 0
END
END
END J %
DUMMY2 J %
END J %
BEGIN %
LABEL LAR1, LAR2, LAR3, LAR4, LAR5, LAR6 J %
COMMENTS THE SHORT BOUNDARIES HAVE BEEN REMOVED, NEXT THE BOUND
ARIES ARE TRACED. THE STRING OF WORDS REPRESENTING THE TRACE IS STORED IN \( E_1 \). \( E_{1k} \) REPRESENTS THE \( k \)TH ELEMENT OF \( E_1 \), WHICH MUST BE A TWO-DIMENSIONAL ARRAY BECAUSE OF THE LIMITATION ON THE SIZE OF ANY ONE DIMENSION OF AN ARRAY. IN EACH WORD OF \( E_1 \), THERE ARE THE PARTS DISCUSSED IN THE TEXT, BIT 29 IS THE SIGN, 25 - 29 MAGNITUDE, 30 - 31 DIRECTION, 32 - 45 POSITION, 46 BRANCH, AND 47 INDICATES A NEW STARTING POSITION. THE PUSHDOWN LISTS ARE \( P_k \), \( P_J \), AND \( P_K \), RECORDING RESPECTIVELY THE COORDINATES OF THE POINT TO BE REMEMBERED AND THE OUTPUT WORD (\( E_I \)) IN WHICH THE ENTRY WAS MADE. \( H \) COUNTS THE POSITION OF THE LIST. \( K \) IS THE NUMBER OF THE \( E_I \) WORD. 10 AND 11 ARE MISCELLANEOUS COUNTERS. 12 IS THE NUMBER OF THE PASS (1, 2, OR 3), AND 13 IS THE CURRENT SIGN OR DIRECTION OF TRACING. THE LABELS CORRESPOND TO THOSE IN FIGURE 6 OF THE TEXT.

\[
\begin{align*}
\text{FOR} & \ 1 \ \text{TO} \ 100 \ \text{DO} \ \text{FOR} \ J \ \text{TO} \ 100 \ \text{DO} \\
\text{BEGIN} & \\
E_{1k} & = 1 \ \text{J} \ \% \\
\text{END} & \\
\text{END} & \\
\text{IF} & \ 01 \ \text{SUB} < 0 \ \text{THEN} \ 10 = 10 + 1 \ \text{ELSE} \ \text{IF} \ 01 \ \text{SUB} > 0 \ \text{THEN} \ 11 = 11 + 1 \ \text{J} \ \% \\
\text{IF} & \ 02 \ \text{SUB} < 0 \ \text{THEN} \ 10 = 10 + 1 \ \text{ELSE} \ \text{IF} \ 02 \ \text{SUB} > 0 \ \text{THEN} \ 11 = 11 + 1 \ \text{J} \ \% \\
\text{IF} & \ 01 \ [1, J + 1] > 0 \ \text{THEN} \ 10 = 10 + 1 \ \text{ELSE} \ \text{IF} \ 01 \ [1, J + 1] < 0 \ \text{THEN} \ 11 = 11 + 1 \ \text{J} \ \% \\
\text{IF} & \ 02 \ [1, J + 1] > 0 \ \text{THEN} \ 10 = 10 + 1 \ \text{ELSE} \ \text{IF} \ 02 \ [1, J + 1] < 0 \ \text{THEN} \ 11 = 11 + 1 \ \text{J} \ \% \\
\text{IF} & \ 10 = 1 \ \text{AND} \ 11 = 0 \ \text{AND} \ 12 = 1 \ \text{OR} \ 10 > 0 \ \text{AND} \ (10 \neq 1 \ \text{OR} \ 11 \neq 1) \ \text{AND} \ 12 = 2 \ \text{OR} \ 10 > 1 \ \text{AND} \ 12 = 3 \ \text{THEN} \\
\text{BEGIN} & \\
11 & = 1 \ \text{J} \ \% \\
\text{END} & \\
\end{align*}
\]
JJ + J * K
EIK. (32 + 14) * 101 * I + JJ * K
LAB21: TO + 0 I %
FOR II = - Q1 (II, JJ) * 13, - Q2 (II, JJ) * 13, Q1 (II) +
1, JJ1 * 13, Q2 (II, JJ1) = 13 DO IF II > 0 THEN
BEGIN %
EIK, (30 + 2) + TO J %
E1K, (25 + 5) + JJ J %
GO TO LAB3
END ELSE TO 10 + 10 + 1 J %
LAB3: IF 10 = 0 THEN
BEGIN %
Q1 (II, JJ) + 0 I %
II + II = 1
END ELSE IF 10 = 1 THEN
BEGIN %
Q2 (II, JJ) + 0 I %
JJ + JJ = 1
END ELSE IF 10 = 2 THEN
BEGIN %
Q1 (II + 1, JJ) + 0 I %
II + II + 1
END ELSEF
BEGIN %
Q2 (II, JJ + 1) + 0 J %
JJ + JJ + 1
END I %
K + K + 1 J %
EIK + 0 I %
TO + 0 J %
IF JJ1 + 1, JJ) < 0 OR 13 * Q2 (II, JJ) < 0 OR 13 * Q1 (II) > 0 OR 13 * Q2 (II, JJ + 1) > 0 THEN
BEGIN

10 0 \%
IF 01 [II, JJ] # 0 THEN 10 + 10 + 1 \%
IF 02 [II, JJ] # 0 THEN 10 + 10 + 1 \%
IF 01 [II + 1, JJ] # 0 THEN 10 + 10 + 1 \%
IF 02 [II, JJ + 1] # 0 THEN 10 + 10 + 1 \%
IF 10 > 1 THEN
BEGIN

H + H + 1 \%
PK [H] + K \%
PI [H] + II \%
PJ [H] + JJ \%
E1K, [46 + 1] + 1
END \%
GO TO LAB2
END ELSE
BEGIN

E1K, [47 + 1] + 1 \%
LAB41: IF H = 0 THEN GO TO LAB1 ELSE
BEGIN

II + PI [H] \%
JJ + PJ [H] \%
H + H - 1 \%
I3 + 0 \%
FOR TI = Q1 [II, JJ], - Q2 [II, JJ], Q1 [II + 1, JJ]
1, Q2 [II, JJ + 1] DO IF TI # 0 THEN
BEGIN

I3 + SIGN (II) \%
GO TO LAB5
END \%
LAB5: IF I3 # 0 THEN
BEGIN

E1K, [24 + 1] + IF I3 > 0 THEN 1 ELSE 0 \%
END \%

GO TO LAB2
END ELSE
BEGIN %
E1 [PK [H + 1] DTV 1000, PK [H + 1] 400 1000]
$46 = 1] + 0 ] %
GO TO LAB4
END %
END %
END %
END %
END %
END %
END %
E1K, [37 = 14] + 10201 $ %
K + K + 1 ] %
IF I2 = 1 THEN 15 + K ELSE IF I2 = 2 THEN 16 + K
END %
COMMENT 27 THE E1 ARRAY NOW HAS THREE SEPARATE LISTS END TO END, F
ROM THE THREE PASSES. THE FIRST LIST STARTS AT K = 0, THE SECOND A
T 15, AND THE THIRD AT 16. THE FOLLOWING STATEMENTS SORT THESE 3 P
ARTS AND MERGE THEM SO THAT THE ORIGINAL STARTING POINTS (IN BIT
S $32 - 45$) ARE IN ORDER FROM THE TOP OF THE PICTURE. THE MERGED L
IST IS STORED IN E2, WITH A WORD COUNTER $L$ %
14 + L = 0 ] %
LAB6I K + 14 ] %
19 + E1K, [32 + 14] ] %
K + 15 $ %
18 + E1K, [32 + 14] ] %
K + 16 $ %
17 + E1K, [32 + 14] ] %
1F 19 < 1R AND 19 < 17 THEN
BEGIN %
K + 14 ] %
14 += 1
END ELSE IF IA < IB AND IB < IC THEN BEGIN %
  K = 15
  T5 = -1
END ELSE
BEGIN %
  K = 16
  T6 = -1
END %
IF E1K. [32 : 14] # 10201 THEN BEGIN %
  DO BEGIN %
    E2L = E1K
    L = L + 1
    K = K + 1
  END UNTIL E1K. [32 : 14] # 0 %
  IF T4 = -1 THEN T4 + K ELSE IF T5 = -1 THEN T5 + K ELSE T6 + K %
  GO TO LAR6
END ELSE E2L = E1K
T4 = L - 1
COMMENT 2A THE BOUNDARY HEIGHT IS QUANTIZED NEXT. THE COARSELY QUANTIZED MAGNITUDE IS STORED IN BITS 22 - 23, AND THE PRESENCE OF A TWILIGHT ZONE IS RECORDED IN BIT 21. THEN HYSTERESIS SMOOTHING IS DONE IN BOTH DIRECTIONS. SMOOTHING IS NOT DONE BETWEEN ADJOINING WORDS IF THE BOUNDARY HAS A BRANCH OR A NEW STARTING POINT BETWEEN THEM %
T5 = 0 %
FOR L = 0 STEP 1 UNTIL 14 DO
BEGIN
END I %
15 + 0 I %
FOR L = 14 STEP -1 UNTIL 0 DO
BEGIN
  E2L, [21 : 3] + 15 I %
END I %
END I %
BEGIN %
LABEL S1, S2, S3, S4, S5, S6, S7, S8, S9 I %
COMMENT 29 NOW PIECE-WISE LINEAR CODING BEGINS. THE INPUT IS E2L
AND THE OUTPUT IS PUT IN E1K. LABELS CORRESPOND TO FIGURE 9 IN THE
TEXT.  11 COUNTS THE NUMBER OF POSITION WORDS, 12 MAGNITUDE WORDS,
13 LINE SEGMENT DESCRIPTION WORDS, 14 DECISION WORDS, AND 15 SIGNAL
WORDS.  POS REPRESENTS POSITION AND MAG MAGNITUDE.  X AND Y ARE
DISPLACEMENTS ALONG A LINE SEGMENT, AND XX AND YY ARE TENATIVE VALUES
FOR X AND Y.  1 IS THE DIFFERENCE OF THE TWO INTEGRALS DISCUSSED IN THE
TEXT. 11 + 12 + 13 + 14 + 15 + 0 I %
K + L + POS + X + Y + 1 + 0 I %
K0 + E2L, [32 : 14] I %
S2 = IF 10 = 10201 THEN
BEGIN %
  E1K + 245 I %
K + K + 1 I %
11 + 11 + 1 I %
GO TO S1
END J %
WHILE TO - POS > 254 DO
BEGIN %
     I1 = I1 + 1 J %
     E1K = E1K + 254 %
     K = K + 1 %
     POS = POS + 256
END J %
E1K = TO - POS %
K = K + 1 %
POS = TO %
I1 = I1 + 1 %
S31 NAG = TO * E2L, [22 + 21] %
E1K = TO %
K = K + 1 %
T2 = T2 + 1 %
S41 TO = E2L, [30 + 2] %
XX = XX %
YY = YY %
IF TO = 3 THEN
BEGIN %
     YY = YY + 1 %
     T = T + Y
END ELSE IF TO = 0 THEN
BEGIN %
     XX = XX - 1 %
     T = T - Y
END ELSE IF TO = 1 THEN
BEGIN %
     YY = YY - 1 %
     T = T - Y
END ELSE
BEGIN %
XX + X + 1 j %
I + I + Y
END %
IF XX = 0 OR XX > 7 OR YY = 0 OR YY > 7 OR ABS (I) > ABS (XX) + ABS (YY) THEN
BEGIN %
  E1K + (X + B) x 16 + Y + 8 j %
  K = K + 1 j %
  X + Y + 10 j %
  T3 + T3 + 1 j %
  E1K + 3 j %
  T4 + T4 + 1 j %
  K = K + 1 j %
  GO TO $4 %
END ELSE
BEGIN %
  X = XX j %
  Y = YY j %
END %
L = L + 1 j %
IF E2L. [47 = 1] = 1 THEN
BEGIN %
  E1K + (X + B) x 16 + Y + 8 j %
  K = K + 1 j %
  X + Y + 10 j %
  T3 + T3 + 1 j %
  E1K + 0 j %
  K = K + 1 j %
  T4 + T4 + 1 j %
  T0 = E2L. [32 + 1A] j %
  IF E0 = 0 THEN GO TO $2 %
BEGIN  
     E1K = (X + 8) x 16 + Y + 8 %
     K + K + 1 %
     X + Y + I %
     13 + 13 + 1 %
     E1K + 1 %
     K + K + 1 %
     T + 10 + 1 %
     GO TO S3
END %

IF MAG = E2L. [22 + 2] THEN GO TO S4 %
E1K = (X + 8) x 16 + Y + 8 %
K + K + 1 %
X + Y + I %
13 + 13 + 1 %
E1K + 2 %
K + K + 1 %
T + 10 + 1 %
GO TO S3 %
S11 %

COMMENT 30 NOW THAT THE ENCODING IS DONE, THE BITS REQUIRED ARE COUNTED AND THE TOTAL BIT COUNT FOR THE CODE IS PRINTED %
19 + 8 x 11 + 2 x 12 + 8 x 13 + 2 x 14 + 15 %
BITS + BITS + 10 %
WRITE (PAPER, NUM, BITS) %
END %

COMMENT 31 NOW THE BOUNDARIES WILL BE DECODED, THE LABELS REFER TO
FIGURE 10 IN THE TEXT. DECODED BOUNDARIES ARE STORED IN Q1 AND Q2.

IN THIS PART OF THE PROGRAM, X AND Y ARE DISPLACEMENTS ALONG A LINE SEGMENT, WHILE (XX, YY) IS THE END OF THE LINE SEGMENT. OTHER VARIABLES HAVE SIMILAR MEANINGS TO ABOVE.

K + POS + H + 0 1 %
FORALX Q1 SUP + Q2 SUP + 0 J %
P31 IF E1K = 255 THEN GO TO P5 1 %
P32 POS = POS + E1K 1 %
IF E1K = 254 THEN
BEGIN 1 %
    K + K + 1 J %
GO TO P2
END 1 %
I + POS DIV 101 1 %
J + POS MOD 101 1 %
I3 + 1 1 %
P33 K + K + 1 J %
MAG + 2 * (E1K + 1) 1 %
P41 K + K + 1 J %
10 + E1K 1 %
X + 10 DIV 16 - A 1 %
Y + 10 MOD 16 - A 1 %
11 + SIGN (X) 1 %
12 + SIGN (Y) 1 %
X + AAS (X) 1 %
Y + AAS (Y) 1 %
10 + XX + YY + 0 1 %
WHILE XX # X OR YY # Y DO IF 10 > 0 OR 10 = 0 AND 11 # 0 THEN
BEGIN 1 %
    10 + 10 = Y 1 %
    XX = XX + 1 1 %
    1 + I + 11 1 %

Q1 \( T1 + \text{REAL}(T1 < 0) \times J \) + MAG \( \times T1 \times I3 \)

END ELSE

BEGIN %
    IO = IO + X 1 %
    YY = YY + 1 1 %
    J = J + I2 1 %
Q2 \( T1, J + \text{REAL}(T1 < 0) \times \text{MAG} \times I2 \times I3 \)
END 1 %

K = K + 1 1 %

IF E1K = 3 THEN GO TO P4 J %

IF E1K = 2 THEN GO TO P3 J %

IF E1K = 1 THEN

    BEGIN %
        H = H + 1 %
        PI [H] = I 1 %
        PJ [H] = J 1 %
        GO TO P3

    END 1 %

    IF H = 0 THEN

        BEGIN %
            K = K + 1 1 %
            GO TO P1
        
        END 1 %

        J = PJ [H] 1 %
        J = PJ [H] 1 %
        H = H - 1 1 %
        K = K + 1 1 %
        I3 = IF E1K = 1 THEN 1 ELSE - 1 1 %
        GO TO P3 1 %

P5: 1 %

BEGIN %

LABEL DUMMY3 1 %

COMMENT 32 NOW THAT THE BOUNDARIES HAVE BEEN DECODED, THE SYNTHETI
C HIGH FREQUENCY PART WILL BE PRODUCED AND STORED IN D1
% FORALL D1 SUR + 0, 1%
FORALL
REGIN %
  I1 + 0 %
  TO + Q1 SUR + %
WHILE TO # 0 DO
REGIN %
  IF I > I1 THEN D1 [I - I1, J1] = D1 [I - I1, J1] + TO @ %
  IF I + I1 < 100 THEN D1 [I + I1 + 1, J1] = D1 [I + I1 + 1, J1] + TO @ %
END
END J %
FORALL
REGIN %
  I1 + 0 %
  TO + Q2 SUR + %
WHILE TO # 0 DO
REGIN %
  IF J > J1 THEN D1 [I, J - J1] = D1 [I, J - J1] + TO @ %
  IF J + J1 < 100 THEN D1 [I, J + J1 + 1] = D1 [I, J + J1 + 1] + TO @ %
END
END J %
FOR I = 1 STEP 1 UNTIL 100 DO
REGIN %
  D1 [I0, I] = D1 [I0, I] + D1 [I0, I] %
  D1 [I01, I] = D1 [I01, I] + D1 [I00, I] %
D1 [1, 0] + D1 [1, 1] %
D1 [1, 101] + D1 [1, 100]
END J %
D1 [0, 0] + D1 [0, 101] + D1 [101, 0] + D1 [101, 101] + 0 J %
LIMAVG (D1, 3) %

COMMENT 33 THE VARIANCE WILL NOW BE DECODED. FIRST THE SAMPLES ARE EXTENDED TO COVER THE ORIGINAL AREAS, THEN THE VARIANCE IS AVERAGED %
FOR II + 1 STEP 1 UNTIL 4 DO
BEGIN %
FORALL IF D2 SUB = 0 THEN IF D2 [I, J] = 11 # 0 AND D1 [I, J - 1] = 0 THEN D3 SUB + D2 [I, J - 1] ELSE IF D2 [I, J - 1] # 0 AND D2 [I - 1, J] = 0 THEN D3 SUB + D2 [I - 1, J] ELSE IF D2 [I - 1, J] = 0 AND D2 SUB = 0 THEN D3 SUB + D2 [I, J - 1] ELSE IF D2 [I + 1, J] = 0 AND D2 SUB = 0 THEN D3 SUB + D2 [I + 1, J] ELSE D3 SUB = D2 [I, J]
FORALL IF 02 SUB = 0 THEN 02 SUB = 03 SUB
END J %
LIMAVG (D2, 4) %
COMMENT 34 FOR EACH POINT ON THE PICTURE, THE VARIANCE WILL BE REPLACED BY A RANDOM VARIABLE WITH ZERO MEAN AND THE INDICATED VARIANCE. RANDOM BITS FROM THE R ARRAY ARE USED J %
K = 0 J %
FORALL
BEGIN %
10 + D2 SUB J %
11 + 0 J %
FOR I? + 1 STEP 1 UNTIL 10 DD
BEGIN %
11 + 11 + (IF R [K] THEN 1 ELSE -1) J %
K = IF K = 1022 THEN 0 ELSE K + 1
END J %
D2 SUR + II
COMMENT 35 THE THREE PARTS OF THE PICTURE ARE NOW ADDED. THE RESULT
LIMITED IN AMPLITUDE AND THE FINAL DECODED PICTURE PRINTED OUT.
THIS COMPLETES THE PROGRAM.

FOR ALL
BEGIN

   10 + A SUB + D1 SUB + D2 SUB A

   IF 10 < 0 THEN 10 + 0

   IF 10 > 31 THEN 10 + 31

   A SUB + 10

END

FULLOUT (A)

DUMMY31

END.

END.
LITERATURE CITED


OTHER REFERENCES


