SIMULATION AND PRELIMINARY STUDY OF LOW RESOLUTION PCM AND DPCM PICTURE TRANSMISSION SYSTEMS

by

DONALD PETER ALLAN

B.A.Sc., University of British Columbia, 1968

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

We accept this thesis as conforming to the required standard

Research Supervisor

Members of Committee

Head of Department

Members of the Department of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

September, 1971
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date Sept 17, 1971
ABSTRACT

A monochromatic image can be represented by a real function \( f(x,y,t) \) of two spatial variables and time. Picture transmission research is directed at finding methods of transmitting a facsimile of \( f(x,y,t) \) at minimum cost which still satisfies some quality criteria. The ensemble of source functions, and the measures of cost and quality are defined by the application. For many applications, the subjective quality is an important measure of performance.

Generally, it is difficult if not impossible to find a tractable analytic relationship between the source and system parameters, and the subjective quality. Thus, for picture processing research, equipment must be available for simulating proposed systems so that pictures can be produced and estimates made of the subjective quality.

This thesis will describe a system which was devised for studying low resolution picture transmission systems. The system employs an image dissector to input picture data to a PDP-9 computer. The data can be stored on magnetic tape, processed and then displayed on a picture output device which employs a Tektronix 561 Oscilloscope. A set of general purpose programs was developed for manipulating picture data using this hardware configuration.

Three test pictures were read in and particular
programs were written to calculate some picture statistics. The statistics include signal histograms, auto-covariance functions and, some run length statistics for the two level material. These statistics were then used as design information for some simulation programs.

A program was written to simulate Pulse Code Modulation (PCM) and Differential Pulse Code Modulation (DPCM) systems in order to evaluate the subjective effect of channel errors on pictures transmitted by PCM and DPCM. An attempt was also made to develop a simple two-dimensional filter program to reduce the effect of channel noise. Since the channel noise in the PCM and DPCM systems produced two-dimensional noises which were very different in appearance, two filtering routines were necessary.

Subjective rating tests were conducted to evaluate the filtered and unfiltered PCM and DPCM pictures. For the unfiltered systems, these tests indicate that for good channels, the DPCM system is better than the PCM system while for very bad channels, the pictures produced by the PCM system are subjectively better than those of a DPCM system using the same number of bits. It was found also that the PCM filter proposed, improved the noisy pictures while the proposed DPCM filter produced no improvement and even lowers the quality of some of the pictures.
# TABLE OF CONTENTS

I  THESIS INTRODUCTION

1.1 Introduction.............................................. 1
1.2 Previous Research......................................... 3
1.3 Scope of the Thesis........................................ 5

II  PICTURE PROCESSING HARDWARE

2.1 Introduction.............................................. 7
2.2 Image Dissector Camera.................................... 9
   2.2.1 Image Dissector Tube Operation...................... 9
   2.2.2 Camera Hardware...................................... 11
   2.2.3 Image Dissector Signal to Noise Ratio.............. 13
2.3 Display Hardware for the 561 Oscilloscope.................. 16
   2.3.1 Introduction........................................ 16
   2.3.2 Positioning the Spot................................ 18
   2.3.3 Exposure Control..................................... 21
   2.3.4 Display Summary..................................... 25

III  PICTURE PROCESSING SYSTEM SOFTWARE

3.1 Introduction.............................................. 27
3.2 Hardware Checkout Programs................................ 27
   3.2.1 Camera Programs..................................... 27
   3.2.2 Display Programs.................................... 29
3.3 Processing Software...................................... 31
   3.3.1 Introduction........................................ 31
   3.3.2 Data Input Program IMAGE............................ 33
   3.3.3 Processing Routines................................. 35
   3.3.4 The Display Program................................. 43
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Library Programs</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
<td>47</td>
</tr>
<tr>
<td>IV</td>
<td>STATISTICAL ANALYSIS OF PICTURE DATA</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>48</td>
</tr>
<tr>
<td>4.2</td>
<td>Signal Histograms</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>Signal Auto-Covariance Functions</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Run Length Statistics</td>
<td>56</td>
</tr>
<tr>
<td>4.5</td>
<td>A Line Display Program</td>
<td>62</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>62</td>
</tr>
<tr>
<td>V</td>
<td>DPCM AND PCM SIMULATION PROGRAMS, SUBJECTIVE TEST PROCEDURES AND RESULTS</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>Simulation Programs</td>
<td>64</td>
</tr>
<tr>
<td>5.2.1</td>
<td>PCM and DPCM Simulation Program SYSSIM</td>
<td>64</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Picture Smoothing Programs</td>
<td>71</td>
</tr>
<tr>
<td>5.2.3</td>
<td>DGRADE a Standard Producing Program</td>
<td>76</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Simulation Program Summary</td>
<td>78</td>
</tr>
<tr>
<td>5.3</td>
<td>Subjective Test Procedures</td>
<td>78</td>
</tr>
<tr>
<td>5.4</td>
<td>Subjective Test Data</td>
<td>79</td>
</tr>
<tr>
<td>5.5</td>
<td>Data Presentation and Results</td>
<td>84</td>
</tr>
<tr>
<td>5.6</td>
<td>Subjective Test Summary</td>
<td>89</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>DISPLAY COMMANDS AND CONNECTIONS</td>
<td>90</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>PCM AND DPCM TEST PICTURES</td>
<td>91</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>97</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>I Camera Checkout Programs</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>II Display Checkout Programs</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>III Data Handling System Programs</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>IV Fortran Routines Defined by FPACK</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>V General Purpose Library Routines</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>VI Picture Files</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>VII Programs for Calculating Picture Statistics</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>VIII Run Length Statistics Produced by RLENGH...</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>IX Programs to Produce New Picture Files</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Block diagram of a digital picture transmission system</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Flow chart of picture processing research</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Block diagram of the computer simulation system hardware</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Conceptual diagram of an image dissector tube</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Block diagram of the image dissector camera and its interface</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Average camera noise power and quantization noise versus maximum signal amplitude</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Block diagram of the 561 display</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>Intensities of two points on the CRT face</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>Data file format</td>
<td>36</td>
</tr>
<tr>
<td>10</td>
<td>Simplified flow chart illustrating the components of a processing program</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>Flow chart of program PROCES</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Simplified flow chart of program PROCR</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>Simplified flow chart of program READ</td>
<td>45</td>
</tr>
<tr>
<td>14</td>
<td>Photographs produced on the 561 from data in the three picture files</td>
<td>49</td>
</tr>
<tr>
<td>15</td>
<td>Signal histogram of file HEATHER09</td>
<td>51</td>
</tr>
<tr>
<td>16</td>
<td>Signal histogram of file SCHEMATC6</td>
<td>51</td>
</tr>
<tr>
<td>17</td>
<td>Signal histogram of file TEXTSPC06</td>
<td>52</td>
</tr>
<tr>
<td>18</td>
<td>Auto-covariance of file HEATHER09</td>
<td>54</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Auto-covariance of file SCHEMATC6</td>
<td>54</td>
</tr>
<tr>
<td>20</td>
<td>Auto-covariance of file TEXTSPC06</td>
<td>55</td>
</tr>
<tr>
<td>21</td>
<td>Expanded display of the auto-covariance of file TEXTSPC06</td>
<td>55</td>
</tr>
<tr>
<td>22</td>
<td>Normalized auto-covariance functions for the three picture files</td>
<td>57</td>
</tr>
<tr>
<td>23</td>
<td>Run length histograms for file TEXTSPC06</td>
<td>60</td>
</tr>
<tr>
<td>24</td>
<td>Run length histograms for file SCHEMATC6</td>
<td>61</td>
</tr>
<tr>
<td>25</td>
<td>Block diagrams of systems simulated by the program SYSSIM</td>
<td>65</td>
</tr>
<tr>
<td>26</td>
<td>Transition diagram for a binary symmetric channel</td>
<td>64</td>
</tr>
<tr>
<td>27</td>
<td>Mean-square signal to mean-square error ratio of a PCM system</td>
<td>67</td>
</tr>
<tr>
<td>28</td>
<td>Variation of signal to noise ratio with alpha for a three bit DPCM system</td>
<td>70</td>
</tr>
<tr>
<td>29</td>
<td>Mean-square signal to mean-square error ratio of a DPCM system</td>
<td>67</td>
</tr>
<tr>
<td>30</td>
<td>Data points used by the PCM filter to smooth point X</td>
<td>72</td>
</tr>
<tr>
<td>31</td>
<td>Data points used by the DPCM filter to smooth point X</td>
<td>74</td>
</tr>
<tr>
<td>32</td>
<td>Filter algorithm for the DPCM filtering routine</td>
<td>75</td>
</tr>
<tr>
<td>33</td>
<td>Subjective SNR versus P for a PCM system</td>
<td>81</td>
</tr>
<tr>
<td>34</td>
<td>Subjective SNR versus P for a DPCM system</td>
<td>81</td>
</tr>
<tr>
<td>35</td>
<td>Subjective SNR versus P for a filtered PCM system</td>
<td>82</td>
</tr>
<tr>
<td>Figure</td>
<td>Subjective SNR versus P for a filtered DPCM system</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>36</td>
<td>Subjective iso-quality lines and lines of constant mean-square error for a PCM system</td>
<td>82</td>
</tr>
<tr>
<td>37</td>
<td>Subjective iso-quality lines and lines of constant mean-square error for a DPCM system</td>
<td>83</td>
</tr>
<tr>
<td>38</td>
<td>Subjective iso-quality lines for the DPCM and PCM systems</td>
<td>86</td>
</tr>
<tr>
<td>39</td>
<td>Subjective iso-quality lines for the filtered and unfiltered PCM systems</td>
<td>88</td>
</tr>
<tr>
<td>40</td>
<td>Subjective iso-quality lines for the filtered and unfiltered DPCM systems</td>
<td>88</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

I am grateful to the Defence Research Board of Canada for support received under grant 66-2826 and for support received from the National Research Council of Canada.

I would like to thank my supervisor Dr. R. W. Donaldson for his guidance throughout the project, Dr. J. S. MacDonald for allowing me to use his image dissector camera and Dr. G. B. Anderson for reading a draft of this thesis and commenting.

I wish to acknowledge the assistance provided by Mr. Mike Koombes who designed and constructed the display electronics and Mr. H. Black who provided assistance during the project and in the preparation of this thesis.

I wish to express my appreciation to my mother for typing the final draft of this thesis and to Messrs. K. A. Spencer, G. Austin, and J. Yan for many useful discussions.
I. THESIS INTRODUCTION

1.1. Introduction

A monochromatic image can be represented as a real function \( f(x,y,t) \) of three variables, two spatial variables, \( x \) and \( y \), and time \( t \). The basic problem in picture communication system design is to transmit a facsimile \( \hat{f}(x,y,t) \) of the source image \( f(x,y,t) \) which has satisfactory quality at minimum cost \( 1,2 \). The ensemble of source functions, the cost and the quality measure will be specified by the application. In source coding or bandwidth compression research, the cost is the number of bits required to encode the picture and the quality is the subjective quality.

If only digital systems are considered, the system is usually represented as shown in figure 1. The source encoder maps the signal \( f(x,y,t) \) into a binary vector. The vector may be further encoded by the channel encoder, transmitted and decoded. Viewing the system in this way allows the problem to be segmented. The channel encoding problem is common to all digital transmission schemes and has been considered separately. The source encoding schemes which are dependent on the source materials are considered separately. Picture source encoding schemes try to take advantage of the structure of the video signal \(^3\).

Because of the difficulty of characterizing the
Figure 1 Block diagram of a digital picture transmission system.
source ensemble of pictures and the difficulty of defining an analytic quality measure which accurately reflects the subjective quality of the image, picture processing relies strongly on experimentation. Picture processing research could be considered to follow the flow chart shown in figure - 2. For the purposes of this discussion it is possible to divide source encoding schemes into two classes. The first class includes for single frame encoding two-dimensional techniques such as the Hadamard\(^4\) and Fourier transforms\(^5,6\). The second class consists of one-dimensional methods, where the two-dimensional picture \(f(x,y,t)\) is first transformed into a one-dimensional signal \((v(t))\) using a scanning operation. One-dimensional schemes then try to efficiently encode the resulting time signal. Most operational picture transmission systems are of the one-dimensional type, and further discussion shall be restricted to that type of system.

1.2 Previous Research

Previous workers have analyzed the properties of the video signal \(v(t)\) derived by scanning the input image \(f(x,y,t)\). Others have measured some properties of the functions \(f(x,y,t)\) and the video signal \(v(t)\). Kretzmer\(^9\) used analogue techniques and O'Neal\(^10\) used digital techniques to measure the probability density function (pdf) and auto-correlation function of \(f(x,y)\)
ANALYTIC

Hypothesize a Model of the Picture Random Process

Check the Model

Design the System with Available Data. Optimize W.R.T. some Objective Measure

Change the System

EXPERIMENTAL

Measure Material of Interest

Simulate the Operations the Proposed System Would Produce

Check Objective Measures. Check Results Subjectively to Verify the Performance Measure

Figure -2 Flow chart of picture processing research.
and \( v(t) \) respectively. Run length statistics for video signals derived by scanning two-level images such as text and line drawings have also been measured and published\(^{11,12}\). Frank\(^{13}\) proposed a model for the random video process. Many workers have been active in proposing and testing possible source encoding schemes using simulation\(^{10,14,15,16}\). Research has been conducted on the effects of noise on the video signal\(^{17,18,19,22}\), the relationship between the video noise and the resulting two-dimensional noise in the image\(^{20,19}\), and the effects of two-dimensional noise\(^{21}\). This noise research resulted in the use of weighted noise spectrum as an objective measure of system performance\(^{19,21}\). The application of this measure to optimize the design of a system has also been reported\(^{23}\).

1.3 **Scope of the Thesis**

This thesis will describe a system which was developed for simulating the transmission of low resolution pictures, and will describe some preliminary results obtained using this system. Chapter II discusses the hardware which was used to input and output picture material. Chapter III explains the system of programs which were developed to evaluate the picture input output (I/O) equipment and to facilitate the manipulation of picture data. Chapter IV briefly describes several programs used to calculate picture statistics. Results
obtained when these programs were used to analyze three test files are presented and include signal histograms, auto-covariance functions and some run length statistics. Chapter V describes the simulation of Pulse Code Modulation (PCM) and Differential Pulse Code Modulation (DPCM) systems with noisy channels, the subjective test procedures used to rate the pictures and the results of the subjective tests. The subjective tests resulted in each picture being assigned an equivalent or subjective signal to noise ratio. The subjective test data was plotted as sets of iso-quality curves in the \((N,P)\) plane, where \(N\) is the number of bits per sample and \(P\) is the bit error probability. These curves indicate that DPCM channel noise is more annoying than PCM channel noise.
II. PICTURE PROCESSING HARDWARE

2.1 Introduction

This chapter will briefly describe system hardware and then discuss in detail the picture input and output devices.

Computer simulation is a widely used technique for picture system simulation. Once suitable input and output equipment is constructed, the computer simulation system is very versatile and reproducible. However, the size and speed of present day computers limit the technique to single frame simulation. For real time systems special hardware must still be used for simulation. However, since only single frames were being studied, a computer simulation was satisfactory for this investigation.

The hardware configuration used is shown in figure -3. The computer equipment employed was a PDP-9 computer with 16,000 words of memory, extended arithmetic equipment (EAE), a tape controller and 3 dectape drives, a type 30 graphical display with light pen, a teletype and a general purpose digital interface which comprises a 12 bit analogue to digital (A/D) converter, two 12 bit digital to analogue (D/A) converters, several general purpose input and output registers, and several general purpose flags and control pulses. For picture processing work this system was augmented by a picture input device,
Figure -3 Block diagram of the computer simulation system hardware.
an image dissector camera with interface\textsuperscript{28} and a picture output device which was based on a 561 Tektronix Oscilloscope with appropriate controlling electronics.

Three pictures were mounted in the camera and a digitized version of each picture was read into the computer from the image dissector camera. This raw data consisting of six or nine bit samples was written in a file on one of the dectape units. The data was then used as input for processing programs which read the raw data from tape, processed it and wrote the processed data in a file on another tape unit. A display program could read any of the picture files from the tape and display them on the 561 oscilloscope where they could be photographed.

2.2 Image Dissector Camera

2.2.1 Image Dissector Tube Operation

The operation of the image dissector camera (figure -4) and its interface has been described in a previous thesis\textsuperscript{28}. An image of the input picture is focused onto the photocathode of the image dissector tube. Each point on the photocathode emits a current which is proportional to the intensity of the light falling on it. Electrons from any particular point on the photocathode can be focused onto the aperture and amplified by the multiplier circuit to produce a current at the output of the tube. Thus by controlling the deflection currents it is possible to examine the current
Figure - 4 Conceptual diagram of an image dissector tube.
from any point on the photocathode and in turn the intensity of light at any point on the photocathode.

There are two significant features of this tube. First, the tube may operate at any scan rate. This makes it useful for computer input applications where a fixed scan rate may not be desirable. Second, image dissector tubes can be produced which have excellent spatial resolution by using a small aperture. The tube suffers from one shortcoming in that it has a very low signal to noise ratio. This limitation will be considered later.

2.2.2. Camera Hardware

A block diagram of the complete image dissector camera and its interface is shown in figure -5. Measurements were made to locate sources of error in the system and to determine the camera tube's signal to noise ratio (SNR).

The first source of error found was the poor design of the video pre-amplifier which caused any noise on the power supply lines to appear amplified at the output of the amplifier. This problem was corrected by replacing the pre-amplifier with a NEXUS amplifier. The second defect found was that the value of intensity read into the computer would change with time. Further testing uncovered two sources of this drift, the A/D converter in the camera interface and the change of photocathode sensitivity due to heating of the photocathode
Figure - 5 Block diagram of the image dissector camera and the camera interface.
by the light source\textsuperscript{25,26}. The drift was controlled by using the 12 bit A/D converter in the digital interface and operating the camera with a less intense light source. Some of the measurements were used to estimate the fundamental limit on the camera's performance imposed by the tube's SNR.

2.2.3 Image Dissector Camera Signal to Noise Ratio

Consider the output of the camera when it is focused on the point \((x,y)\). The signal at the output of the low pass filter \(V_s(x,y)\) is\textsuperscript{27}.

\[
V_s(x,y) = S \cdot I(x,y) \cdot A \cdot G \cdot R \cdot K
\]  

(2-1)

where

\(I(x,y)\) = the intensity at point \((x,y)\) on the photocathode
A = area of the dissector aperture
S = Sensitivity of the photocathode
G = current gain of the multiplier circuit
K = voltage gain of the video pre-amplifier
R = the tube load resistance

\(S \cdot A \cdot G \cdot R\) is fixed by the design of the tube and choice of \(R\).

Let \(C_1 = S \cdot A \cdot G \cdot R\) (2-2), then

\[
V_s(x,y) = C_1 \cdot K \cdot I(x,y)
\]  

(2-3)

The signal fluctuates due to the random nature of the electron emission and multiplication processes. The mean-square value of this noise or fluctuation \(V_n^2\) is given by\textsuperscript{27}. 


\[ v_n^2 = 2 \cdot k^2 \cdot s \cdot i \cdot a \cdot w \cdot g^2 \cdot r^2 \cdot k^2 \]  \hspace{1cm} (2-4)

where

- \( W \) = the bandwidth of the low pass filter
- \( k \) = a factor arising from the random nature of the secondary emission process

Let \( C_2 = 2 \cdot k^2 \cdot s \cdot a \cdot g^2 \cdot r^2 \) \hspace{1cm} (2-5), then

\[ v_n^2 = k^2 \cdot c_2 \cdot w \cdot i \]  \hspace{1cm} (2-6)

Thus for a particular value of \( i \) the SNR is given by

\[ \text{SNR} = \frac{v_s^2}{v_n^2} = \frac{k^2 \cdot c_1 \cdot i^2}{k^2 \cdot c_2 \cdot i} = \frac{c_1 \cdot v_s}{c_2 \cdot k \cdot w} \]  \hspace{1cm} (2-7)

This is the SNR for a particular value of \( v_s \). It is necessary to define an average SNR for the whole picture.

\[ \overline{\text{SNR}} = \frac{\overline{v_s^2}}{\overline{v_n^2}} \]  \hspace{1cm} (2-8)

To evaluate (2-8) it was necessary to assume a probability density function (pdf) for the signal \( v_s \). The pdf was assumed to be uniform and is given by

\[ p_{v_s}(x) = \frac{1}{v_m} \quad 0 < x < v_m \]

\[ = 0 \quad v_m > x \]

\[ \overline{v_s^2} = \frac{c_1^2 \cdot k^2 \cdot i^2}{3} \]  \hspace{1cm} (2-9)

\[ \overline{v_n^2} = k^2 \cdot c_2 \cdot w \cdot i_m^2 / 2 \]  \hspace{1cm} (2-10)

From equations (2-3), (2-9) and (2-10) substitute into (2-8) to express \( \overline{\text{SNR}} \) in terms of \( v_m \) as
\[
\text{SNR} = \frac{2}{3} \left( \frac{C_1}{K \cdot C_2} \right) \frac{V_m}{W} \tag{2-11}
\]

From (2-11) it can be seen that the SNR can be increased either by increasing \(V_m\) or by decreasing \(W\). \(V_m\) cannot increase indefinitely without the danger of damaging the tube. For the pre-amplifier gain used it was determined that an output voltage of 2.5 volts is the maximum at which the tube can safely operate. It would appear that the SNR could be raised indefinitely by decreasing \(W\). However, if \(W\) is decreased too much, the time required to input a picture will become impractically long and any reduction in random noise errors will be masked by errors due to drift in the system.

Let \(B = \frac{K \cdot C_2}{C_1} \tag{2-12}\)

Then from (2-7) for a particular \(V_s\)

\[
\text{SNR} = \frac{V_s^2}{2V_n} = \frac{V_s}{B \cdot W} \tag{2-13}
\]

\[
B = \frac{V_n^2}{V_s \cdot W} \tag{2-14}
\]

Measurements were made on the camera to determine the size of the factor \(B\). It was found that

\[B \sim 9.2 \times 10^{-7} \text{ (volts/Hz)} \tag{2-15}\]

With this value of \(B\)
\[ \text{SNR} = \frac{V_m}{3(4.6 \times 10^{-7})W} \]  

The noise was plotted on a graph (figure - 6) which also showed for comparison the quantizing noise produced by uniform quantization to various resolutions. When all factors were considered it was decided to operate the camera with \( V_m \) equal to 2.5 volts and \( W \) equal to 200 Hz. With this operating point ten minutes are required to input a picture containing 256x256 points. It can be seen from the graph in figure - 6 that the resulting SNR should be slightly better than five bit PCM. This noise level places a limit on the type of systems which can be simulated as it is meaningful only to simulate systems whose SNR is much less than that of the I/O equipment.

Once the camera had been checked and modified as previously mentioned, the problem of developing equipment to produce a photographic output from the computer was approached.

2.3 Display Hardware for the 561 Oscilloscope Display

2.3.1 Introduction

To produce digitized pictures, equipment must be available which will produce a matrix of dots on film whose position and exposure can be controlled by the computer. If such equipment exists, desired pictures can be produced by choosing the appropriate dot exposures and co-ordinates. A cathode ray tube (CRT) is a convenient way of realizing this objective since its spot position and
Figure - 6 Camera noise power and quantization noise power versus maximum signal amplitude.
intensity can be controlled quickly and precisely. The display equipment which was developed is shown in figure-7.

2.3.2 Positioning the Spot

The spot deflection voltages are controlled by two Analogic 12 bit D/A converters. Data for these converters is placed in Standard Output Interface (SOI) 31 and then strobed into buffer registers in the respective D/A converters. If $x_c$ and $y_c$ are numbers in the computer which represent the x and y co-ordinates of the desired spot then an ideal display would produce a spot whose co-ordinates $x_d$ and $y_d$ are related to $x_c$ and $y_c$ by equations of the form

$$x_d = k_1 \cdot x_c + d_1$$

$$y_d = k_2 \cdot y_c + d_2$$

It was found for the 561 oscilloscope finally used and for several other CRT displays tested that the actual relationships are more complex functions

$$x_d = f(x_c, y_c, t)$$

$$y_d = g(x_c, y_c, t)$$

where the functions $f$ and $g$ differ enough from the ideal case (2-17) to cause problems. In picture output hardware, where closely spaced parallel lines are being produced, the requirements on equipment are very stringent since an observer can easily detect lines which are not identical and parallel.
Tectronix 561 Oscilloscope

C - 12 Oscilloscope Camera

Half Silvered Mirror

Horz. Amp.

Vert. Amp.

Light Emitting Diodes

PMT

X D/A Converter

Y D/A Converter

Exposure Control Electronics

SOI 31

Load X D/A

Load Y D/A

Intensify

SOI 32

Interface D/A

Flag 56

Digital Interface

(!DX) IOT 6302

(!DY) IOT 5302

(EXPS) IOT 5201

Figure - 7 Block diagram of the 561 display.
There are two possible approaches to the problem. The first is to purchase a high quality CRT display which satisfies equations (2-17). This approach could not be followed. The second method is to determine the factors which caused $f$ and $g$ to differ from the ideal case and reduce these factors to tolerable levels.

After some investigation it was found that $f$ might be a hypothetical function of the form.

$$x_d = k_1 \cdot x_c + k_2 \cdot x_c^2 + k_3 \cdot \sin(2\pi \cdot 60t + \theta) + k_4 \cdot d(t) + d_1$$

(2-20)

The factor $k_2 \cdot x_c^2$ represents the non-linearity which is present in the oscilloscope. The factors which caused most problems were the time varying terms. The sinusoidal term represents spot wobble due to the presence of 60 Hz fields in the vicinity of the display. The function $d(t)$ represents drift in the spot position which was evident if the display was cycled on and off. The 60 Hz fields produced hum bar patterns while the short term drift caused small discontinuities on the picture where whole lines were shifted by a couple of point positions.

After much testing it was found that these problems could be overcome by a combination of hardware and programming techniques. To reduce the hum pickup, the cooling fan of the 561 scope was disconnected and all signal lines to the oscilloscope were shielded. It was then found that if the display program was synchronized with the
60 Hz line, and if the hardware and the program were structured so that the sweep was sufficiently fast, then the hum would be invisible.

To reduce the effect of the drift, it was found necessary to require that the display was not stopped during the display of a picture. These requirements placed several constraints on the display program which will be discussed in the next chapter.

2.3.3 Exposure Control

Once methods had been devised to produce a good matrix of points it was necessary to control the point exposure. The exposure at point \((x_f, y_f)\) is given by

\[ E(x_f, y_f) = \int_0^T I(x_f, y_f, t) \, dt \]  \hspace{1cm} (2-21)

If it is assumed that \(I\) is independent of time then

\[ E(x_f, y_f) = I(x_f, y_f) \cdot T \]  \hspace{1cm} (2-22)

It can be seen that there are two simple ways of controlling the exposure. Either, fix \(T\) and vary \(I\), or fix \(I\) and vary \(T\). The second method was chosen since it was easier to implement.

Initially it would appear that this type of control could be completely open loop thus not requiring any type of light sensor. Unfortunately, the intensity of a spot on the CRT face is not a constant, independent of the spot position. These intensity variations arise due to differences in electron path geometry to different
points on the screen and due to graininess and thickness variations of the phosphor. These variations required a light sensor be constructed. In addition to controlling exposure, a stable light sensor provides an indicator for adjusting the oscilloscope intensity to prevent day to day variations in the pictures.

The system which was developed functions as follows. A number proportional to the exposure is loaded into SOI 32. This produces a voltage at the output of an interface D/A converter, which is supplied to the display electronics. When the intensify command is given, the CRT is unblanked and the light from the CRT spot is picked up by a photomultiplier tube (PMT). The PMT output is integrated until the integral of the photomultiplier voltage equals the reference voltage. When this occurs, the spot is blanked again. It can be seen that the exposure on the film should be proportional to the reference voltage supplied to the display electronics. Once the type of control system was chosen it was necessary to consider what type of phosphor to use and the placement of the photomultiplier tube.

Two points on the screen with steady state intensities of $I_1$ and $I_2$ respectively are to have identical exposures (figure - 8)

$$E_1 = \int_0^{T_1} i_1(t)\,dt + \int_{T_1}^{\infty} i_1(t)\,dt$$  \hspace{1cm} (2-23)
Fig. 8 Intensities of two points on the CRT face.

\[ E_1 = I_1(T_1 + \tau(e^{-T_1/\tau} - 1)) + I_1(1 - e^{-T_1/\tau})\tau \quad (2-24) \]

\[ E_2 = \int_0^{T_2} i_2(t)\,dt + \int_{T_2}^{\infty} i_2(t)\,dt \quad (2-25) \]

\[ E_2 = I_2(T_2 + \tau(e^{-T_2/\tau} - 1)) + I_2(1 - e^{-T_2/\tau})\tau \quad (2-26) \]

The circuitry ensures that the first exposure term is the same in each case.

\[ E_1 = C_1 + I_1(1 - e^{-T_1/\tau})\tau \quad (2-27) \]

\[ E_2 = C_1 + I_2(1 - e^{-T_2/\tau})\tau \quad (2-28) \]

Assume that \( T_1 > 5 \cdot \tau \) and \( T_2 > 5 \cdot \tau \) then

\[ \Delta E = E_1 - E_2 = \tau \cdot \Delta I \]

In general from (2-24) \( E_1 = I_1 \cdot T_1 \)

\[ \frac{\Delta E}{E_1} = \frac{\tau \cdot \Delta I}{T_1 \cdot I_1} = \frac{\tau}{T_1} \left( \frac{\Delta I}{I_1} \right) \quad (2-29) \]
From equation (2-29) it can be seen that a short phosphor time constant reduces error due to intensity variation. A short time constant also allows faster operation since it is not necessary to wait for one spot to decay before another spot is started. A P-16 phosphor was chosen which has a time constant of less than 0.1 usec. Since exposure times were of the order of 100 usec, exposure variations due to differences in intensities will be reduced.

To demonstrate the effect the placement of the PMT has on the field uniformity, consider a point with display co-ordinates \((x_d, y_d)\) which produces a spot on the film at the point \((x_f', y_f')\). The light falling on \((x_f', y_f')\) will be a function of \((x_d, y_d)\) due to the variation in the light's path length and the size of the lens aperture. If this variation is expressed by a function \(T_f(x_d, y_d)\) then the film exposure will be given by

\[
E_f (x_f', y_f') = T_f(x_d, y_d) \cdot I(x_d, y_d) \cdot T \tag{2-30}
\]

However, with the PMT, the intensity is measured from another position and there will generally be another transmittance function \(T_t(x_d, y_d)\)

\[
E_t = T_t(x_d, y_d) \cdot I(x_d, y_d) \cdot T \tag{2-31}
\]

To produce a uniform exposure over the entire display area, \(E_t\) will be fixed and the exposure time \(T\) is given by

\[
T = \frac{C}{T_t(x_d, y_d) \cdot I(x_d, y_d)} \tag{2-32}
\]
The film exposure then becomes

\[ E_f(x_f, y_f) = \frac{T_f(x_d, y_d) \cdot C}{T_t(x_d, y_d)} \]  

(2-33)

Thus it can be seen that the exposure will be uniform over the display area only if the two transmittance functions \( T_f \) and \( T_t \) are related by a constant.

\[ T_f = K \cdot T_t \]  

(2-34)

When the display was initially designed \( T_f \) was unknown and \( T_t \) was made constant by placing the PMT a long distance from the CRT face. It later became evident that \( T_f \) is not constant and some shading in the photographs was observed and attributed to this cause. In a more precise design, this factor would have to be considered.

2.3.4 Display Summary

A display was developed based on a Tektronix 561 Oscilloscope. The display employs a CRT with a P-16 phosphor and some control electronics which comprise two deflection D/A converters and exposure control electronics. Some design considerations concerning the choice of phosphor and the positioning of the PMT were presented. Due to the oscilloscope having some defects it was found necessary to impose the following constraints on the display program. The display must be synchronized with the 60 Hz. line, the sweep speed of the display must be fairly high and the display must run continuously until the picture is completed.
The commands used by the computer programs to control the display are shown in appendix - A. Developed concurrently with the display was a system of programs for processing picture data and for evaluating and adjusting the hardware.
III. PICTURE PROCESSING SYSTEM SOFTWARE

3.1 Introduction

In the development of the picture processing system, numerous programs were written. The programs which evolved can be placed in three categories:

(1) hardware checkout programs, written specifically for the evaluation and adjustment of the camera and display;
(2) data manipulation programs, which are the core of the picture processing software system and include programs for reading, manipulation and displaying picture data; and
(3) library programs, a set of utility programs which were written to perform tasks which occur frequently in the other programs. The library programs include a random number program, and a general purpose graphical display program.

3.2 Hardware Checkout Programs

The hardware programs can be subdivided into camera checkout and display checkout programs.

3.2.1 Camera Programs

All input from the camera is controlled by a subroutine called NPOINT. The calling program supplies this sub-program with the co-ordinates of the point to be read and the subroutine returns with the data in the accumulator. The program is versatile and allows the operator to read six bit samples from the camera's own A/D converter or six or nine bit samples from the interface.
A/D converter. In addition, the program delay can be varied so that the subroutine can operate with any low-pass filter bandwidth. Three programs which use this routine are listed in table - I.

**TABLE I.**

**Camera Checkout Programs**

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOISE</td>
<td>To compile a signal and noise histogram with data acquired by scanning a 3600 point array 10 times.</td>
</tr>
<tr>
<td>AERG</td>
<td>To check the uniformity of the light source by measuring the average intensity in 64 equal area squares which partition the image area.</td>
</tr>
<tr>
<td>LINE</td>
<td>A program for inputing a line of data from the camera and displaying it on the type 30 display.</td>
</tr>
</tbody>
</table>

All these programs are loaded with a command decoder, which allows any program to be run simply by typing a five letter command on the teletype.

NOISE was the first camera checkout program written. It reads and stores an array of 3600 points from the camera. It then reads the same array again, updates a signal histogram, finds the difference between the present value and the stored value of each sample and stores this difference value in a noise histogram. This sequence is normally repeated ten times. After compiling the histograms, the program can display the data in three ways. First, it can output on the teletype a table of signal and noise histogram values.
Second, the histograms can be displayed on the type 30 display. Third, a small Fortran subroutine can be used to calculate the signal mean and the noise variance. This program was useful for checking the noise magnitude and signal drift when evaluating the system and is also used for adjusting the light intensity when operating.

AERG is a program for checking the uniformity of the light source. The program reads the intensity at 1024 points in each of 64 equal squares which completely covered the image area. The program then prints a table of the 64 average intensities from which it is possible to determine the light distribution.

LINE is a program for reading one horizontal line of data and displaying it on the type 30 display. It is useful for verifying that the camera is operating correctly and in particular is used for checking the camera focus when a razor blade is used as the input image.

3.2.2 Display Programs

The programs which were written for display checkout are listed in table II.

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXPO</td>
<td>To generate a table of test exposures and display these exposures with the type 30 display.</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>FAST1</td>
<td>To display the exposure table from NEXPO on the 561 display using a display format of 256&lt;sup&gt;2&lt;/sup&gt; points or 512&lt;sup&gt;2&lt;/sup&gt; points.</td>
</tr>
<tr>
<td>NFRAME</td>
<td>To display the outline of a square on the 561 display for the purpose of adjusting the size and position of the display image and the focus of the 561 oscilloscope.</td>
</tr>
<tr>
<td>XTEST</td>
<td>To generate a raster of points whose scan format can be changed and whose exposure is varying slowly with time for use in checking and adjusting the exposure control hardware and the 561 spot intensity.</td>
</tr>
</tbody>
</table>

The first two programs NEXPO and FAST1 are used for generating exposure test patterns. These patterns consist of either eight equal vertical bars with different exposures or 64 squares of equal area, but different exposures. The test patterns serve two purposes. First, the exposure is checked and second, the patterns can be examined for display defects such as hum and drift.

NFRAME is a short program which generates a square whose size is that of the pictures to be displayed. It is used primarily for focusing, positioning and adjusting the size of the image of the 561 oscilloscope. To maintain size uniformity, tape markers were put on the face of the scope. At the start of each new session the program NFRAME is run and the size and position of the image are adjusted until the square is aligned with these markings.

XTEST is a program which generates points whose exposure varies slowly with time. By examining various voltages with an oscilloscope, the operation of the exposure electronics can be checked and adjusted.
This program is also used to adjust the 56l intensity. For this task an oscilloscope is connected to the output of the photomultiplier amplifier. The light emitting diodes (LED) are turned on to verify that the PMT is operating and that its gain is correct. The LEDs are then turned off and the oscilloscope intensity is turned up until it reaches a specified level. It is worth noting that at the intensity levels used for photography, the spot is almost invisible to the eye. The reason for operating at this low level is to ensure the screen is not burned and to preserve the spot focus which is inversely proportional to the spot intensity. These three programs are used to check and adjust the display and its peripheral electronics.

In addition to these programs which are essential for the operation of the hardware it was necessary to write programs to process picture data. The set of programs which evolved after considerable programming effort allows the processing of picture data quickly and easily.

3.3 Processing Software

3.3.1 Introduction

When considering processing picture arrays on a small computer, several characteristics of the small computer must be kept in mind. First, on a small computer, floating point arithmetic is more than an order
of magnitude slower than fixed point arithmetic. This difference is important for picture processing operations which require extensive calculations. Thus, all the processing programs written, use fixed point arithmetic. Fixed point arithmetic requires care in order to reduce round off errors but the execution time saving makes these efforts worthwhile. Second, the core memory of a small computer is too small to hold a complete picture array. Consider as an example a $256^2$ point picture represented by six bit samples which are packed three samples per word. Such a picture would require 21,844 words of storage which exceeds the PDP-9 memory capacity of 16,000 words. This small memory size requires that a bulk storage device such as a magnetic tape or disk memory be used. In this case dectape is used as the bulk storage device. The sequential nature of a tape file lends itself to simulating systems which are sequential in nature. Systems which transmit a picture by scanning and then transmitting the resulting video signal by PCM or DPCM are of this type. It would not, however, be easy to simulate some two-dimensional schemes such as those which employ two-dimensional transforms. A third minor consideration is that small computers do not have available powerful libraries of mathematical routines which are available with large computing machines such as an IBM 360-67.
The software developed allows the simulation of sequential systems such as PCM and two-dimensional operations which require data from up to sixteen consecutive horizontal lines. The set of processing programs is listed in Table III.

### TABLE III. Data Handling System Programs

<table>
<thead>
<tr>
<th>NAME</th>
<th>DATA SOURCE</th>
<th>DATA DESTINATION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE</td>
<td>Camera</td>
<td>Dectape</td>
<td>To input picture files from the camera.</td>
</tr>
<tr>
<td>PROCES</td>
<td>Dectape</td>
<td>Dectape</td>
<td>To input data from tape to supply it to processing subroutines and then to transfer processed data to a new file.</td>
</tr>
<tr>
<td>PROCR</td>
<td>Dectape</td>
<td>Dectape</td>
<td>A more general program than PROCES for unpacking data for processing and for packing and storing the results.</td>
</tr>
<tr>
<td>READT</td>
<td>Dectape</td>
<td>Film</td>
<td>To display a picture on the 561 display for photographing.</td>
</tr>
<tr>
<td>PHOTOR</td>
<td>Dectape</td>
<td>PROCR</td>
<td>Defines a set of Fortran subroutines so that picture data can be processed with subroutines written in Fortran.</td>
</tr>
<tr>
<td>FPACK</td>
<td>PROCR</td>
<td>PROCR</td>
<td></td>
</tr>
</tbody>
</table>

Each program will be described in more detail in the following text.

#### 3.3.2 Data Input Program IMAGE

IMAGE is the program used to read data from the
camera and store it on tape. The program uses the subroutine NPOINT to input data from the camera. With IMAGE the operator has the choice of A/D converters provided by NPOINT and the choice of storing six bit data samples, packed three per word or nine bit samples packed two per eighteen bit word. Also he can choose the scan format of the picture. In other words he can specify the number of points in the x and y directions and their spacing. For the work to be reported, picture files consist of $256^2$ point arrays of data quantized to six or nine bits.

The picture data is stored on dectape. Dectape as used here is formatted into 576 blocks of 256 words. It was decided to use DEC supplied tape handling programs and to use the file oriented software. When file oriented software is used, eight blocks of the tape are used for a directory which contains tables which include a nine character name for each file and the location of the file on tape. A single dectape can store four nine bit pictures or five six bit pictures.

When a user loads the image program he must specify to the program the location and size of an area of core memory which can be used as a data buffer. When the user has specified the scan format and the number of bits per sample, the program calculates the size of two data buffers which are nearly equal in size to each other which hold integral numbers of scan lines and which will fit into the buffer area allowed. The reason for splitting
the buffer area in two will become evident when the display program is considered. In what follows these two buffers are referred as Bl and B2 respectively. A picture file will consist of a series of alternate segments composed of data first from buffer Bl and then from buffer B2. The first segment is preceded by a header area which contains the scan parameters. The other data segments are separated by smaller headers which contain some control data. The last segment is adjusted to include any lines which do not fit exactly into buffers of size Bl or B2. The complete file format is shown in figure -9. The checksum with each segment is the modulo two sum of all the data in that buffer. It is included to ensure that no data errors are made by the tape drives. Once the data is written with the desired format, it can be processed or displayed.

3.3.3 Processing Routines

Any processing program for this system will have the general structure shown in figure -10. It can be seen that except for the actual processing subroutine, each processing routine has the same format. Thus it was decided to write a general purpose routine which would handle the tasks which are common to all processing, namely, fetching data for processing and storing processed picture files for display. Two such programs were written.

A simplified flow chart for the first routine PROCES is shown in figure -11. This program is suitable
Figure - 9 Data file format
Figure - 10 Simplified flow chart illustrating the components of a processing program.
Get File Name
Clear Write Flag

Writing?

Yes
Set Write
FLAG

No

Initialize Process
Subroutine

Read a buffer of
Data

Unpack a
Data Sample

Write Flag
Set?

No

Yes

Pack and Store
the Processed Data

Buffer
Exhausted?

No

Yes

Write the buffer of
Processed Data if
the Write Flag is set

Data
Exhausted?

No

Yes

Close the
File

Return to
Command Decoder

Initialize the
Routine for
Processing

Processing
Subroutine

Figure - 11 Flow chart of program PROCES
for simulating DPCM or PCM systems. The user types in the name of the picture file which is the data source and indicates whether a new file of processed data is to be written. The program reads a buffer of data from the file, unpacks the data and exits to the processing subroutine with a sample value in the accumulator. On return from the subroutine, PROCES expects the accumulator to contain the output of the simulated system. If the user has specified a new file, this data is packed and written on tape. If the user has not specified a new file this data is ignored. In either case another data value is supplied to the processing subroutine until the input file is exhausted. The mode of operation where no file is specified is useful for routines which calculated signal statistics.

PROCES is useful for simulating systems whose output depends on only a small set of past and present input signal values such as the output from a PCM or DPCM system. However, it is too restrictive for simulating two-dimensional operations where the output often depends on past, present and future lines.

To remove these restrictions a more general and complex program called PROCR was written. The program has the components shown in figure -12. Each time a processing subroutine enters routine FETCH, it is returned with a new data value in the accumulator. Each time a routine calls STORE with a data value in the accumulator,
Figure -12 Simplified flow chart of program PROCR.
this value is packed and written on tape.

A program called FPACK was written for use with PROCR which defines two Fortran subroutines and a function subprogram. The Fortran routines are listed in table IV.

TABLE IV.

Fortran Routines Defined by FPACK

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FETCHL(I,N)</td>
<td>A subroutine to read N lines of data into the data buffer starting with line number I.</td>
</tr>
<tr>
<td>IPICT(I,J)</td>
<td>Function subprogram which will fetch the value of the picture function at row I, column J, provided that row I has been read into the data buffer using the FETCHL subroutine. The function defined is a Fortran integer variable.</td>
</tr>
<tr>
<td>PACK(IV)</td>
<td>A subroutine which packs and stores a processed point in the output file. The output values must be presented to the PACK subroutine sequentially.</td>
</tr>
</tbody>
</table>

The FETCHL routine is restricted to calling lines sequentially into a data buffer. The last sixteen lines called are held in core and are accessible to the program IPICT. Fortran's increased execution time was felt to be justified in view of the simplicity of programming.

To illustrate the simplicity and power of these routines consider the following example. Assume it is necessary to replace each picture point by an average of itself and its eight nearest neighbours. This task could be accomplished with the following Fortran subroutine.
(note- for this example the effect of the picture edges was ignored)

C
C ENTER THE PROGRAM AND PRIME THE DATA BUFFER
C
SUBROUTINE FILTER
CALL FETCHL (1,8)

C
C DO THE CALCULATION FOR ALL POINTS
C
DO 20 I=1,256
  DO 30 J=1,256
    IV=0
    DO 10 K=1,3
      DO 10 L=1,3
        IV = IV + IPICT(I+K-2,J+L-2)
      10 CONTINUE
    IV = IV/9
  30 CONTINUE

C
C STORE THE NEW OUTPUT
C
CALL PACK (IV)

C GET A NEW LINE IF ALL DATA HAS NOT BEEN READ
C
LN= I + 8
LIF(LN.GT.256) GO TO 20
CALL FETCHL (LN,1)

20 CONTINUE
RETURN
END

One can see from the example subroutine that the Fortran routines defined by FRACK greatly simplify the writing of data processing subroutines.

Both PROCES and PROCR handle six and nine bit files and write files whose format is identical to those written by IMAGE. Thus the output files from the processing programs can be displayed or used as input files for further processing.
3.3.4 The Display Program

A program for displaying the picture files is the final data handling program to be considered. The display program was subject to several constraints which were mentioned in Chapter II. These were: the display must be synchronized with the 60 cycle line; a high speed must be maintained and the display must operate continuously.

The program for displaying data was split into two parts. The main routine called READ reads data from tape and supplies it to the second part of the program called PHOTOR the actual display subroutine. The first two constraints were satisfied easily. PHOTOR accumulates a buffer containing a complete line of picture data before displaying and then uses the computer's real time clock to synchronize with the 60 Hz. line. Each line is then displayed several times quickly in succession rather than once slowly in order to maintain a high sweep speed.

The third constraint, that of running the display continuously was harder to realize. As stated earlier, it is impossible to store a complete picture in the computer's memory before displaying. Thus it was necessary to interleave the reading of data from the tape and the display of data. To accomplish this it was first necessary to modify one of the dectape handlers (.DTD version B) in order that it would allow interleaving.
Next the program READ whose flow chart is shown in figure -13 was written in order to implement this interleaving.

Two buffers are set up in core. Before the display is started, the first buffer B1 is filled with data and the data pointer set to B1. The program then starts displaying data from buffer B1 and reading from dectape into buffer B2. When B1 is exhausted, the program checks to see if B2 is full. When it is full, the program swaps the read and data pointers and displays data from buffer B2 while reading into buffer B1. The program continues swapping buffers in this fashion until the entire picture has been displayed.

The output format used for all pictures was $256^2$ points. The bottom twelve lines of the picture were used as an eight level gray scale in order to maintain a constant check on the display operation. All pictures were taken with type 47 Polaroid film.

Three types of data handling programs have been described: the data input program IMAGE; the data processing routines PROCES and PROCR and the data output routines READ and PHOTOR.

3.4 **Library Programs**

The last group of programs listed in table V are the library programs which are used by all the previous programs.
Figure - 13 Simplified flow chart of program READ
TABLE V.
General Purpose Library Routines.

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAPH</td>
<td>Displays a buffer of data graphically on the type 30 display. Features lightpen control for expanding and moving the display. Uses the program .LTA.</td>
</tr>
<tr>
<td>FRAND</td>
<td>Routine for generating random numbers with a uniform probability distribution.</td>
</tr>
<tr>
<td>INPUTR</td>
<td>General purpose octal and decimal number input and output routine.</td>
</tr>
<tr>
<td>BUFOUT</td>
<td>Routine for printing a buffer of data in tabular form on the teletype.</td>
</tr>
<tr>
<td>CORDR</td>
<td>Re-entrant program for calculating coordinates of a point in a raster scan.</td>
</tr>
<tr>
<td>DTAPE</td>
<td>Short tape handling program works in conjunction with Dec handlers .DTA or .DTD; simplifies the writing, reading, deletion and renaming of picture files.</td>
</tr>
</tbody>
</table>

Two of the programs FRAND and GRAPH are worthy of note. FRAND is a copy of the random number routine used on the IBM 360. It employs the Lehmers multiplicative congruence method for generating a sequence of 17 bit numbers which have a uniform probability density function and a period of $2^{32}$.

GRAPH is a general purpose graphical display program which when used in conjunction with display handler .LTA allows either a linear or semi-logarithmic display of a buffer of data. GRAPH was used extensively in all the processing routines for checking the operation of the programs and for gaining insight into the operation.
of the simulated systems. This program features light pen control for controlling the display. Several examples of displays generated using this program will be presented in the next chapter.

3.5 Conclusion

This chapter has described general purpose programs which evolved for picture processing and equipment development. These programs comprised hardware check-out programs, data handling programs and library programs. The next chapter will describe particular routines which were evolved to calculate signal statistics. Results obtained when the programs were applied to picture files will also be presented.
CHAPTER IV.
Statistical Analysis of Picture Data

4.1 Introduction

To test the system, three pictures were read into the computer and stored on tape. The pictures represented the type of material which might be transmitted on a facsimile system. They consist of a picture of a face, a sample of text and a section from a circuit schematic. The files are listed in table VI. and displayed in figure -14. The results to be presented in this chapter will consist of the output which resulted from analyzing these three files.

TABLE VI.

<table>
<thead>
<tr>
<th>FILE NAME</th>
<th>NO. BITS</th>
<th>NO. POINTS</th>
<th>TYPE OF MATERIAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEATHER09</td>
<td>9</td>
<td>$256^2$</td>
<td>Picture of a face</td>
</tr>
<tr>
<td>TEXTSPC06</td>
<td>6</td>
<td>$256^2$</td>
<td>Picture of text</td>
</tr>
<tr>
<td>SCHEMATC6</td>
<td>6</td>
<td>$256^2$</td>
<td>Picture of a section of a schematic</td>
</tr>
</tbody>
</table>

The programs which calculated picture statistics from the data, operate in conjunction with the data handling routine PROCES and do not produce any new pictures. The routines are listed in table VII.

TABLE VII.

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIS</td>
<td>To compile a signal value histogram</td>
</tr>
<tr>
<td>DLINE</td>
<td>To display a line of data from a picture file</td>
</tr>
</tbody>
</table>
Figure - 14 Photographs produced on the 561 from data in the three picture files.
AUTO To calculate the auto-correlation function of a picture file.

COVAR To calculate the auto-covariance function of a picture file.

RLENGH To compile runlength histograms and calculate some statistics.

4.2 Signal Histograms

The program PHIS compiles a signal histogram from the picture data. The program splits the signal range into 64 equal parts. A signal value which falls within one of these 64 ranges is assigned to that particular range. For six bit files there are only 64 possible signal values. Therefore, each signal value has its own range in the histogram. For nine bit files there are 512 possible signal values. Thus each histogram range contains eight signal values or in other words, only the six most significant bits are used in compiling the histogram of a nine bit file. The program can output the histogram in tabular form on the teletype or as a graph on the type 30 display by using the subroutine GRAPH. Photographs of histogram graphs showing log frequency plotted versus signal range are shown for the three data files in figures 15, 16 and 17. The histogram of the face file is fairly uniform as would be expected. The histograms of the picture of the text and schematic are more peaked. The logarithmic nature of the display de-emphasizes the sharpness of these peaks.
Figure - 15 Signal histogram of file HEATHER09.

Figure - 16 Signal histogram of file SCHEMATC6.
4.3 Signal Auto-Covariance Functions

The largest statistical programs are the programs AUTO and a closely related variation COVAR which calculate the autocorrelation and the closely related auto-covariance functions respectively. There the autocorrelation ($R_j$) and auto-covariance ($C_j$) are defined to be

$$R_j = \frac{1}{N} \sum_{i=1}^{N} x_i x_{i-j}$$
(4-1)

$$C_j = \frac{1}{N} \left( \sum_{i=1}^{N} x_i x_{i-j} \right) - (\bar{x})^2$$
(4-2)

$$C_j = R_j - (\bar{x})^2$$
(4-3)

Both $R$ and $C$ are calculated for $j$ having a range from zero to 255 or one complete scan line. In each case $N$ is such that all data was used; thus $N$ is approximately 65,000. Both programs can output the normalized functions $R_j/R_0$ or $C_j/C_0$ in tabular form on the teletype or as a graph on the type 30 display. The photographs of the auto-covariance functions obtained for the three picture files are shown in figures 18, 19 and 20. The correlation functions are useful for estimating the power spectrum of the random process of which these pictures are sample functions.

The auto covariance function for the face decays slowly, indicating the presence of low frequencies and hence a low detail picture. The increase in the center of the graph is caused by the correlation between the large signal representing the light part of the
Figure - 18 Auto-covariance of file HEATHER09

Figure - 19 Auto-covariance of file SCHEMATC6
Figure 20 Auto-covariance of file TEXTSPC06

Figure 21 Expanded display of the auto-covariance of file TEXTSPC06
face and a large signal from the bright background.

The auto-covariance of the file SCHEMATC6 decays most quickly, indicating that it was the most detailed picture. All three functions display a periodicity with a period of one scan line length which is expected for video signals. Figure -21 shows the left half of the TEXTSPC06 auto-covariance function expanded. The figure shows that the function has a periodic ripple superimposed on it. This ripple was assumed to be caused by the periodic nature of the text picture.

A commonly assumed auto covariance function for video is given by

$$\rho(\tau) = e^{-K|\tau|}$$  \hspace{1cm} (4-4)

To test to see if these files supported this assumption, the three auto-covariance functions were plotted on semi-logarithmic paper in figure -22. It can be seen that the covariance function of the file SCHEMATC6 agrees with the assumed function but the other functions are not in close agreement although the agreement is not unreasonable. The values of K for the two lines in figure -21 are 0.165 and 0.056. Wintz obtained a value of K of 0.090 for a picture of a face and K in the range of 0.1 to 0.2 for more detailed material.

4.4 Run Length Statistics

Run length encoding is a method which has been suggested for encoding two level pictures such as text
Figure - 22 Normalized auto-covariance functions for the three picture files.
or line drawings $^{11,12}$. In order to determine the compression possible by run length encoding, a program was written to compile run length histograms for the files containing text and the schematic. The six bit files contain signal values between zero and sixty-three. The program treats all signals between zero and thirty-one as zero and all signals from thirty-two to sixty-three as ones. The program generates several statistics (table VIII) such as transition probabilities which might be useful in modelling the source and statistics which indicate that it should be possible to obtain a compression of about three by using run-length encoding.

**TABLE VIII.**

**Runlength Statistics Produced by RLENGH**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>File TEXTSPC06</th>
<th>File SCHEMATC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average length of the black runs</td>
<td>5.43</td>
<td>4.44</td>
</tr>
<tr>
<td>Average length of the white runs</td>
<td>18.4</td>
<td>24.7</td>
</tr>
<tr>
<td>Entropy of the black histogram</td>
<td>2.73 (bits)</td>
<td>3.00 (bits)</td>
</tr>
<tr>
<td>Entropy of the white histogram</td>
<td>4.19 (bits)</td>
<td>5.44 (bits)</td>
</tr>
<tr>
<td>Entropy of the combined histogram</td>
<td>3.90 (bits)</td>
<td>4.49 (bits)</td>
</tr>
<tr>
<td>P(0)</td>
<td>0.225</td>
<td>0.152</td>
</tr>
<tr>
<td>P(1)</td>
<td>0.775</td>
<td>0.848</td>
</tr>
</tbody>
</table>
\[
P(0 \mid 0) = 0.816 \quad \text{0.774}
\]
\[
P(1 \mid 0) = 0.184 \quad \text{0.226}
\]
\[
P(1 \mid 1) = 0.945 \quad \text{0.955}
\]
\[
P(0 \mid 1) = 0.055 \quad \text{0.045}
\]

Once the run length histograms are compiled, the program can output them on the teletype or display them as a graph on the type 30 display. Pictures of graphs obtained in this way are shown in figures 23 and 24. As expected, the average white run lengths are much longer than the average black run lengths. The entropies shown in table VIII are calculated in the following manner. Consider a particular histogram which contains data on \(N\) runs which can have lengths of from 1 to \(m\). Let \(n(j)\) be the number of runs of length \(j\). In this program \(m\) is 256 which appears to be sufficiently large to accommodate all the runs. \(H\) is given by

\[
H = \sum_{j=1}^{m} \frac{n(j)}{N} \cdot \log_2 \left( \frac{n(j)}{N} \right) \quad \text{(bits)} \quad (4-5)
\]

The entropy is calculated to provide an indication of the number of bits which would be required to encode the run lengths if some efficient coding were used.

The theoretical compression ratio depends strongly on the material being transmitted. Ratios varying from five to seventeen have been reported. Some hardware has been reported which used runlength encoding on the white runs only to obtain a data com-
Figure - 23 Run length histograms for file TEXTSPC06.
Figure - 24 Run length histograms for file SCHEMATC6
pression of four. The actual compression obtained in practice may be much lower than the theoretical since it is usually not economical to implement hardware which would approach the theoretical performance.

4.5 A Line Display Program

DLINE is a short program which displays a line of data from a picture file. The operator can input line numbers from the teletype and the program will read the line of data from tape and display it using the GRAPH subroutine. DLINE was used primarily for checking on the operation of other data processing programs.

4.6 Summary

The programs just described provided picture statistics which were used in designing the simulation programs and in providing insight into the nature of the picture material. The programs in Chapter V, listed in table IX, are different in that they used the input picture data to produce new pictures.
CHAPTER V.

DPCM and PCM Simulation Programs

Subjective Test Procedures and Results

5.1 Introduction

This chapter will present a study made of picture degradation caused by channel errors in PCM and DPCM systems. The chapter first describes programs, Table IX, used to perform the following three functions: (1) simulation of DPCM and PCM systems with noisy channels, (2) smoothing the noisy pictures to reduce the visibility of the channel noise and (3) preparation of a set of standard pictures for use in subjective rating tests. Second, the subjective test procedures used to estimate the subjective quality of the noisy pictures are described. Last, the result of the subjective tests are presented.

TABLE IX.

Programs to Produce New Picture Files

<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSSIM</td>
<td>To simulate DPCM and PCM systems.</td>
</tr>
<tr>
<td>DGRAD</td>
<td>To prepare pictures with standard impairments.</td>
</tr>
<tr>
<td>FILTER</td>
<td>A non-linear filtering program for smoothing PCM Pictures.</td>
</tr>
<tr>
<td>NLDPMF</td>
<td>A non-linear filtering program for smoothing DPCM pictures.</td>
</tr>
</tbody>
</table>
5.2 Simulation Programs

5.2.1 PCM and DPCM Simulation Program SYSSIM

In order to simulate the pictures which would be produced by PCM or DPCM systems with noisy channels, a program called SYSSIM was written which could simulate either of the systems shown in figure -25. The program is restricted to simulating systems which require up to five bits of non-redundant channel coding. For this work it was not necessary to consider higher quality systems because of the noise limit imposed by the camera.

The program allows the assignment of any quantizer characteristic and any non-redundant channel code to represent the quantizer levels. The only restriction on the quantizer is that the DPCM quantizer characteristic $y = q(z)$ be an odd function of $z$.

The noisy channel model used is a binary symmetric channel (BSC) which has a transition diagram as shown in figure -26.

![Diagram](attachment://binary_symmetric_channel_diagram.png)

Fig. 26 Transition diagram for a binary symmetric channel
Figure 25 Block diagrams of systems simulated by the program SYSSIM
This type of channel can be characterized by one parameter, the bit error probability $P$. The relationship between $P$ and parameters of some other channel models has been discussed by Yan\textsuperscript{31}.

For the PCM system, the operator has the choice of quantizer characteristic including the number of levels, the magnitude of a random number or dither signal ($d_i$) and the bit error probability $P$. It was decided that since the picture signal histogram or probability density function was fairly uniform, uniform quantization and natural coding would be used\textsuperscript{32}. Using that configuration a set of PCM pictures was produced from the file HEATHER09 with the number of channel bits equal to 2, 3 and 4 and the bit error probabilities of $10^{-4}$, $10^{-3}$, $10^{-2}$ and $10^{-1}$. The bit range of two to four was chosen because one bit PCM would not be meaningful and five bit PCM was too high quality for simulation with the available hardware. The bit error rate $P$ was chosen to yield a wide range of picture quality from essentially error free with $P$ equal to $10^{-4}$ to very noisy pictures with $P$ equal to $10^{-1}$. Figure -27 shows the measured inband SNR obtained for the various PCM systems.

For the DPCM system\textsuperscript{30} the operator has control over the quantizer characteristic, the feedback parameter $\alpha$ and the bit error probability $P$. 
Figure - 27 Mean-square signal to mean-square error ratio of a PCM system.

Figure - 29 Mean-square signal to mean-square error ratio of a DPCM system.
Since the probability density function or signal histogram of the difference signal $z^*$ is sharply peaked around zero it was decided to use a non-uniform quantizer which quantized small values of $z$ more accurately than it quantized large values. The quantizer chosen was a logarithmic quantizer which followed the compression law.

$$V_{out} = \frac{V}{\log(1 + \mu)} \cdot \log(1 + \frac{\mu|V_{in}|}{V}) \cdot \text{sgn}(V_{in}) \quad (5-1)$$

where $\mu$ was 100 and $V$ was 160. The quantizer range was approximately $6\sigma_z$ where $\sigma_z^2$ is the variance of the signal $z$.

The choice of $\alpha$ was a tradeoff between decreasing quantizing noise and increasing sensitivity to channel noise. In a well designed DPCM system employing optimum prediction and where the quantizer range is scaled according to $\sigma_z^2$ then 38.

$$\overline{\epsilon^2} = (1 - \alpha^2) \overline{x^2} \cdot n_q + \frac{1}{(1 - \alpha^2)} \cdot \{n_c(1 - \alpha^2) \overline{x^2}\} \quad (5-2)$$

Where

- $n_q$ = quantization noise when the quantizer input signal $z$ has unit variance.
- $n_c$ = channel noise when the quantizer input signal $z$ has unit variance.
- $\overline{x^2}$ = input signal power.

It can be seen if optimum prediction is used then theor-
ically the quantizing noise is decreased and the channel noise is the same as for a PCM system. There are two assumptions underlying this formula which are not satisfied in this case. First, it is assumed that quantizing noise is negligible in the feedback loop. This is not true for the low number of bits used in this system. Secondly it is assumed that the quantizer range and hence the channel noise power are scaled down exactly by the factor \((1 - a^2)\).

In practice with this system it was found that if \(a\) approached the value of \(\frac{R(1)}{R(0)}\) then the sensitivity of the error power to channel noise increased. The effect of \(a\) on the system SNR is shown in figure -28. It can be seen that as \(a\) approaches one the system is very strongly affected by noise, while if \(a\) is about 0.9 the system is not as sensitive to the noise although the noise free performance is still not drastically degraded. For this reason it was decided to use an \(a\) of 0.9 for all DPCM simulations.

Once these design considerations had been settled, a set of pictures was generated with the number of bits equal to 1, 2, 3 and 4, and with \(P\) assuming values of \(1/50, 10^{-2}, 10^{-3}\) and \(10^{-4}\). The range of the parameters for the DPCM system were chosen using the same considerations as for the PCM system. Figure -29 shows a graph of the SNR attained by the various DPCM
Figure - 28 Variation of signal to noise ratio with $\alpha$ for a 3 bit DPCM system.
systems. It was found that when \( P = 10^{-1} \) the DPCM system produced pictures of such poor quality that they could be ignored.

After these sets of PCM and DPCM noisy pictures had been generated an attempt was made to reduce the visibility of noise produced by channel errors. Because of the vastly different nature of the noise in the PCM and DPCM pictures, it was necessary to develop two filtering routines, one for PCM pictures and one for DPCM pictures.

5.2.2 Picture Smoothing Programs

The filters used, were in both cases non-linear. Linear filtering operations were tested but it was found the blurring introduced by the low pass nature of these filters was more annoying than the noise they were supposed to remove.

In a PCM system, each point is transmitted independently. Thus if a transmission error occurs, only a single point will be changed. As a result, PCM pictures with errors tend to have noise which can be described as "salt and pepper" noise. Some examples of noisy PCM pictures are shown in appendix -B. A method for reducing salt and pepper noise was mentioned by Rosenfeld and operates as follows.
Consider the point \( X \) in figure 30. To determine if \( X \) was received correctly form the difference between \( X \) and the average of its eight nearest neighbours. If the difference is less than a threshold value assume the point is correct and leave it. If the difference exceeds the threshold, replace it by the average value. In other words if the point \( X \) has a value of \( IP(i,j) \) form

\[
D = \left| \frac{1}{8} \sum_{k=1}^{3} \sum_{l=1}^{3} IP(i+k-2,j+l-2) - 9 \cdot IP(i,j) \right| \quad (5-3)
\]

If \( D < \) Threshold \( \rightarrow OP(i,j) = IP(i,j) \)
If \( D > \) Threshold \( \rightarrow OP(i,j) = AVG \)

\[
AVG = \frac{1}{8} \sum_{k=1}^{3} \sum_{l=1}^{3} IP(i+k-2,j+l-2)
\]

A subroutine called FILTER was written in Fortran to implement this algorithm. The program was run with the data handling programs PROCR and FPACK and a set of
pictures was produced by filtering the PCM pictures which had been produced previously.

DPCM picture noise due to channel errors has quite a different nature from PCM noise and required a completely different filtering strategy. This nature can be viewed in two ways.

The first way is to consider the DPCM source decoder to be a low pass filter. Thus any wide band noise added to \( \hat{Y}_i \) by the channel noise will be filtered and a low frequency noise will appear at the system output \( \hat{X}_i \). It can be shown that low frequency noise in the video signal gives rise to two dimensional noise which is very anisotropic. The two dimensional noise will have a wide band of frequencies in the vertical direction and only low frequencies present in the horizontal direction. As a result the noise will vary much more rapidly in the vertical than in the horizontal direction. The noise appears as a series of horizontal streaks whose lengths depend on the parameter \( \alpha \) which determines the filter bandwidth.

A second view of the noise is to consider the system output (figure -25)

\[
\hat{X}_i = \hat{Y}_i + \alpha \hat{Y}_{i-1} + \alpha^2 \hat{Y}_{i-2} + \cdots \quad \text{(5-4)}
\]

\[
\hat{X}_i = \sum_{j=0}^{n} \alpha^j \hat{Y}_{i-j} \quad \text{(5-5)}
\]
It can be seen that each received point depends not only on the present value of \( \hat{\mathbf{x}}_i \) but also on a weighted sum of past values of \( \hat{\mathbf{x}}_i \). If one of the \( \hat{\mathbf{x}}_i \) is incorrect, it will cause an error in \( \hat{\mathbf{x}}_i \) and also in subsequent points \( \hat{\mathbf{x}}_{i+k} \), \( k=1,2 \ldots \) until its effect has decayed. The decay rate depends directly on \( \alpha \). If \( \alpha \) is close to one, the error will decay slowly and a long error streak will appear. If \( \alpha \) is less than one, the streak length decreases proportionately.

Because of the anisotropic nature of the DPCM noise, it is possible to distinguish the noise from the picture material only by its high frequencies in the \( y \) direction. A linear filter which filtered the picture only in the \( y \) direction was tried. This filter introduced more degradation due to blurring than it removed. The filter which was finally used operated as follows.

\[
\begin{align*}
J & \quad \text{PM} \\
(I-1) & \quad \text{PM} \\
I & \quad \text{P} \\
(I+1) & \quad \text{PM}
\end{align*}
\]

\[
d_1 = P - \text{PM} \quad (5-6)
\]

\[
d_2 = \text{PP} - P
\]

Fig. 31 Data points used by the DPCM filter to smooth point \( X \)

\( X \) is again the point being considered. \( P \) is the value of the picture function at point \( X \) in the input picture.
OP will be the value of the picture function in the output picture. Form the two differences $d_1$ and $d_2$ and then perform the logic shown in figure -32.

$$\text{sgn}(d_1) = \text{sgn}(d_2)$$

If $|d_2| > \frac{1}{2}|d_1|$, then

$$\text{OP} = \frac{1}{2}( PM + PP )$$

If $|d_2| \leq \frac{1}{2}|d_1|$, then

$$\text{OP} = IP(i,j)$$

Fig. 32 Filter algorithm for the DPCM filtering routine

It can be seen that such a filter will smooth any parts of the picture which change rapidly in the vertical direction. This filter was not as successful as the PCM filter. The algorithm breaks down when $P$ becomes large enough that adjacent error lines occur. Adjacent error lines were found to be a problem when $P$ was more than $10^{-2}$.

A program NLDPMF using this algorithm was written using a combination of Fortran and assembly language and was used with the data programs PROCR and FPACK. With these programs a set of filtered DPCM pictures was
produced by filtering all the DPCM pictures previously produced.

The objective of generating all these pictures was to study the subjective effect of the channel noise. It was decided to use a picture rating system where subjects would compare the noisy pictures with a set of standard pictures. The third distinct picture processing program was DGRADE which produced these standards.

5.2.3 DGRADE a Standard Producing Program

For a rating, subjects were asked to compare the test pictures with a set of standard pictures. When a subject had decided which standard had the same quality as the test picture, the SNR of the standard "X" was assigned to the test picture as an indication of its quality. The test picture is then said to have an equivalent SNR of X db.

The set of standard picture should ideally satisfy two conditions. First, the quality range in the standard pictures must be large enough to match the range of test picture quality. Second, for consistent results, the nature of impairment in the standard pictures should be the same as that in the test pictures. The second condition was impossible to satisfy since the PCM and DPCM pictures had four distinct types of impairment. In the PCM pictures, degradation was due to the low number of bits used and channel noise. DPCM
pictures were degraded by the same factors but they produce errors whose appearance was different from the PCM errors.

The set of standards which were used were generated using a method suggested by Schroeder\textsuperscript{34}. The output signal is given by

\[ \hat{x}_i = (1 + \alpha^2)^{-1/2} x_i (1 + \alpha \cdot \varepsilon_i) \]  \hspace{1cm} (5-7)

Where \( \varepsilon_i \) is a noise signal defined by the following equations.

\[ \varepsilon_i = \pm 1 \quad \varepsilon_i = 0 \]  \hspace{1cm} (5-8)

\[ \varepsilon_i \cdot \varepsilon_j = \delta_{ij} \quad \varepsilon_i \cdot x_i = 0 \]

This produces pictures whose SNR was given by

\[ \text{SNR} = \frac{1}{\alpha^2} \]  \hspace{1cm} (5-9)

A program called DGRADE was written for use with PROCES which produced an output as defined by equation (5-7). The noise process was simulated using the random number program FRAND. Using this program and appropriate \( \alpha \) a set of pictures was produced with SNR's of 6, 10, 14, 18, 22 and 26 db, for use in the subjective tests as standards. This range of SNR was chosen after some inspection of the programs output. In addition to generating these standards, pictures with SNR of 8, 12 and 16 db were generated. These pictures were included
as test pictures in the subjective tests to check the results.

5.2.4 Simulation Program Summary

This section has described picture processing routines which produced three sets of pictures. First a set of pictures which simulated pictures which had been transmitted by PCM and DPCM systems with noisy channels were described. Second a set of pictures was produced by filtering the noisy PCM and DPCM pictures. Third, a set of standard pictures was generated for use in a subjective test. The next section will discuss the subjective tests which were performed with the pictures.

5.3 Subjective Test Procedures

As mentioned in the previous section it was decided to use a rating type of subjective test for determining the subjective quality of the test pictures. The programs in the previous section had produced 59 2½ inch square test pictures. The set of pictures contained twelve PCM pictures, sixteen DPCM pictures, twelve filtered PCM pictures, sixteen filtered DPCM pictures and three test standard pictures.

Before a test, the subject was given a copy of the instruction which are reproduced below.

You will be asked to rate the subjective quality of a set of pictures by comparing the test pictures with a set of standard pictures. The standard pictures are identified by numbers which appear on the right sides of the pictures.
You will be given one test picture at a time. For each test picture, write on the test form the number of the standard picture which you feel has the same quality as the test picture. If the test picture quality is between two standards, assign the test picture a rating which is midway between the two standard values. If the quality of the test picture is worse than the quality of the poorest standard, give it a rating of 0 (zero). If the quality of the test picture is better than that of the best standard give it a rating of 28.

View the pictures from a distance of 15 inches. Please do not touch the front of the photographs.

For the test, the pictures were arranged in random order. Two different random orderings were used in order to reduce the effect of any interaction between the test pictures. Each of the ten subjects was tested individually. The subjects were all graduate students in the Department of Electrical Engineering at the University of British Columbia and had no previous experience with subjective evaluation of pictures.

5.4 Subjective Test Data

Once the test had been completed, the data from the subjects' answer sheets was unscrambled and combined to obtain ten estimates of picture quality for each picture.

For a particular picture, the ten estimates by the different subjects are in general not identical. It was desirable to assume that for a particular picture the grades are samples drawn from a population with a normal distribution with parameters \( \mu \) and \( \sigma \). If this assumption is true, it would be possible to use the extensive theory which is available
concerning sampling from normal populations. To check this assumption, a Kolmogorov-Smirnov\textsuperscript{39} test was performed on all sets of data. The hypothesis was accepted at the 5% significance level for all but one set of data and was accepted by all sets of data at the 1% significance level. Each picture was assigned a quality which was the average of the sample data for that picture. The 95% confidence interval was calculated (equation 5-10) using the student t distribution.

$$\bar{x} - t_\alpha \frac{s}{\sqrt{n-1}} < \mu < \bar{x} + t_\alpha \frac{s}{\sqrt{n-1}}$$

(5-10)

The data appeared to be quite consistent within each set and between sets. The consistency within each set of data is indicated by the reasonably small confidence intervals. The consistency between sets of data can be observed in figures 33, 34, 35 and 36, where the subjective signal to noise ratios are plotted for the four systems. Two points initially appeared to be in error. It appeared that the data for these points had been reversed. However, a careful check revealed no bookkeeping errors. As a result a short test was made with the two anomalous points and a few adjacent points. The results from this spot check were in agreement with the rest of the data.

Once the data had been tested to verify the normality assumption and confidence intervals calculated, several graphs were plotted in order to place any general conclusions in evidence.
Figure - 33 Subjective SNR versus P for a PCM system.

Figure - 34 Subjective SNR versus P for a DPCM system.
Figure - 36 Subjective SNR versus P for a filtered DPCM system.

Figure - 35 Subjective SNR versus P for a filtered PCM system.
Figure - 37 Subjective iso-quality lines and lines of constant mean-square error for a PCM system.

Figure - 38 Subjective iso-quality lines and lines of constant mean-square error for a DPCM system.
Two types of signal to noise ratio (SNR) are referred to on the graphs. The first, the objective SNR is defined by equation (5-11).

\[
\text{SNR}_{\text{obj}} = \frac{\overline{x_i^2}}{(x_i - \overline{x}_i)^2} \quad (5-11)
\]

The second, the subjective SNR is the value which was obtained from the subjective tests.

Two sets of graphs were plotted. The first comprises graphs showing subjective SNR plotted versus bit error probability P with the number of bits per sample N as a parameter. In the second set of graphs, the parameter was the quality q and lines of constant quality or iso-quality curves were plotted on the (P-N) plane. From these curves it is possible to draw some conclusions.

5.5 Data Presentation and Results

The iso-quality graphs in figures 37 and 38 illustrate the performance of the PCM and DPCM systems without filtering. Curves of constant mean-square error (MSE) were plotted on the same axis in order to examine the relationship between the subjective quality and the MSE.

\[
\text{MSE} = \overline{e^2} = (x_i - \overline{x}_i)^2
\]

(5-12)

It can be seen that for the PCM system the shape of the
MSE curves and the iso-quality curves are very similar. Thus, for the range of parameters used, the MSE would appear to be a good indicator of the subjective picture quality.

In figure -38 similar curves are shown for a DPCM system. It can be seen that the subjective quality curves are affected more by channel error than the MSE curves. This is indicated by the fact that the subjective iso-quality lines curve to the right more quickly than the MSE lines do. Thus in the study of channel errors in DPCM systems, the MSE might be an over optimistic indicator of subjective quality.

Figure -39 was plotted so that the DPCM and PCM systems could be compared. From the figure, it can be seen that for good channels with $P$ less than $10^{-3}$ the DPCM systems outperforms the PCM system. However, for channels error probabilities between $10^{-2}$ and $10^{-3}$, the curves cross over and the PCM system produces pictures which are subjectively better. The cross over shows that the MSE is not a good indicator for comparing two different source encoding schemes since if figures 27 and 29 are compared it can be seen that the MSE of the DPCM system is equal to or better than the MSE of the PCM system for all values of $P$ which are considered. This is in agreement with work done previously on subjective noise weighting functions $^{17,19}$, where it was found that the noise weighting function was low-pass in nature. In other
Figure 39: Subjective iso-quality lines for the DPCM and PCM systems.
words, low frequency was more annoying than high frequency noise. As mentioned previously, the DPCM source decoder behaves like a low pass filter and changes the wideband channel noise into a low frequency signal. No such filtering action takes place in a PCM system. Thus it would be expected that the low frequency DPCM noise could be more annoying than the wide band PCM noise even though the PCM noise might be of greater power.

In order to determine the effect of the noise filtering, iso-quality curves for an ordinary PCM system and one with two-dimensional smoothing were plotted in figure -40. It can be seen from this curve that the filter does allow the system to operate with higher channel error probabilities. This improvement can also be seen if figures 33 and 35 are compared.

Similar curves were plotted so that the DPCM filter algorithm could be assessed (figure -41). From these curves, it can be seen that for good channels, the filter lowers the subjective quality while for poor channels the quality is the same as the unfiltered system. Thus it appears that filtering DPCM pictures with the proposed filter produced no improvement in quality. The failure of this filter was attributed to the fact that it produced some blurring in the picture. Thus, although it did reduce the visibility of the noise, the improvement was more than offset by the degradation due to blurring.
Figure - 40 Subjective iso-quality lines for the filtered and unfiltered PCM systems.

Figure - 41 Subjective iso-quality lines for the filtered and unfiltered DPCM systems.
5.6 **Subjective Test Summary**

In this chapter subjective rating tests were described which gave fairly consistent results. The tests indicate that the subjective quality of pictures transmitted by PCM systems over poor channels can be better than the subjective quality produced by DPCM systems, even though the MSE of the DPCM system was smaller. The test also indicate that the PCM noise filter tested improves the subjective quality of the noisy pictures while the proposed DPCM filter produces no improvement and even tends to lower the quality of pictures transmitted over good channels.
APPENDIX A. DISPLAY COMMANDS AND CONNECTIONS

### COMMANDS

<table>
<thead>
<tr>
<th>Neumonic</th>
<th>Code</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>S31</td>
<td>703107</td>
<td>Load SOI 31 with the contents of the accumulator</td>
</tr>
<tr>
<td>S32</td>
<td>703207</td>
<td>Load SOI 32 with the contents of the accumulator</td>
</tr>
<tr>
<td>LDX</td>
<td>706302</td>
<td>Load the x D/A converter buffer from SOI 31</td>
</tr>
<tr>
<td>LDY</td>
<td>705302</td>
<td>Load the y D/A converter buffer from SOI 31</td>
</tr>
<tr>
<td>EXPS</td>
<td>705701</td>
<td>Expose a point</td>
</tr>
<tr>
<td>SFIN</td>
<td>705601</td>
<td>Skip if the display busy flag is clear</td>
</tr>
</tbody>
</table>

### CONNECTIONS

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOI 31</td>
<td>E(13-22)</td>
<td>Position D/A inputs *</td>
</tr>
<tr>
<td>IOT 6302</td>
<td>L-3</td>
<td>Load x D/A pulse *</td>
</tr>
<tr>
<td>IOT 5302</td>
<td>G-3</td>
<td>Load y D/A pulse *</td>
</tr>
<tr>
<td>SOI 32</td>
<td>H(11-22)</td>
<td>G(11-22)</td>
</tr>
<tr>
<td>SBUSY</td>
<td>SBUSY*</td>
<td>F-10</td>
</tr>
<tr>
<td>CBUSY</td>
<td>CBUSY*</td>
<td>F-9</td>
</tr>
<tr>
<td>IOT 5601</td>
<td>G-8</td>
<td>F-8</td>
</tr>
<tr>
<td>IOT 5701</td>
<td>J-2</td>
<td>Expose pulse *</td>
</tr>
<tr>
<td>Reference</td>
<td>Output</td>
<td>VREF *</td>
</tr>
<tr>
<td>Voltage</td>
<td>D/A #2</td>
<td></td>
</tr>
</tbody>
</table>

*indicates a point on the external display electronics
APPENDIX B. PCM AND DPCM TEST PICTURES

This appendix contains copies of the pictures used in the subjective tests. Each picture is identified by a nine character code which has the following form

XXXXnBTPm

The first four characters XXXX are a code identifying the type of system simulated. The following codes were used:

- PCMN: PCM system
- DPCM: DPCM system
- PNFL: PCM system with filtering
- DPNL: DPCM system with filtering
- DGRAD: Subjective test standard picture

The character n specifies the number of bits per picture element. The character m specifies the bit error probability.

<table>
<thead>
<tr>
<th>m</th>
<th>Error Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$10^{-1}$</td>
</tr>
<tr>
<td>2</td>
<td>$10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>8</td>
<td>$2 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

For example DPNL3BTP4 represents a three bit DPCM system with filtering and a bit error probability of $10^{-4}$. Pictures produced by the program DGRADE are identified by a different code of the form DGRADssDB which represents a standard picture with a SNR of ss decibels.
REFERENCES


