AUTOMATIC SCANNING OF BRAIN SECTIONS
PREPARED BY AUTORADIOGRAPHIC METHODS

by

Richard Eugene Hamilton

B.S.E.E., Purdue University (1970)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1979

Signature of Author______________________________________________

Interdisciplinary Science Program

Approved by_____________________________________________________

Technical Supervisor (CSDL)

Certified by_____________________________________________________

Thesis Supervisor

Accepted by_____________________________________________________

Chairman, Interdisciplinary Science Committee

Archives

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Libraries

1
AUTOMATIC SCANNING OF BRAIN SECTIONS
PREPARED BY AUTORADIOGRAPHIC METHODS

by

Richard Eugene Hamilton

Submitted to the Interdisciplinary Science Department in January 1979, in partial fulfillment of the requirements for the Degree of Master of Science

ABSTRACT

An automatic autoradiography grain-counting system has been developed. In Syrian hamster and mouse brain hemi-sections, a 100-mm² area can be scanned in less than an hour at 40× magnification using a single plain of focus with isolated grain density independence. Two forms of autoradiographically prepared brain tissue were studied: cell development material and axon transport material. The system was also used to quantify degenerating axons and terminals in silver-stained material. Emphasis was placed on dynamic range and data acquisition rather than on data analysis or accuracy evaluation. Data are transferred to mass storage for further processing. The system consists of a microscope, stepper motor stage, linear photoreceptor CCD array, computer interface, computer, mass storage, and display peripherals. Image processing hardware with grain feature extraction hardware and software made rapid grain counting viable. Microcomputer control of the 10-μm stepper stage made 0.15-μm picture-element scanning viable.

KEY WORDS

Autoradiography, Brain, Computer, Pattern Recognition, Microprocessor Image Processing, CCD Linear Array, Degeneration

Thesis Supervisors:

Gerald E. Schneider, Professor of Psychology
Brock S. Dew, Section Chief, Charles Stark Draper Laboratory
ACKNOWLEDGMENT

I express my appreciation to The Charles Stark Draper Laboratory, Inc. (CSDL) for supporting my education and research. This report was prepared at CSDL and its publication was funded by National Institute of Health (NIH) Grant No. GM 22965-03.

My warmest appreciation goes to Brock S. Dew and Gerald E. Schneider, my thesis advisors, for their guidance and patience throughout the research. I would like to thank my mentors Tom McMahon, John Dowling, Richard Sidman, David Hubel, Gerald Schneider, Ann Graybeil, William Schreiber, David Marr, Brock Dew, and Murray Eden for the education I have gained in accomplishing this project. Thomas King has been very helpful in the design, troubleshooting, and building of the computer interface hardware. William Selles has aided the system software development. This project would not have been possible without the support of these people.

Publication of this report does not constitute approval by CSDL or NIH of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOSSARY</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>1.1 Research Objectives</td>
<td>9</td>
</tr>
<tr>
<td>1.2 Autoradiographic Features</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Background</td>
<td>15</td>
</tr>
<tr>
<td>1.4 Rationale</td>
<td>18</td>
</tr>
<tr>
<td>1.5 Specific Aims</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>PROCEDURE</td>
</tr>
<tr>
<td>2.1 System Optomechanical Description</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Microscope Hardware</td>
<td>28</td>
</tr>
<tr>
<td>2.3 Computer Interface Hardware</td>
<td>33</td>
</tr>
<tr>
<td>2.4 Computer Hardware</td>
<td>52</td>
</tr>
<tr>
<td>2.5 System Software</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>RESULTS</td>
</tr>
<tr>
<td>3.1 System Accuracy Results</td>
<td>72</td>
</tr>
<tr>
<td>3.2 Axon Transport Material Results</td>
<td>80</td>
</tr>
<tr>
<td>3.3 Cell Development Material Results</td>
<td>87</td>
</tr>
<tr>
<td>3.4 Degeneration Material Results</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>CONCLUSION</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>101</td>
</tr>
<tr>
<td>APPENDIX—SOFTWARE SOURCE LISTING</td>
<td>103</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Silver grains</td>
</tr>
<tr>
<td>2</td>
<td>Silver grain optical properties</td>
</tr>
<tr>
<td>3</td>
<td>Automatic microscope development system</td>
</tr>
<tr>
<td>4</td>
<td>System block diagram</td>
</tr>
<tr>
<td>5</td>
<td>System command flow diagram</td>
</tr>
<tr>
<td>6</td>
<td>Information flow diagram</td>
</tr>
<tr>
<td>7</td>
<td>Automatic microscope</td>
</tr>
<tr>
<td>8</td>
<td>Automatic focus mechanism</td>
</tr>
<tr>
<td>9</td>
<td>CCD photoreceptor linear array</td>
</tr>
<tr>
<td>10</td>
<td>Computer interface cards</td>
</tr>
<tr>
<td>11</td>
<td>Command control box</td>
</tr>
<tr>
<td>12</td>
<td>Image enhancement filter</td>
</tr>
<tr>
<td>13</td>
<td>Image preprocessor</td>
</tr>
<tr>
<td>14</td>
<td>Feature extractor operation modes</td>
</tr>
<tr>
<td>15</td>
<td>Actuator system analysis</td>
</tr>
<tr>
<td>16</td>
<td>Actuator system block diagram</td>
</tr>
<tr>
<td>17</td>
<td>Actuator command translator schematic</td>
</tr>
<tr>
<td>18</td>
<td>Actuator command translator card</td>
</tr>
<tr>
<td>19</td>
<td>Actuator driver card</td>
</tr>
<tr>
<td>20</td>
<td>System operation commands</td>
</tr>
<tr>
<td>21</td>
<td>Microscope command interpreter</td>
</tr>
<tr>
<td>22</td>
<td>Operation flow diagram</td>
</tr>
<tr>
<td>23</td>
<td>Initialize procedure</td>
</tr>
<tr>
<td>24</td>
<td>Focus procedure</td>
</tr>
<tr>
<td>25</td>
<td>Scanning procedure</td>
</tr>
<tr>
<td>26</td>
<td>Actuator command generator</td>
</tr>
<tr>
<td>27</td>
<td>Actuator command translator</td>
</tr>
<tr>
<td>28</td>
<td>Data compression software</td>
</tr>
<tr>
<td>29</td>
<td>Optical transfer characteristics</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>30</td>
<td>Preprocessor image enhancement</td>
</tr>
<tr>
<td>31</td>
<td>Dense grain preprocessor response</td>
</tr>
<tr>
<td>32</td>
<td>Test slide scan</td>
</tr>
<tr>
<td>33</td>
<td>Scan linearity—slow speed</td>
</tr>
<tr>
<td>34</td>
<td>Scan linearity—fast speed</td>
</tr>
<tr>
<td>35</td>
<td>Axon transport material—G1 scan</td>
</tr>
<tr>
<td>36</td>
<td>Axon transport material—G1 photo</td>
</tr>
<tr>
<td>37</td>
<td>Axon transport material—G1 detail scan</td>
</tr>
<tr>
<td>38</td>
<td>Axon transport material—G2</td>
</tr>
<tr>
<td>39</td>
<td>Superior colliculus count histogram</td>
</tr>
<tr>
<td>40</td>
<td>Cell development material photo</td>
</tr>
<tr>
<td>41</td>
<td>Cell development material scan</td>
</tr>
<tr>
<td>42</td>
<td>Cell development material detail scan</td>
</tr>
<tr>
<td>43</td>
<td>Cell development material cortex scan</td>
</tr>
<tr>
<td>44</td>
<td>Cell development count histogram</td>
</tr>
<tr>
<td>45</td>
<td>Degeneration material scan</td>
</tr>
<tr>
<td>46</td>
<td>Degeneration material detail scan</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Actuator generator</td>
<td>The program that generates motion command lists for the microscope stage.</td>
</tr>
<tr>
<td>Actuator translator</td>
<td>The microcomputer that decodes the motion command lists and drives the microscope stage.</td>
</tr>
<tr>
<td>Autoradiography</td>
<td>Histological process involving the development of a photographic emulsion exposed to radiochemicals.</td>
</tr>
<tr>
<td>Binary</td>
<td>The black or white representation of optical density.</td>
</tr>
<tr>
<td>Buffer</td>
<td>A hardware device used to absorb timing variances in a computer system.</td>
</tr>
<tr>
<td>CCD line array</td>
<td>The photoreceptor pel string electronic sensor.</td>
</tr>
<tr>
<td>END</td>
<td>The two-dimensional location of a silver grain on a microscope slide.</td>
</tr>
<tr>
<td>Feature extractor</td>
<td>The electronics required to recognize grain END points.</td>
</tr>
<tr>
<td>Gray level</td>
<td>The white through gray to black representation of optical density.</td>
</tr>
<tr>
<td>Hardware</td>
<td>The physical electronic components of a computer system.</td>
</tr>
<tr>
<td>Hybrid scanning</td>
<td>The technique of imaging using mechanical scan and electrical scan together.</td>
</tr>
<tr>
<td>Joystick</td>
<td>The physical device used to command three-dimensional motion.</td>
</tr>
<tr>
<td>Latent image</td>
<td>The undeveloped image in a photographic emulsion.</td>
</tr>
<tr>
<td>Mel</td>
<td>Motion element, size of motion quantum step in the microscope stage.</td>
</tr>
<tr>
<td>Multiplexing</td>
<td>The technique of selecting any one of many inputs for only one output.</td>
</tr>
<tr>
<td>Pel</td>
<td>Picture element, size of image quantizing element.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Point spread</td>
<td>The normal distribution of optical density after passing through optics.</td>
</tr>
<tr>
<td>Preprocessor</td>
<td>The electronics required to normalize the video for the thresholding operation.</td>
</tr>
<tr>
<td>Radiochemical</td>
<td>A radioactively tagged chemical.</td>
</tr>
<tr>
<td>Silver grains</td>
<td>The silver particles resulting from photographic development.</td>
</tr>
<tr>
<td>Software</td>
<td>The program instructions to run the hardware.</td>
</tr>
<tr>
<td>Threshold</td>
<td>The operation required to transform gray level to binary.</td>
</tr>
<tr>
<td>Throughput</td>
<td>The total time required to process material in a computer system.</td>
</tr>
<tr>
<td>Video</td>
<td>The voltage representation of optical density sensed by the line array.</td>
</tr>
<tr>
<td>$E$-Emission</td>
<td>The atomic particle that forms the latent image.</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Research Objectives

The objective of this thesis is to show the feasibility of an inexpensive rapid automatic data-acquisition system that will count silver grains in autoradiographically prepared animal tissue and store the data for later analysis. Syrian hamster and mouse brain sections were used for test material. Another tested application of the system was the counting of silver particles in axonal degeneration material stained by silver methods.

The grain-counting system is a modification of an existing metaphase-cell-location system developed at The Charles Stark Draper Laboratory, Inc. (CSDL) (System Development Division) with funds from National Institute of Health (NIH). A strong attempt was made to limit the redesign of the metaphase-cell-location system so that the general context of using an automatic microscope system to solve many biological problems could be maintained.

The grain-counting system consists of a microscope, imaging electronics, a motorized microscope stage, a computer interface, and
computer software. All components can be purchased commercially, except
the computer interface.

1.2 Autoradiographic Features

The use of radiochemical tracers in the investigation of cells
and intercellular relations is an important area of biological research.
The radiochemicals are injected in the living animal and remain in the
tissue after the animal is sacrificed, or they are absorbed in vitro
during tissue preparation. The chemicals are then detected in the tis-
sue by developing a latent image caused by $\beta$-emission into a photo-
graphic emulsion overlying a radioactive site. Spatial specificity of
the radiochemical tracer is thus achieved by the silver grain location.
The exact location of the radioactive site is limited by the exposed
crystal location and by the random nature of grain development. In-
creased specificity can be obtained by using smaller crystals, but this
is limited by the development process and by perceptual detection of
contrast, which depends on the diffraction limit of light in the micro-
scope. The exposed crystals (see Figure 1) around a radioactive site
may form a normal distribution of grain density (see Figure 2); half-
distance units have been used as a spatial measure. A half-distance
unit is the radial distance from the center of the grain distribution
where half of the total grains in the distribution lie.\(^{(1)}\)*

* Superscript numerals refer to similarly numbered items in the List of
References.
Figure 2. Silver grain optical properties.
through the microscope optics, any individual grain forms a normal distribution of optical density around its site.

The grain optical distribution is known as the optical point spread (Figure 2). The point spread is used to set the recognition threshold and define the size of the image on the photoreceptor picture element, called a pel unit. Two pel units should spatially encompass half of the grain point-spread photon density. The photoreceptor array contains 1024 pel units, 13 μm per pel. Thus, the objective magnification determines the size of the grains recognized by magnifying a grain to exactly two pel units (26 μm).

The two methods of applying a radiosensitive material on a tissue slide are stripping film and dipping emulsion. The most common in use are Kodak NTB and Kodak NTB-2 dipping emulsions, and Kodak Aî-10 stripping film sheets. All films have an approximate undeveloped crystal size of 0.2 μm in diameter. The latent image formed by exposure times of 5 days to 2 years to the radiochemical is developed in Kodak D-19 for 3-5 minutes. (1) The resultant silver grains range from 0.25 to 1.0 μm in diameter, depending on the development time.

There is always a background of grains that has nothing to do with the radiochemically produced grains. The background is caused from natural radioactivity in the air and in tissue. The background is often ignored, but in quantitative studies it is relevant. The most common background measurement technique is to use radioactive and nonradioactive
slides in every experiment. Statistical grain-count distributions can be used to determine background bias.

The strongest application of autoradiography is in the neurosciences, and the types of application are: the site, time of origin, routes, and rates of cell migration; the tracking of axon pathways using labeled axonally transported proteins; and the physiological characteristics of histologically defined structures revealed by the amount of glucose intake during exposure to a stimulus.

An application of cell-development autoradiography (sometimes referred to as chromosome autoradiography) is the measurement of cell generation cycles. The most reliable method of measuring interphase, metaphase, telephase, and prophase is to pulse dose cells with tritiated thymidine ($H^3$-TdR), a DNA precursor, and to sample at hourly intervals for labeled mitotic nuclei. This information is helpful in studies of the "to-fro" growth during the formation of the fundamental cortical layers in early brain development. In addition to cell proliferation, autoradiography is used to study migration and differentiation in postmitotic cells. Cell birthdays—the last day on which nuclear DNA is replicated in that cell line—are obtained by injecting cells with $H^3$-TdR during development and allowing growth to maturity before sacrifice. Heavily labeled cells are those that stopped dividing on the day of injection, and lightly labeled cells are the progeny of further divisions after that time. $H^3$-TdR is available for synthesis only 2 hours after injection, but labeled cells remain labeled for life. Once a population
of cells is known to exist in a specific area, migration studies of this population can be done using a series of genetically identical animals sacrificed on different days. The labeled cells move from one area to another. In this way, the "inside-out" development of the cerebral cortex and most of the laminar patterns of development in other structures throughout the brain can be studied.

The research method involving axon transport of tritiated amino acids (proline and leucine) is used for cell connectivity experiments revealing physiological pathways in the brain. The cell bodies at the site of injection take up the labeled amino acids and transport them down their axons to the terminals, where silver grains will appear after autoradiographic tissue preparation. This method has been used extensively in studies of retina-to-cortex visual pathways.

Another neuroanatomical research method called degeneration is related to but different from autoradiography. A lesion is made at a site, and all of the axons that were connected to the dead cell bodies take up a silver stain, resulting in silver particles that have features similar to grains.

1.3 Background

The collection of quantitative data from autoradiographs involves two steps: first, recognition of the desired area for grain counting, and then counting of the number of silver grains in each area. Recognizing areas to be examined is easy for the human, but the counting
procedure is very tedious, and reproducibility of visual counts falls off rapidly with continuous counting and high grain densities. Therefore, the need for an automated grain-counting system exists.

One of the oldest and most widely used methods of obtaining grain density is the measurement of transmittance or absorbance using a photometer.\(^{(2-4)}\) However, this method does not lend itself to automated processes.

With the development of television video systems, scan-line analyzers such as the Leitz Classimat and the Metals Research, Ltd. (IMANCO) Quantimat, have been built. These have been combined with computers to form grain-counting systems based on grain count, grain area, and diameter measurements.\(^{(5-8)}\) However the researchers involved in scan-line analysis\(^{(5-8)}\) did not tackle the problem of full-slide grain counting. The Quantimat systems are fairly expensive ($50k to $100K) because of the use of costly very high resolution television cameras, which are difficult to interface and use quantitatively with a computer. The emphasis on elaborate hardware with wide sales appeal for multiple purposes and profitability makes these systems costly to the small biology laboratory.

A few research groups have attempted interfacing a computer to an automatic stage, thus giving the ability to scan under high magnification areas defined under low magnification. L.E. Lipkin\(^{(9)}\) has used a computer interfaced with a Quantimat using an elaborate console to control the modes of operation. He was concerned with single measurements
and did not scan entire slides. D.F. Wann and W.M. Cowan\(^{(10)}\) are the only researchers who have counted grains with respect to topology of the tissue.

Wann and Cowan used software grain recognition, similar to the Quantimat's anticoincidence point, as well as an automatic stage and television camera. Applying grain recognition in software (as opposed to hardware) is expensive in terms of scan time and results in throughput reduction. The disadvantages in using a television camera are spatial nonuniformity in sensitivity, limited tube life, complicated deflection circuits, and inflexible speeds, which complicate computer interfacing. Without the aid of an image preprocessor, the ability to set the video threshold for grain detection is limited and requires human visual intervention. Wann and Cowan have noted some of these disadvantages. With the advent of solid-state imaging devices and fast microscope stage control, there are advances to be made beyond this system, which is the best automatic grain counter to date.

Beginning in 1971, CSDL collaborated with Peter Neurath of Tufts New England Medical Center to start the development of a system to find and classify white blood cells. This included the development of a high-speed servo stage coupled to a special slow-speed television camera and display system.\(^{(11, 12)}\) Through this experience, limitations of television cameras were realized and the development of self-scan image line array photoreceptors was proposed. The development of the high-speed stage proved complex and costly. Therefore, fractional step-motor-control and acceleration-control algorithms were proposed to enable the
use of standard commercial microscope stages. These techniques have been implemented in an automatic metaphase-cell-location system under a consortium grant with Tufts New England Medical Center.

1.4 Rationale

It is evident that a general-purpose grain counter has yet to be built. There are common properties in present systems that should be incorporated into such a system, but no system built thus far has completely solved the question of quantitative autoradiography. This thesis shows that an inexpensive ($30k to $50k) rapid (100 mm$^2$/h) grain-counting system is feasible.

A much deeper question is what to do with the extracted information. At present, some neuroscientists would not know what to do with the grain-count information. Autoradiographs are discussed in terms of presence of grains and not quantity of grains. Then why build an automatic grain counter? The answer to this question lies in the future. The field of radiochemical tracer research in neuropharmacology has just begun to be explored. The ability to find brain feedback control networks related to specific neurotransmitters is foreseeable. The technology exists for studying every network of the brain, resulting in potential knowledge of total anatomical circuitry. At present, whole-brain studies are not done because researchers lack either an effective means of analyzing the data or an effective means to store and retrieve the amount of unquantized information that whole-brain
studies would require. With the data base this automatic grain-counting system is capable of creating, three-dimensional brain topology using existing computer graphics software can be displayed and manipulated, opening the field of neuroscience to mathematics and computer software science. Therefore, this system was not designed as much for present neuroscientific technology as for future neuroscientific research methods.

1.5 Specific Aims

The specific aim of this thesis project was to redesign and integrate the existing hardware and software of the metaphase-location system discussed in Chapter 1.3, to enable the creation of a rapid grain-counting system. The system's scan hardware consists of a linear photoreceptor array, giving a one-dimensional image, and a motorized microscope stage, giving the second dimension of the image. Video information is transferred to a computer through an interface. A preprocessor board makes video thresholding stable, a feature-extractor board makes rapid grain recognition possible, and an actuator translator board makes rapid stage movement possible. The grain-counting system can count a 100-mm² area in 45 minutes, using a 40-power objective, or it can count the same area in 4.5 hours, using a 100-power objective. The objective determines the size of the grain recognized, 0.3 μm at 100 power, and 0.6 μm at 40 power. It was necessary to redesign the preprocessor and feature-extractor boards of the metaphase-location system.
The software portion of the grain-counting system consists of a interrupt service program, motor control programs, a control box command interpreter, data-management programs, and data-analysis programs. These programs run under a Data General operating system called RDOS.

There were many possible areas for research in this project. Therefore, a subset of design tasks and concepts was chosen for the thesis. Emphasis is placed on fast large-area grain counting and the transferral of the count information to mass storage. Some analysis of count and scan accuracy is included. The area of data analysis is so large that only illustrative programs were written to show the potential of further information extraction. Data acquisition is the major concern in the body of the thesis.
CHAPTER 2

PROCEDURE

2.1 System Optomechanical Description

The automatic grain-counting system is an extension of an existing automatic cytology system development station at CSDL (see Figure 3). The automatic microscope development system consists of: a Diablo 2.5-megabyte disk, a paper-tape reader, a paper-tape punch, a NOVA 820 24k computer, a Tektronix CRT display, a Princeton Electronics scan converter, a Conrac video monitor, a Tektronix video hard copy, a Singer 300-character-per-second terminal, and a plotter. A white blood cell location system microscope stage with interface and a metaphase location system have been developed on this equipment.

The automatic grain-counting system components are: a Leitz-Dialux microscope with a Zeiss 10-µm stepper motor stage, a Fairchild 1024-element CCD line array, a computer interface consisting of seven custom-cards, a custom command control box, and computer hardware with a display from the development system.

The grain-counting system software runs under a Data General operating system called RDOS, and consists of a microscope command interpreter,
Figure 3. Automatic microscope development system.
actuator command generator, interrupt service, double-buffer management, and data-reduction programs.

Figure 4 is a block diagram of the optomechanical system.

The grain-counting system command flow (see Figure 5) starts with human intelligence input to the microscope command control box. The computer interprets these commands into actuator stage control and interface control signals.

The actuator command signals generated in minicomputer software by the actuator command generator and passed to the microcomputer actuator command translator software are turned into stepper motor drive voltages by the actuator driver interface card. The actuator commands can be automatic, as in the case of a scan command input, or they can be operator controlled by use of the joystick, as in manual command input.

The interface control signals consist of timing controls for different scan speeds, software parameter modification for different microscope objectives or information packing densities, and enable signals for different pieces of interface hardware operation.

The resultant data extraction from a commanded scan operation involves a hybrid scanning (patent pending) technique. The 1024-pel (picture element) array is electronically clocked into the interface, forming a one-dimensional scan. Upon completion of a scan line, the actuators are moved one mel (motion element), thus forming a two-dimensional scan. The mel size varies with the objective magnification.
Figure 4. System block diagram.
Figure 5. System command flow diagram.
One pel equals one pel divided by magnification power. One pel is $13 \times 13 \, \mu m$.

The system information flow (see Figure 6) starts with an unmodulated 100-watt halogen light source, which transmits photons through a stimulus filter attenuating the blue range. This provides higher contrast between grains and background for blue-stained slides. After the light is passed through the specimen and magnified by the optics, it is transformed into a modulated analog voltage, which is quantized into 1024 pels, representing a line of the optical density from the object. The pels are clocked into the image enhancement board of the computer interface. After passing through a preprocessing image filter, which differentiates and thresholds the signal into a binary 1024-pel bit stream, it enters the feature extractor. The feature extractor makes a hardware decision as to the presence or absence of a grain. All x-y coordinates of grains are buffered and data is channeled into the computer memory. A display of the grain coordinates is provided at this point in the information flow. Because there are billions of grains in a hemi-section scan, it is impossible to store all the grain coordinates. Data compression is accomplished by counting the number of grains in a small two-dimensional area called a block. The block counts are then buffered and transferred to disk storage. The analysis of the block grain counts can be done at a later time or even on a different computer. Display programs of the block-count storage are provided as the last step in the information processing.
2.2 Microscope Hardware

The microscope hardware consists of a Leitz-Dialux microscope (see Figure 7) with 6.3-, 25-, 40-, and 100-power objectives and a 100-watt halogen light source with various filters manually placed at the field diaphragm. The filter in use is a deep blue reject intended to increase contrast by blocking blue light, the predominant color in stained tissue.

The x-y actuator mechanism, providing two degrees of motion, consists of a Zeiss 10-μm step motor stage drilled to match Leitz mounting holes. The cables were rewired at the connector end to place pairs of windings in series, with phases within each pair reversed. This permits commanded sine and cosine waves to simulate the four sequential step pulses that the manufacturer recommends to drive the stage. The four pulses are the positive and negative lobes of the sine and cosine waves. Therefore, one sine or cosine lobe represents a 10-μm step pulse. The sine-cosine lobes are quantized into 64 levels, and each level is called a microstep pulse. The step motors can travel at 450 steps per second or 28,800 microsteps per second. With the scan rate at 357 lines per second, the stage travels at 390 μm per second at 40 power, and 780 μm per second at 100 power. With a maximum scan rate of 2500 lines per second, the stage travels at 780 μm per second, far below the maximum travel rate of 4500 μm per second. Therefore, the stage is not the limiting factor in the scan time limit. The light collecting time of the array sets the scan limit. This method of driving a stepper motor with microstep pulses is termed "fractional stepping".
Figure 7. Automatic microscope.
Figure 7. Automatic microscope.
The focus mechanism is a stepper motor custom-fitted to the manual fine-focus knurled knob (see Figure 8). The motor case is bolted to the microscope base via a plate, and the motor shaft turns the fine focus via a snug-fit adapter. The adapter was drilled for a safety stop, after several slides were broken by computer runaway. Another method of focus motor fit is to adapt the motor case to the coarse-focus knob and the shaft to the fine-focus knob, with no connection to the microscope base. When the maximum fine-focus travel is reached, the motor case starts to turn the coarse-focus knob, thus providing full dynamic focus range. The stage and focus motors were termed actuators to imply the inclusion of gearing and mechanisms necessary to provide eventual motion. If the mechanisms can be modeled, the drive signals can be adjusted to remove any nonlinearity in motion.

The photoreceptor linear array is a Fairchild 1024-element CCD array with support electronics (see Figure 9). The array and card can be purchased as a unit for less than $1k. The picture elements (pels) are semiconductor energy wells that collect electrons when hit with photons. They are spaced evenly at 13-\textmu m intervals, over a total length of 13.3 mm. At a scan line end, all elements are passed laterally to storage registers and clocked out serially to the interface. The scan rate is set at 357 lines per second or 366,000 pels per second, which is the minimum time required to collect light from the specimen.
Figure 8. Automatic focus mechanism.
Figure 9. CCD photoreceptor linear array.
Figure 9. CCD photoreceptor linear array.
2.3 Computer Interface Hardware

The computer interface hardware controls actuator movement, interprets image data from the line array, and transforms human intelligence input commands. The input-output channels are the computer, the actuators, the line array, and the control box. The computer interface is by far the most complicated element in the system and one that cannot be purchased. Microprocessor technology will greatly reduce the complexity and cost of design and construction in the near future.

The interface communicates with the computer by transferral of 16-bit memory words via input-output instructions. The computer input words from the interface are: x-joystick rate, y-joystick rate, z-joystick rate, command button enables, xend-yend address word, and xedge-yedge address word. The computer output words are: interface control word, x-actuator velocity, y-actuator velocity, z-actuator velocity, control box lamp enables, and video threshold.

The control box is composed of seven cards, each assuming a unique function described as follows:

(1) CARD2-ACTUATOR DRIVER—The current feedback motor drive circuits for x-y-z stage motion.

(2) CARD3-TIMING—Interface and array timing clocks, computer input-output word control, and cable drivers.

(3) CARD4-IMAGE PREPROCESSOR—The image processing filter, digital-to-analog conversion of threshold, and threshold operation.
(4) CARD5-FEATURE EXTRACTOR—Digital template control (edge, end, edgend), and template address word packing.

(5) CARD6-DMA CONTROL BUFFER—Direct memory access (DMA) control to the computer with $16 \times 64$-bit buffer management.

(6) CARD7-CONTROL BOX DRIVER—Input and pack command button signals, enable command button lamps, read joystick x-y-z analog voltages, analog-to-digital conversion, and pack joystick commands.

(7) CARD8-ACTUATOR COMMAND TRANSLATOR—Intel 8080 microprocessor with support chips responsible for position pulse train to driver card for x-y-z actuator movement control.

The location of the computer interface cards is shown in Figure 10.

There is a large need in biology and chemistry just to reliably move the slide around without any elaborate data-taking hardware. An automatic microscope stage controller would need only the interface box with reduced complexity to perform an effective role in stage random-position movement, programmed scan movement, or operator feedback-controlled movement. Technology is changing so fast that a dedicated automatic microscope controller will be cost effective for any scantly funded laboratory. Commercially available equipment and microprocessor technology is available, but design concepts for an automatic stage controller have not been made public.
Figure 10. Computer interface cards.
Figure 10. Computer interface cards.
2.3.1 **Control Box**

The control box (see Figure 11) is the means by which a human transfers his intelligence into the computer. The vocabulary is composed of 12 command buttons, and their definitions are programmed in the computer. The command set came about by programmer interaction with the system in an attempt to downgrade teletype dependence (i.e., so the teletype could possibly be eliminated). The command function selection is not intended to be optimal; merely sufficient to run the system. The function of each command button is briefly defined here; an operation description is presented in Chapter 2.5.

(1) **IDLE**—A state where no action is taken providing a return point from other function states.

(2) **MANUAL**—Manual mode; allows joystick velocity control of the x-y-z actuators.

(3) **SNG SCN**—Single scan mode; allows one swath scanning.

(4) **FOC SET**—Focus set; when multiplexed, gives automatic focus, maximum z in x direction, maximum z in y direction, and origin z value.

(5) **SCAN**—Scan; allows programmed automatic scanning in four increasing size limits.

(6) **ONE SET**—Sets the 1 bit for multiplexing modes.

(7) **TWO SET**—Sets the 2 bit for multiplexing modes.

(8) **PAR SET**—Sets system parameters via the teletype or joystick.
Figure 11. Command control box.
(9) MODE SET—Sets the feature extractor mode to no data, end points, edge points, or edgend points.

(10) SYS SPD SET—System speed set; divides the master system clock to provide slower system operation.

(11) MAG SET—Magnification set; adjusts the motor step size to match the pel size for the objective in use.

(12) PAK SET—Pack set; allows for variation in the block size for grain-count data reduction.

Each control box button has a lamp inside, which is illuminated when the computer recognizes a button has been pushed.

The joystick is included in the control box and consists of three variable resistors linked to a movable shaft. The angle of the shaft sets the resistance value. Two trim potentiometers are provided to allow for zero-position offsets. A large positive angle represents a fast actuator velocity command in the positive direction. Forward-back, right-left, and clockwise-counterclockwise provide x, y, and z motion, respectively.

2.3.2 Image Preprocessor

A difficulty in all grain-counting systems reviewed was where to place the threshold. The image plain of an objective lens does not provide a uniformly illuminated plain. The objective bends light and the center portion will be more intensely illuminated than the surrounding
portions of the image field. This results in a parabolic-shaped photon-intensity measurement independent of specimen material. The specimen modulation of light intensity is superimposed on this parabolic background. The overall magnitude of the signal is proportional to the summed intensities and varies considerably in magnitude but not in shape. The threshold is a flat signal, and when comparisons are made to the parabolic signal there could be many desirable setting levels. Most researchers\(^9, 10\) have relied on visual feedback for best threshold setting. If the parabolic shading could be removed and magnitude variations made constant, the threshold would have only one best setting and the signal would be normalized.

A method of normalizing is to differentiate the video signal. The low-frequency background shading does not vary throughout the slide, but the modulated higher frequency information does. If the low-frequency information is differentiated and the higher frequency information is passed and amplified, the signal will appear flat and the contrast will be enhanced. Very high frequency information is usually due to electronic noise, and should be integrated, resulting in a smoothing operation. The frequency at which information is only amplified, not integrated or differentiated, is called the "notch" frequency. An electronic device that can normalize or smooth a signal is a bandpass filter.

The design of the image-enhancement bandpass filter requires knowledge of the grain frequency in temporal, not spatial, units. The point spread of the optics indicates how many pels the grain will
be imaged on. As the array is clocked into the preprocessor card, the spatial pel elements can be regarded as temporal units and assigned a frequency. The notch of the bandpass filter is placed at this grain frequency. One difficulty with this technique is that the objective magnification determines the grain frequency, thus the notch of the filter must be moved with different objectives or large differences in grain sizes. The high-frequency break point of the notch need not be moved because the electronic noise remains constant. The low-frequency break point of the notch is critical to discrimination between cell bodies, staining, and grains. A circuit is shown that can vary the low notch breakpoint under program control. The system speed also changes the grain frequency even though the grains image on the same number of pels, because the entire line rate is changed.

Figure 12 shows the image enhancement filter, and Figure 13 is a photograph of the image preprocessor.

2.3.3 Feature Extractor

Because the item of interest is the existence of grains, not their optical density, only a binary, not a gray-level video signal, is required. (A gray-level signal is a white-gray-black representation of intensity.) The thresholding operation accomplished in the preprocessor converts gray level to a binary signal, which represents intensity in white or black. After thresholding, all pel units above threshold are
Figure 12. Image enhancement filter.
represented by a 1 (black), and all pel units below threshold are represented by a 0 (white). From the stream of black and white pel units, the existence of grains can be recognized.

Wann and Cowan\(^{(10)}\) used a software program that tagged the end of every continuous black blob. This was the end of grain mark or end point. The blob size did not matter; it mattered only that the blob shape was convex. The time required to do this processing was the limiting factor in material throughput of their system. They could scan 1 mm\(^2\) in 2 hours. The author chose to put this decision process in hardware, and a digital filter was designed to extract the end-of-blob property from the binary video.

A grain is assumed to be a circularly symmetric black body of 0.5 to 1.0 \(\mu m\) diameter. Using a 40-power objective, this corresponds to two pels in two dimensions. A second dimension is achieved by 1024-pel delay lines and single-pel storage units. At any point in time, a 4 \(\times\) 4 pel window can be examined which corresponds to a spatial 4 \(\times\) 4 window on the specimen. The 4 \(\times\) 4 pel window address is stored when a specific combination of black-white pel units enters the window. The author's original design of the black-white pel combination involved a "don't care" pel, followed by two black and one white pels in two dimensions. It was felt that this combination recognized the two pels above threshold required for 0.5-\(\mu m\) grains, but larger grain ends could also be recognized. The white pel is the stopper for continuous black-blob recognition. This combination generally worked, but it was
experimentally found to excessively count 45-degree black pel strings. The result was that cell bodies were counted. The combination was changed to the white, black, black, white pel series shown in Figure 14. This stopped the cell body count, but limited the grain count to those of exactly two-pel diameter. The error tradeoff favored this combination because with different magnifications all grain sizes could be counted with successive scans. The speed of this hardware decision process allows a 100-\(\text{mm}^2\) area to be counted in 1 hour. This is 200 times faster than Wann and Cowan's system.

Other digital filter modes are edge and edgend. The edge mode stores the address when a white-black or black-white combination is encountered in either dimension. This mode is convenient for imaging and is used in test slide scanning. The edgend mode was added to allow for variable-size grains or black bodies by storing the address if a concave or convex end is encountered. It has the expense of software-width calculation time. This mode was never used, but could be used for grain-size histograms to select the proper magnification for the faster end mode scans.

2.3.4 **Actuator Command Translator and Driver**

The minicomputer could deliver the microstep position increments to the actuator driver, but that would tax its computing capability, leaving no room for additional computation. This function has been transferred to a microcomputer for its implementation. As the polycomputer (i.e., a computer consisting of many computers) develops, methods
Figure 14. Feature extractor operation modes.
of modularizing computation for microcomputers will abound. The key piece of hardware that enables the precision and speed of scanning is the actuator command translator microcomputer.

Actuator system analysis is shown in Figure 15. The translator is passed a velocity command word from the minicomputer every 64-scan-line-interrupt period. The microcomputer successively adds this velocity word to a position word and uses the higher order byte to look up a function in a programmable read-only memory (PROM), which is a harmonically corrected sine wave. The result is an even distribution of position increments over the interrupt period.

The function burned into the PROM was obtained by taking the Fourier transform of the position error and subtracting the significant harmonics from the fundamental sine wave to decrease the error. This is a step-to-microstep error correction, and is not relevant for large stage motion.

Although the stage is a standard off-the-shelf item, the windings are wired somewhat differently. Unlike normal stepper action, where only one pole is activated at a time, this scheme has all four poles activated simultaneously with sine-cosine waves, and the drive current is incrementally passed from one pole to the next. i.e. poles act in a push-pull sequence instead of a pull-idle sequence. In this configuration, the motor can be thought of as a continuous permanent-magnet motor instead of a stepper motor.
Figure 15. Actuator system analysis.
The actuator system can be looked at in block diagram form (see Figures 15 and 16). The command rate from the actuator command generator is integrated in the translator to form the sine and cosine command angles. The winding voltages are formed by the actuator driver, resulting in commanded torque in the motor poles. The commanded torque times the electrical representation of the rotor torque plus any disturbance torque is the net driving torque to the rotor. Passing through friction and drag, the rotor obtains a resultant angle. The error is the commanded angle minus the resultant angle.

The actuator translator is described in detail for tutorial purposes and for replication. The schematic (shown in Figure 17) is complete, and a software description (with listing) is included in Chapter 2.5. Figure 18 is a photograph of the actuator command translator board component layout.

The actuator command translator block diagram illustrates the functional flow, whereas the translator schematic illustrates the components and wiring. Three velocity words are asynchronously loaded into 16-bit storage chips with a data-ready flag set. Three program loops stored in PROM monitors the flags for their enablement. The random-access memory (RAM) is used for scratch-pad calculations. The chip-select chips decode address bits for chip-enable signals and for output data-ready signals. The output data is on an 8-bit bus, and is latched into the actuator driver card by the data-ready enables.
Figure 16. Actuator system block diagram.
Figure 17. Actuator command translator schematic.
Figure 18. Actuator command translator card.
Figure 18. Actuator command translator card.
The actuator driver card (see Figure 19) consists of six identical circuits. The latching register holds the sine or cosine digital representation for digital-to-analog conversion. The current amplifier uses this analog voltage to drive the motor windings.

2.4 Computer Hardware

The computer hardware portion of the system includes a Nova 820 Data General 32k Computer, a Diablo 2.5-megabyte moving head disk, a Princeton Electronics scan converter, a Conrac high-resolution monitor, and a Tektronix video hard copy.

The operating system occupies 12k memory words, and the system software with buffers occupies 20k memory words. The system software was not optimized for low memory requirements and could be reduced in size. Overlays were used on some subroutines. The disk is used for data-file storage and in a small system could be replaced by a cassette tape or a soft disk unit.

The scan converter is an x-y addressable storage tube that has gray-level storage capability. The tube has support electronics for a video monitor that provides an image of the stored information. The raw-data grain printouts were made by storing a black point at the x-y address of every grain. Once the grain has been stored, its address cannot be recovered. The scan converter allows images involving billions of grains to be created where memory storage of these grain addresses would be impossible. The images in the data section were created in this
Figure 19. Actuator driver card.
manner using the Tektronix to make a hard copy of the image displayed on the Conrac monitor.

2.5 System Software

Basically, the software has to accomplish two tasks: control the actuators and handle the data. Data General's RDOS operating system has tasking capabilities. Tasking software means the computer can have two or more programs running simultaneously. The grain-counter system software runs under a two-task configured operating system. The tasks are MICROSCOPE COMMAND INTERPRETER, which is responsible for human command control interpretation, and RADOUT, a data-management program.

The MICMND (microscope command interpreter) program reads a 16-bit word from the command control box and jumps control to 12 different programs, each of which corresponds to a control-box button. Each subroutine program may or may not use a timing queue from the interrupt service program, which is hardware driven by the computer interface interrupt occurring every 64 scan lines. The tasks can be in active or wait states, and the timing queue tells the program to wait for the interrupt.

The control-box functions are multiplexed. ONE SET and TWO SET buttons allow four states for every mode button. The state selection must precede the mode selection. The modes, shown in Figure 20, are:

(1) IDLE—The idle mode initializes program flags and provides a no action state.
Figure 20. System operation commands.
(2) MANUAL—This mode allows operator control of the stepper stage actuators via a joystick. The command for x-y-z motion is velocity, not position. The perceptual rate is dependent on system speed set by another button.

(3) SNG SCN—Single scan mode is used when a through-focus series is desired from the same scan swath. This mode scans the same area four times at different focus positions.

(4) FOC SET—Focus set is the mode for manual focus selection. FOC(1) is the selection of origin in the x-y-z coordinate system. FOC(2) is the delta z in the maximum x direction. FOC(3) is the delta z in the maximum y direction. FOC(0) is automatic focus on retrace.

(5) SCAN—The automatic scan is implemented in successive swaths with a square scan area defined by the multiplex value, SCN(0) being the smallest. The operating system has two tasks to time share during this mode: calculate and update the motor velocity, and double buffer management, which includes data reduction and display.

(6) ONE SET—Sets the one bit in the multiplexor word.

(7) TWO SET—Sets the two bit in the multiplexor word. If one and two bits are set, the multiplexor word is three. If neither are set, the multiplexor word is zero. Thus, four states are achievable.
(8) PAR SET—System parameters are input in this mode. PAR(0) sets the threshold using the joystick. The other parameter modes are not used.

(9) MODE SET—The feature extractor mode is set with this button. MODE(1) is grain end, MODE(2) is black body edge, MODE(3) is edgend, MODE(0) is no data.

(10) SYS SPD SET—The master clock is divided down to provide slower scan rates: SYS(0), divide by one, SYS(3) divide by fifteen.

(11) MAG SET—The mel size is matched to the pel size for different objective use: MAG(0), one microstep equals one mel; MAG(1), two microsteps equal one mel; MAG(2); four microsteps equal one mel; MAG(3) sixteen microsteps equal one mel.

(12) PAK SET—The block size can be set to vary the resolution of the scan. PAK(0) is 25 μm², PAK(3) is 1600 μm².

During the scanning operation, a second task RADOUT is activated. It is in charge of double-buffer management, data compression, and raw-data display. The first task is to move the stage during scanning time.

A block diagram of the microscope command interpreter is given in Figure 21.

57
Figure 21. Microscope command interpreter.
2.5.1 **System Command Flow**

The control commands may be executed in any order, but the system may not work properly. There is a preferred system operation command flow (see Figure 22). The three basic operations are initialize, focus, and scan.

The initialization (see Figure 23) should involve setting system speed, setting magnification size, selecting feature mode, and setting packing block size. These may occur in any order, but must occur before other functions. The system speed for all test data was SYS(2). The magnification was MAG(2) (40-power objective). The mode was always MODE(1) unless test slides were used; then MODE(2) was also used. MODE(3) was never used and is not implemented.

The threshold set by PAR(0) need be set only once, unless an extremely different type of material is used. It was found that the preprocessor worked so well that the threshold could remain constant for all scans.

The second procedure in system operation is the manual focus definition (see Figure 24). The manual focus procedure should follow the sequence FOC(1), FOC(2), and FOC(3). An appropriate point, most likely selected under low magnification, should be chosen for the origin. It should be in the lower right-hand corner of the area to be scanned. The manual mode should then be entered to focus with the joystick. FOC(1) is set to define the scanning coordinates. The manual mode is then entered, and the maximum x scan distance traveled and focused.
Figure 22. Operation flow diagram.
INITIALIZE PROCEDURE

EACH +/- SET BUTTON MUST BE PRECEDED
BY NUMBER SET [1 2] OR [1 2]

Figure 23. Initialize procedure.
FOC(2) is then set defining the delta z increment in the x plane. The y distance is traversed and FOC(3) is set, defining the delta z in the y direction. A plane has now been defined to the computer for automatic focus during scan. This method of manual focus was used sometimes, but the scan was generally visually monitored with continuous operator focusing because it was found enjoyable to observe the material in motion, since the scan times were so short.

The FOC(0) automatic focus program uses retrace of the scan swath to form a list of focus points. The focus points are obtained by through-focus grain counts within an interrupt period, remembering the maximum grain-count position. It was never found necessary to go beyond the manual-focus program; therefore this method has never been tried. The method of peak finding is widely used in automatic focus programs.

The third and most important step in system command flow is the scanning operation (see Figure 25). Four scan-size areas are programmed. The parameter NWAIT is an integer variable counting the number of 64-scan-line interrupts. The ISWATH integer variable is the number of scan swaths desired. SCAN(0) is a 64 by 4, NWAIT by ISWATH area. SCAN(1) is 128 by 8, SCAN(2) is 256 by 16, and SCAN(3) is 1024 by 32 areas.

A scan swath is achieved by accelerating the x-axis actuator up to the scanning velocity. The scan velocity is in register with the array scan-line rate, thus forming a two-dimensional scan swath. Electrical scan in one dimension and mechanical scan in the other dimension
Figure 25. Scanning procedure.
form the hybrid scanning technique. At the end of the swath, the stage is decelerated. The data channel enable is not turned on until constant velocity is achieved, and it is turned off at the start of deceleration. Thus, linear pel size is guaranteed. The second scan swath is 1024 pels adjacent to the first swath. A retrace swath with no data-enable is used to position the array in the new x-y start location. The scan size is very flexible and can be defined easily interactively.

2.5.2 Actuator Control Software

The software for actuator control involves three steps: a FORTRAN SEARCH program, which sets up the number of swaths; start and stop locations of each swath; and the terminal velocity of the swath. This information is passed to the actuator command generator software, which forms acceleration lists to the specifications of the SEARCH program. The acceleration lists are used to update the velocity word passed through the interface to the microcomputer software in the actuator command translator. The actuator command translator software integrates the velocity word and outputs the position increments.

The actuator command generator (see Figure 26) forms a command velocity list by integrating a constant acceleration until the specified scan velocity is reached. The scan velocity then remains constant for the desired NWAIT periods. The deceleration is the velocity list output backward. The timing for the velocity updates is queued from the interrupt service program. If a random access position is desired, it can be easily obtained to a resolution of one velocity update period.
Figure 26. Actuator command generator.
by the integrated sum of the acceleration, deceleration, and constant velocity areas. If a finer position command is desired, one constant velocity period must be inserted at the appropriate time in the velocity list. The integrated area of the extra constant velocity period is called the residual. With an acceleration of one, only one period need be inserted. With any other acceleration, more than one period may be needed to form the residual. An illustrative program block diagram for a random access calculation is shown in Figure 26.

The actuator command translator (see Figure 27) software has a reasonably easy calculation, but it has to do it very fast. The translator software loads the velocity word at the interrupt rate, and adds this word to a position word, resulting in a function table pointer. The addition and microstep lookup are done at the array scan-line rate, where the velocity load is done at the 64-scan-line interrupt rate. The table pointer uses the higher order byte to allow for rates less than one microstep per scan line. At 40 power, jumps of two microsteps per scan line are output.

2.5.3 Data Compression Software

During scan operation, grain addresses are data channelled to the computer memory in a 2000-word area called ARRAY. The grain x-y address is packed into a 16-bit word; the right 10 bits are the y address (0 to 1024) of the pel grain location, and the left 6 bits are the x scan line address (0 to 63) within an interrupt period. When the ARRAY buffer is full, the software directs the data to a second 2000-word buffer called
Figure 27. Actuator command translator.
BARRAY. The ARRAY data is unpacked and sent to digital-to-analog converters for raw-data display on the scan converter. The grain addresses are then counted in a 16-bit word for block-count representation. The block counts are buffered for disk memory transfer in 64-word records. This process must be completed before the BARRAY buffer is full or an overflow is detected. The BARRAY data is compressed and displayed when the data input is switched back to the ARRAY buffer. Continuous x addressing is obtained by counting interrupts and adding a 64-line offset to the grain x address. The buffer interrupt marks are a -1, and buffer full marks are a -2. The present disk-packing format is 4 by 16 block counts per record. When pack set is fully developed, the record size and the packing density will be variable. Figure 28 is a block diagram of the data compression software.

2.5.4 Application Software

After a scan is completed, there is a disk file created with the grain counts per unit area stored in the predetermined packed record format. This data base may be manipulated in many forms; the analysis possibilities proved to large a task for this thesis. A two-dimensional display of the grain block counts as gray levels is shown in Chapter 3. Obviously, a dark block has more grains than lighter blocks.

A histogram program of the grain block-count file was included to illustrate some of the statistical possibilities. The histogram program is useful for determining peaks and valleys in grain density
Figure 28. Data compression software.
for background threshold location and anatomical area isolation. The program prints the mean and standard deviations, along with two histogram descriptors, skew and kurtosis.
CHAPTER 3

RESULTS

3.1 System Accuracy Results

3.1.1 Optical Transfer Characteristics

The optical transfer characteristics (see Figure 29) have been measured for the 100-power objective by imaging a 100-μm test slide that has bars spaced at 2-μm intervals. The test slide photo is the photograph of the test slide through the objective. The photoreceptor array response (Figure 29, upper left) is the measurement of the point-spread function of the test slide bars. Each 1-μm bar on the test slide images as a normal distribution on the sequentially spaced pels of the array. This measurement confirms that the grains give the same distribution and that the feature extractor model is correctly founded.

3.1.2 Preprocessor Performance

The preprocessor performance is confirmed photographically (see Figures 30 and 31) by showing the input and output waveforms of the image preprocessor. The grain image depicts the grains on a parabolic background intensity (top of Figure 30). The flat portion is readout time, and
Figure 29. Optical transfer characteristics.
Figure 30. Preprocessor image enhancement.
Figure 31. Dense grain preprocessor response.
represents no light present, i.e., zero voltage. The enhanced grain image (top of Figure 30) depicts the grain voltages amplified, low-frequency voltages attenuated, and very high frequency voltages integrated out. The magnified enhanced grain image (bottom of Figure 30) better illustrates the increase in contrast and the flatness of the signal. The flatness of the signal enables threshold independence of background illumination and shading.

The labeled cell image (Figure 31) shows grains tightly clustered (if not overlapping) superimposed on a nucleus response along with the parabolic background. The enhanced labeled cell image illustrates how the negative portion of the differentiated nucleus takes some of the grain responses below the flat portion of the signal therefore missed by the threshold. This phenomenon could account for missed grains overlying any small dark object or stain patch.

3.1.3 Feature Extractor Performance and Scanning Linearity

The performance of the END mode (or grain-count mode) and the EDGE mode of the feature extractor are depicted in the test slide scans shown in Figures 32, 33, and 34. The EDGE mode extracts only the edge points of any black object and is displayed as black dots or outlines. The END mode points are superimposed as small "t" shaped symbols. The 100-power objective gives a very linear scan, as can be seen by the roundness of the annulus image from the test slide (Figure 32). At this power, the mel size is as small as possible, thus giving the best approximation of continuous motion. The 16-power objective was used to find
Figure 33. Scan linearity—slow speed.
Figure 34. Scan linearity—fast speed.
the worst linearity error because the mel size is large (Figures 33 and 34), thus approaching stepper-type motion. Scan-linearity error at slow speed is most evident in the form of a wavy annulus image of the test slide. The scan linearity at fast speed (still at 16-power objective) is more linear, and gives a less wavy image because the momentum of the stage smooths the stepper motion into seemingly continuous motion.

3.2 Axon Transport Material Results

The raw data G1 scan hemi-section (Figure 35) shows heavy labeling in the dorsal and ventral nuclei of the lateral geniculate body of a Syrian hamster as a result of an injection of H2-protein in the contralateral eye. The G1 scan photo (Figure 36) shows the correspondence between the labeled areas in the actual tissue and the G1 scan dark areas. The G1 detail scan (Figure 37) shows the relationship between raw-data printout and block count printout. The darker blocks correspond to higher counts.

The retina projects to the lateral geniculate in a highly topological sharply defined two-layered pattern. The layers alternate between contralateral and ipsilateral retinal projections, although only two layers exist in the hamster. However, these layers are not evident in the photo or in the computer scan of this particular slide.

When counting grains over this large an area, it is nearly impossible to verify machine grain counts with visual grain counts. There are
Figure 35. Axon transport material—G1 scan.
Figure 36. Axon transport material—GI photo.
Figure 37. Axon transport material—Gl detail scan.
billions of grains represented in the raw-data display. But there is some counting of cell bodies and staining when they have the same spatial frequency and optical density as the grains. This error is verified by observing areas in the photo of G1 that show no grains and the corresponding areas in the G1 scan where grains are indicated. Theoretically, this error would also have been present in Wann and Cowan's system because their system is based on the same principles.

The superior colliculus photo G2 corresponds well with the block-count display G2 (Figure 38). The superior colliculus receives projections almost exclusively from the eye on the contralateral side. The contralateral retinal fibers terminate in the layer that is nearest to the surface. Labeling in a layer is slightly evident in the block-count display.

The HISTBLK program (Figure 39) shows that there are 8681 grains in this scan, with a mean block count of 2.09 grains. The standard deviation was 2.6 grains. The test for a normal distribution shows a positive skew, indicating a long tail toward the higher block counts. The kurtosis test indicates a platykurtic distribution, which has a flat rather than a normal distribution. Grains form distributions around anatomical areas, and the sum of these independent distributions forms a histogram. The block-count histogram shows maxima in the second differences at two, five, and seven grains per block. Every block including and below two grains is probably background. The grains between two and five indicate a common nodal distribution, possibly an anatomical area.
Figure 38. Axon transport material—G2.
HISTBLK
WHAT FILE NAME?  TOTAL = 8681
SCOO
MEAN= 0.209062E 1
STANDARD DEVIATION= 0.262297E 1
SKEW= 0.190213E 1
NORMALIZED SKEW= 0.447764E 0
KURTOSIS= 0.857591E 1
NORMALIZED KURTOSIS= 0.124649E 1
8IN  VALUE  ISTDIF  2NVDIF
 0  1598  -936  826
 1  662  -110  -53
 2  552  -163  82
 3  389  -81  -4
 4  308  -85  0
 5  223  -85  58
 6  138  -35  -7
 7  183  -42  24
 8  61  -18  6
 9  43  -12  1
10  31  -11  3
11  20  -8  1
12  12  -7  5
13  5  -2  3
14  3  1  -4
15  4  -3  4
16  1  1  -2
17  2  -1  2
18  1  1  -3
19  2  -2  3
20  0  1  -2
21  1  -1  1
22  0  0  0
23  0  0  1
24  0  1  -2
25  1  -1  1
26  0  1  0
27  0  0  0
28  0  0  0
29  0  0  0
30  0  0  0
31  0  0  0
STOP

Figure 39. Superior colliculus count histogram.
The grains in blocks between five and seven indicate a second anatomical area or the fringes of the first area. The grains greater than seven do not belong to a nodal distribution and therefore could be from any area. A valuable analysis program would be to enhance the nodal distributions in a display by zeroing all displayed grains not in the selected grain blocks, as suggested previously.

3.3 Cell Development Material Results

The cell development material photo shows the labeled layer in the neocortex with a labeled layer in the hippocampus. The dorsal thalamus is heavily labeled (see Figure 40). The cell development scan (Figure 41) shows the block-count display, the grain number image, and the raw-data display for the heavily labeled dorsal thalamus. The detail scan (Figure 42) shows the grain clusters over labeled cells. A program could count labeled cells by using intercluster grain distances as a discriminant. Tight grain clusters overlie cells and the centroid would indicate cell location.

Staining of the Nissel substance in cell bodies of the neocortex does interfere with the grain count. Comparing the photo (Figure 40) to the cortex scan (Figure 41 (upper left) and Figure 43) indicates many more grains displayed than actually exist. This particular stain (local fast blue) happens to have the same features as the grains. Unstained material would be desirable for grain counting.
Figure 41. Cell development material scan.
Figure 42. Cell development material detail scan.
Figure 42. Cell development material detail scan.
Figure 43. Cell development material cortex scan.
The histogram of the dorsal thalamus grain block count (Figure 44) has a large positive skew, indicating a long tail toward higher grain counts. The kurtosis is leptokurtic, indicating a sharp peak with long tails in the distribution. There seems to be no clear threshold to distinguish background grains.

3.4 Degeneration Material Results

An experimental application of the grain-recognition hardware is the scan of material stained for degenerating axons and axon terminals with the Fink-Heimer stain. A bleaching method has been developed which leaves only silver particles and the nucleoli of cells. The silver particles have properties similar to grains in photographic emulsions and can be counted. The repeatability of counts is questionable because the particles are distributed throughout the tissue, whereas grains in the previously discussed material are in the overlying emulsion.

The degeneration scan (Figure 45) shows terminal degeneration in areas connected to the substantia nigra which is the site of kainic acid injection. The pars compacta was spared. Layering can be observed in the detail degeneration scans of the superior colliculus (Figure 46).
HISTBLK
WHAT FILE NAME?
SIDE15.1
MEAN= 0.232194E 0
STANDARD DEVIATION= 0.707610E 0
SKW= 0.469784E 1
NORMALIZED SKW= 0.789237E 1
KURTOSIS= 0.386747E 2
NORMALIZED KURTOSIS= 0.760410E 2
BIN VALUE ISTD 2NDI
0 14188 -12748 11839
1 1440 -909 580
2 531 -329 210
3 202 -119 79
4 83 -40 10
5 43 -30 22
6 13 -8 6
7 5 -2 2
8 3 0 -3
9 3 -3 3
10 0 0 0
11 0 0 0
12 0 0 0
13 0 0 1
14 0 1 -2
15 1 -1 1
16 0 0 0
17 0 0 0
18 0 0 0
19 0 0 0
20 0 0 0
21 0 0 0
22 0 0 0
23 0 0 0
24 0 0 0
25 0 0 0
26 0 0 0
27 0 0 0
28 0 0 0
29 0 0 0
30 0 0 0
31 0 0 0

STOP

Figure 44. Cell development count histogram.
Figure 46. Degeneration material detail scan.
CHAPTER 4

CONCLUSION

An autoradiography grain-counting system has been designed and developed. The performance of a 100-mm\(^2\)-per-hour grain-counting system, with isolated grain density independence, is two hundred times faster and in theory as accurate as the leading grain-counting system.\(^{(10)}\) The system design was based on the experience of Wann and Cowan,\(^{(10)}\) the author's analysis of silver grain features, and automatic microscope research. A hardware pattern-recognition feature extractor has been implemented to make fast grain (END) counting possible. An image preprocessor provides grain detection independent of threshold setting and slide illumination. A small photoreceptor or picture element (pel) array provides fine-tuned spatial quantization to 0.13-\(\mu\text{m}\) pels (13 \(\mu\text{m}/100\times\)). The hybrid scanning technique of electrical scan in one dimension and mechanical scan in the other dimension requires incremental stage motion (0.15 \(\mu\text{m}\)). This motion is achieved by microcomputer control of a standard stepper microscope stage (10 \(\mu\text{m}/\text{step}\)) to the accuracy of 0.15 \(\mu\text{m}\) motion elements (mels).

The system was tested on three types of neurological material containing silver grains: radiochemically tagged proteins showing axon
transport; radiochemically tagged chromosomes showing cell division and migration; and silver-stained degenerated axons and terminals.

The axon transport material showed the most accurate results and gave the fewest false counts. The original design of the feature extractor, which allowed for variable-sized grains to be counted, recognized 45-degree cell bodies and staining, resulting in false counts. Therefore, it was changed to recognize specific size grains. The scans shown in the results of Chapter 3 used the original feature extractor. Variable-size grain recognition is accomplished by changing the objective magnification. This is a signal-versus-noise tradeoff that must always be made in engineering.

The feature extractor is adequate for its specific purpose, but a more general feature extractor would be a five-by-five pel arrangement, with each pel programmable to black (above threshold), white (below threshold), or "don't care" logical states. This form would allow experimentation with programming ease.

Cell development material scans were not as noise free as axon transport material scans. The stained cell bodies proved to be of the same spatial frequency and optical intensity as the grains. The error was not in the feature extractor. In its present design, the preprocessor caused grains to be pulled below threshold on the negative side (undershoot) of the differentiated cell body. The error caused fewer grains and more cell bodies to be counted. Thus, the signal-to-noise ratio was greatly reduced. In places where the staining was not as intense, the
grain counting was favorable and clusters of grains over cells were recognized. This system has the potential to perform labeled cell counting by the evident grain clusters, as shown in Chapter 3.3.

The preprocessor needs more experimentation on various stained material to help achieve the most general error-free design. An improvement now in progress is to take the absolute value of the signal, thus eliminating undershoot error.

The computer displays of degeneration material scans were of most value. Because the silver stain is located throughout the tissue, it was difficult to obtain a repeatable quantitative number of silver particles. The computer display provided a means of manipulating the image by contrast enhancement, background removal, etc., that could not have been accomplished photographically or visually. A quantitative measure can be extracted, but its value is questionable.

The computer display has great potential for image representation for publication. With the image in computer format, the field of computer graphics can be explored. The displays of the block count data were not as accurate as desired, because it was found that the display device was nonlinear. Therefore, the gray level represented did not linearly reflect the grain block count.

The testing of counting accuracy is difficult to measure because of the lack of a standard. The testing could prove to be quite an extensive project in terms of human grain-counting comparison. It is
felt that the real test in counting consistency will lie in the future use of different material scans for comparison. Accuracy has been tested on standard test slides and proved to be favorable; therefore, it was felt that theory of the grain-counting design has been verified.

It has been shown that an automatic autoradiography grain-counting system that is fast and accurate has the potential of being reproduced in an electronics laboratory with relative safety in the theory of operation and performance. With the development background of this thesis in mind, the question arises as to what form any future grain-counting system should assume. There are two paths to pursue: a very inexpensive system based on microcomputers, much less complex than the one presented here; or a highly complex system with time-shared microscopes, using a fast central processor tied to a central memory bank. The first form requires that each laboratory have its own system; the other form requires that the material be sent to a central laboratory for scanning and the data filed in a multiple-access memory system. The author's preference tends toward the individual laboratory system because of convenience and flexibility in its application.

The individual laboratory system can be of two types: a highly specific totally inflexible version with no computer image storage, using on-line graphic display; or a more flexible system with soft disk storage and computer graphic software. The author's preference lies with the somewhat more expensive flexible system because its programs can be expanded and adapted to other applications.
The inexpensive flexible system could be designed with relative ease, based on the work presented in this thesis. The interface box could be greatly reduced in its complexity by using microprocessors and large-scale integration semiconductor chips, which have become common in the past few years. The central computer could be replaced by a fast microcomputer built into the interface box, thus eliminating the timing interface difficulties. Various FORTRAN graphics packages available from software houses could be stored on the soft disk for image-analysis purposes.
LIST OF REFERENCES


LIST OF REFERENCES (Cont.)


14. Merker, Björn, MIT Psychology Department, Personal Communication.


17. Carpenter, M.B., Human Neuroranatomy, Williams and Wilkins, Baltimore, Maryland.


APPENDIX

SOFTWARE SOURCE LISTING

TYPE GRAIN.LK
DELETE GRAIN.LD
RLDR/N 5/C 2/K GRAIN.LD/L GRAIN/S MICMND ↑
MICRO COUNT [ISCAN, JSCAN, KSCAN, LSCAN] RADOUT UFLOAT ↑
STOPAR LMOVE LSTEP LACCEL ↑
LTRAJEC LACCLST DWSMULDIV ISORT IMPY IDYD ↑
FMT. LB FORT. LB
R

103
PROGRAM MICMND
MICROSCOPE CONTROL BOX INTERPRETER

BUTTON 1, 2, 3 ... (IDLE) (MANUAL) (SINGLE SCAN)
BUTTON 4, 5, 6 ... (FOCUS) (SCAN) (ONE)
BUTTON 7, 8, 9 ... (PARAMETER) (MODE) (TWO)
BUTTON 10, 11, 12 ... (SYSTEM-SPEED) (MAGNIFICATION) (PACK SET)

PARAMETER IXVEL=4, IYVEL=5, IZVEL=6, ILAMP=7, ITHRES=8
PARAMETER IXJOY=14, IYJOY=15, IZJOY=16, IBUTT=17, IFULL=27
PARAMETER IX=22, IY=23, IZ=24, IVONE=28, ICONT=3, IXH=18, IYH=19
PARAMETER IENOM=26, IMWC=9, IEMNIO=1

INTEGER BLOCK (0:45), AKRAY(0:2048), CRRAY(0:1024), BRRAY(0:2048),

INTEGER IORDER(4), IXFOC(4), IYFOC(4), IZFOC(4)
INTEGER LX (15), LY (15), YINC, LZ (15)
EXTERNAL RAVOUT
LOGICAL LOKG
LOGICAL LFOC
LOGICAL LSIINT
LOGICAL LPAK
LOGICAL LTEST

COMMON /GCNT/ ICNT, JCNT, KCNT
COMMON /CKRAY/ CKRAY
COMMON /SCN/ISCAN, JSCAN, NWAIT, NSWATH
COMMON /SPU/ MAG, ISP
COMMON /ARY/ MWU, NWU
COMMON /BU/ MMWC, NMWC, MAND, MBNU, MSW
COMMON /DISP/ IDISP, JDISP, KDISP
COMMON /KAY/ MKEY, MSCK, IPT, MSTK
COMMON /BLK/ BLOCK
COMMON /CBLK/ MODE, LRVS, IBIAS, IGAIN, ISCANSY, ISYSSY, IRW
COMMON /BRAY/ BRRAY
COMMON /AKRAY/ AKRAY
COMMON /CXLIM/ IXLIMIT
COMMON /CYLIM/ IYLIMIT
COMMON /CZLIM/ IZLIMIT
COMMON /CISVEL/ ISVEL
COMMON /CHOC/ IOWUEK, IXFOC, IYFOC, IZFOC
COMMON /CJOY/ IJOY
COMMON /CJACCEL/ JACCEL, JZACCEL
COMMON /CUB/ IUB, IDUB
COMMON /CDEV/ NOUT, NIN
COMMON /CLTEST/ LTEST
COMMON CYMID, YINC
NMAX = 15

initialize

CALL LIMIT (BLOCK)
CALL FTASK (RAVOUT, $105, 0)
CALL PWI (1)

WRITE (NOUT, 1)
FORMAT (', AUTOKARIOGRAPHIC GRAIN COUNTER, 8/9/78, ')

NMET = 0
JButt = 0
Kyutt = 0
LFUNC = 0
ISPD = .TRUE.
ISCAN = 0
MAG = 0
ISC = .FALSE.
JSC = .FALSE.
BLOCK (ITHKES) = 110
LONG = .FALSE.
LFOC = .FALSE.
LSINT = .FALSE.
LHAR = .FALSE.
MDES = 0
LAMP = 0
IGAIN = 0
IBIAS = 0
ISCANSP = 0
ISYSSP = 1
IRW = 0

CALL LCNT (BLOCK, MODE, LAMP, IBIAS, IGAIN, ISCANSP, ISYSSP,
XCHANGE = UFLOAT (IXFOC(2)) - UFLOAT (IXFOC(1))
YCHANGE = UFLOAT (IFYFOC(3)) - UFLOAT (IFYFOC(1))
FDIST = .1 * XCHANGE
XMAX = UFLOAT (IXLIMIT)
YMAX = UFLOAT (IYLIMIT)
IZMIN = - IZLIMIT
IF (LTEST) GO TO 519
IF (.NOT. LHAR) GO TO 90
LHAR = .FALSE.
GO TO 200
C (1) IDLE ... GET BUTTONS, SET LAMPS

90  B:LOCK (ILAMP) = 1K OR JBUTT
100  CALL REC (B:LOCK (I_DONE), MES)
    IF (B:LOCK (IJBUTT) EQ. 0) GO TO 100
105  KBUTT = B:LOCK (IJBUTT)
    B:LOCK (ILAMP) = B:LOCK (ILAMP) OK KBUTT OK JBUTT
    DO 120 I = 1, 12
    IF (KBUTT EQ. 1) GO TO 130
120  KBUTT = KBUTT / 2
    GO TO 90
130  IF (I .NE. 12) L:FUNC = 0
    GO TO (90, 200, 300, 400, 500, 600, 700, 800, 900, 1000

C (2) MANUAL ... STAGE UNDER JOYSTICK CONTROL

200  B:LOCK (ILAMP) = 2K OR JBUTT
    L:FUNC = 3
210  JXVEL = 0
    JYVEL = 0
    JZVEL = 0
211  CALL REC (B:LOCK (I_DONE), MES)
C
    IXPOS = B:LOCK (IXJOY)
    IYPOS = B:LOCK (IYJOY)
    IF (IJOY .LE. 1) GO TO 216
C
216  IF (IABS (IXPOS) GT. I08) GO TO 220
    IXPOS = 0
    GO TO 225
220  IXPOS = IXPOS - ISIGN (IV8, IXPOS)
    IF (IJOY LE 1) IXPOS = IXPOS / 2
225  IF (IABS (IYPOS) GT. I08) GO TO 230
    IYPOS = 0
    GO TO 231
230  IYPOS = IYPOS - ISIGN (IO8, IYPOS)
    IF (IJOY EQ. 1) IYPOS = IYPOS / 2
C
231  IUIFF = IXPOS - JXVEL
    IF (IABS (IUIFF) GT. JACCEL) GO TO 232
    JXVEL = IXPOS
    GO TO 233
232  JXVEL = JXVEL + ISIGN (JACCEL, IUIFF)
233  IF (NOT LONG) GO TO 234
    X = U:LOAD (B:LOCK (IX)) + U:LOAD (B:LOCK (IYVEL) + JXVEL)
    IXNEXT = B:LOCK (IX) + B:LOCK (IXVEL)
    IF (X LT. 0) JXVEL = - IXNEXT
    IF (X GT. XMAX) JXVEL = IXLIMIT - IXNEXT
234  BLOCK (IXVEL) = JXVEL
C
  IDIFF = IVPOS - JYVEL
  IF (IABS (IDIFF) .GT. JACCEL) GO TO 236
  JYVEL = IVPOS
  GO TO 237
236  JYVEL = JYVEL + ISIGN (JACCEL, IDIFF)
237  IF (.NOT. LOKG) GO TO 238
  Y = UFLOAT (BLOCK (IY)) + FLOAT (BLOCK (IXVEL) + JYVEL)
  IYNEXT = BLOCK (IY) + BLOCK (IXVEL)
  IF (Y .LT. 0.) JYVEL = - IYNEXT
  IF (Y .GT. YMAX) JYVEL = IYLIMIT - IYNEXT
238  BLOCK (IXVEL) = JYVEL
C
  IZPOS = BLOCK (IZJOY)
  IF (IABS (IZPOS) .GT. IDIZ) GO TO 240
  IZPOS = 0
  GO TO 250
240  IZPOS = IZPOS - ISIGN (IDIZ, IZPOS)
  IZPOS = IZPOS / 6
  IF (IZJOY .NE. 0) IZPOS = IZPOS / 2
250  IDIFF = IZPOS - JZVEL
  IF (IABS (IDIFF) .GT. JZACCEL) GO TO 251
  JZVEL = IZPOS
  GO TO 252
251  JZVEL = JZVEL + ISIGN (JZACCEL, IDIFF)
252  IF (.NOT. LOKG) GO TO 253
  IZNEXT = BLOCK (IZ) + BLOCK (IZVEL)
  IZNEW = IZNEXT + JZVEL
  IF (IZNEW .LT. IZMIN) JZVEL = IZMIN - IZNEXT
  IF (IZNEW .GT. IZLIMIT) JZVEL = IZLIMIT - IZNEXT
253  BLOCK (IZVEL) = JZVEL
  IF (BLOCK (IYUTT) .EQ. 0) GO TO 211
  BLOCK (IXVEL) = 0
  BLOCK (IXVEL) = 0
  BLOCK (IZVEL) = 0
  GO TO 105
C
C  (3) SNG SCN SINGLE SCAN 0,1,2,3
C
300  BLOCK (ILAMP) = 4K.0W. JYUTT
  GO TO 210
(4) FOCUS ... NEXT 4 POINTS 'MARKED' TAG

400   LFUNC = 1
421   CALL REC(BLOCK(IVONE),MES)
   IF(BLOCK(IYUTT).NE.0) GOTO 421
   IF(ISCAN .NE. 0) GOTO 430
   ACCEPT "ZINC=".IOKDEK(2)
   GOTO 460
430   IF(ISCAN .NE. 1) GOTO 440
   LF=1
   MK=1
   BLOCK(IKH)=0
   BLOCK(IYH)=0
   BLOCK(IZH)=0
   BLOCK(IK)=0
   BLOCK(IY)=0
   BLOCK(IZ)=0
   LOKG= .TRUE.
   IZOC(1)=BLOCK(IZ)
   GOTO 460
440   IF(ISCAN .NE. 2) GOTO 450
   LF=2
   MK=2
   V=IAUS(ISVEL)
   X1=BLOCK(IKH)
   X2=BLOCK(IK)
   X=X1*(2.**16)+X2
   NWAIT=X/V
   IZOC(2)=BLOCK(IZ)
   IOKDEK(1)=(IZOC(1)-IZOC(2))/NWAIT
   WRITE(10) NWAIT,IOKDEK(1)
   GOTO 460
450   IF(ISCAN .NE. 3) GOTO 460
   LF=3
   MK=3
   V=IAUS(VINCE)
   V1=BLOCK(IYH)
   V2=BLOCK(IY)
   Y=Y1*(2.**16)+Y2
   NSWATH=Y/V
   IZOC(3)=BLOCK(IZ)
   IOKDEK(2)=IZOC(1)-IZOC(3))/NSWATH
   WRITE(10) NSWATH,IOKDEK(2)
   GOTO 460
460   CALL REC(BLOCK(IVONE),MES)
   IF(BLOCK(IYUTT).NE.0) GOTO 460
   GOTO 210
   GO TO 210
C (5) SCAN ... SCAN SLIDE 0, 1, 2, 3
C
500 IF (LWG) GO TO 510
WRITE (NOUT, 2)
2 FORMAT ('PLEASE MARK ORIGIN')
501 CALL REC (BLOCK, IDONE, MES)
IF (BLOCK (IBUTT), NE. 0) GO TO 501
GO TO 90
510 BLOCK (ILAM0) = 20K OR. JButt
CALL LKINT (BLOCK, $530)
519 KX = 0
KY = 0
KKVEL = -ISVEL
LSINT = .FALSE.
MSTR = 3
NWD = 2048
MWD = 2048
IPT = 0
CALL LIN1AD (BLOCK, HKRAY, NWD, 8KRAY, NWD)
CALL DCLR
CALL SVALUE (BLOCK, KX, KY, KKVEL, NWAIT, NSWAT, $530
MSC = 5
CALL XMT (MKEY, MSC)
KBUTT = 1
GO TO 540
C
530 MODE = 0
CALL LCONT (BLOCK, MODE, LAMP, IBIAS, IGAIN, ISCANS, ISYSS, 
KKVEL = BLOCK (IXVEL)
KBUTT = BLOCK (IBUTT)
BLOCK (IXVEL) = 0
BLOCK (IXVEL) = 0
CALL REC (BLOCK, IDONE, MES)
KX = BLOCK (IX)
KY = BLOCK (IY)
LSINT = .TRUE.

540 MOVE = 0
CALL LCONT (BLOCK, MODE, LAMP, IBIAS, IGAIN, ISCANS, ISYSS, 
CALL LKINT (BLOCK, 0)
IF (KBUTT .EQ. 2) GO TO 200
GO TO 90
(6) ONE ... PARAMETER SET 2**0

CALL REC(BLOCK(IIVONE),MES)
IF(BLOCK(IIBUTT).NE.0) GOTO 600
ISC=.NOT.ISC
IF( .NOT.ISC) GOTO 650
ISCI=ISCI.OR.000001K
IIBUTT=IIBUTT.OR.40K
GOTO 105

650
ISCI=ISCI.AND.000002K
IIBUTT=IIBUTT.AND.177377K
GOTO 105

(7) PARAMETER ... SET SYSTEM PARAMETERS 0, 1, 2, 3

700
IIC=IIC+100K.0K.IIBUTT
CALL REC(BLOCK(IIVONE),MES)
BLOCK(IITKRES) = ( BLOCK(IITKRES) + BLOCK(IIXJOY)/16 ) .AND. 000777K
IF( BLOCK(IIBUTT).EQ.0) GOTO 700
TYPE "THRESHOLD = \". BLOCK(IITKRES)
GOTO 105
GOTO 105

(8) MOVE ... SET MOVE

800
CALL REC(BLOCK(IIVONE),MES)
IF(BLOCK(IIBUTT).NE.0) GOTO 800
IF(.LE.0) MODE=000000K
IF(.EQ.1) MODE=020000K
IF(.EQ.2) MODE=040000K
IF(.EQ.3) MODE=060000K
GOTO 105

(9) TWO ... PARAMETER SET 2**1

900
CALL REC(BLOCK(IIVONE),MES)
IF(BLOCK(IIBUTT).NE.0) GOTO 900
JSC=.NOT.JSC
IF( .NOT.JSC) GOTO 950
ISCI=ISCI.OR.000002K
IIBUTT=IIBUTT.OR.400K
GOTO 105

950
ISCI=ISCI.AND.000001K
IIBUTT=IIBUTT.AND.177377K
GOTO 105
TYPE MICRO
SP: .TITL MICRO

; LAST REVISED 8/23/78

; ENTRIES:
; .COMM TASK, 2*400+2
; .EXTD CPYL, FRET, RTL
; .EXTN UIEX, IXMT, RELC
; .EXTN SIM80, SIMSV
; .ENT LIMIT, LMMV, LCONT, LREAD, LSTOP, LSTRT
; .ENT SIMFL, CONTW, LOCT, VCLK
; .ZWEL

SIMFL: 0 ; 8080 SIMULATION FLAG
CONTW: 00001 ; CONTROL WORD
; NREL

; DEVICE CODE AND MASK
; .DUSK MENT=25
; .DUSK XYD=23
; .DUSK MMASK=177577

; FORTRAN STACK POINTER AND BLOCK ADDRESS
; .DUSK FSP=16
; .DUSK BLOCK=-167
; .DUSK NOK=JMP +1
; .DUSK NBLOCK=0
; .DUSK MEMO=1
; .DUSK MPXQ=2
; .DUSK CONT=3

; .DUSK XVEL=4.
; .DUSK YVEL=5.
; .DUSK ZVEL=6
; .DUSK LAMPW=7.

111
(*) SPEED ... CHANGE SYSTEM SPEED

1000 CALL REC(BLOCK(IDONE),MES)
   IF(BLOCK(IBUTT) NE 0) GOTO 1000
   IF(ISCAN EQ 0)ISYSSP=1
   IF(ISCAN EQ 1)ISYSSP=3
   IF(ISCAN EQ 2)ISYSSP=7
   IF(ISCAN EQ 3)ISYSSP=15
   CALL LCONT(BLOCK,MOVE,LAMP,IBIAS,IGAIN,ISCANSP,ISYSSP,IR
   GOTO 105

(0) MAG ... CHANGE OBJECTIVE MAGNIFICATION

1100 CALL REC(BLOCK(IDONE),MES)
   IF(BLOCK(IBUTT) NE 0) GOTO 1100
   MAG=ISCAN
   IF(MAG EQ 0) YINC=1200
   IF(MAG EQ 1) YINC=1100
   IF(MAG EQ 2) YINC=1100
   IF(MAG EQ 3) YINC=886
   IF(MAG EQ 0) LWVS=15
   IF(MAG EQ 1) LWVS=7
   IF(MAG EQ 2) LWVS=3
   IF(MAG EQ 3) LWVS=1
   GOTO 105

(*) MAWX ... RECORD STAGE COORDINATES FOR CURRENTLY ACTIVE

1200 IF (LFUNC GT 0) GO TO 1210
   WRITE (NOUT,4)
   4 FORMAT (' NO FUNCTION IS ACTIVE')
1201 CALL REC(BLOCK(IDONE),MES)
   IF(BLOCK(IBUTT) NE 0) GOTO 1201
   GOTO 90

1210 GOTO 90

GOTO 105
END
DUSK
THRES=8.
DUSR
MCW=9.
DUSR
SP70=10.
DUSR
MEMI=11.
DUSR
MPX=12.
DUSR
STAT=13.
DUSR
XJOY=14.
DUSR
YJOY=15.
DUSR
ZJOY=16.
DUSR
YUTT=17.
DUSR
XM=18.
DUSR
YM=19.
DUSR
ZH=20.
DUSR
INTC=21.
DUSR
X=22.
DUSR
Y=23.
DUSR
Z=24.
DUSR
SFLAG=25.
DUSR
ENDM=26.
DUSR
FULL=27.
DUSR
DONE=28.
DUSR
LM=29.
DUSR
LX=30.
DUSR
AC2=31.
DUSR
AC3=32.
DUSR
SB000=33.
DUSR
XVEL20=34.
DUSR
YVEL20=35.
DUSR
ZVEL20=36.
DUSR
INT80=37.
DUSR
XPOS=38.
DUSR
YPOS=39.
DUSR
ZPOS=40.
DUSR
TEMP=41.
DUSR
TEMP1=42.
DUSR
XVOLU=43.
DUSR
YVOLU=44.
DUSR
ZVOLU=45.
DUSR
END8=ZVOLU.

DUSR
BLK=13.

LUCT:
STSAV
17777777
METS
STSAV:
BLK 10
BLOCK:
BLK 1

113
SIMSV:  SIMSV
LUCT:  LUCT
OMASK:  17777
ENHO:  177776
ENSO:  177777
CM:  6

BETSV:  LDA  2, LUCT
        MOV  2, 1
        LDA  2, LBLK, 2
        STA  1, AC2, 2
        STA  3, AC3, 2
        MOV  2, 3
        LDA  0, SFLAG, 3
        MOVZR#  0, 0, SZC
        JMP  MSTWT
        DIBC  2, MET
        STA  2, MPXI, 3
        VIA  2, MET
        STA  2, MEMI, 3
        LDA  0, CONT, 3
        MOVL#  0, 0, SZC
        JMP  +3
        JMP  MCONT
        JMP  +1
        LDA  0, ENVS
        STA  0, 0, 2
        LDA  0, @MSW
        MOVZR  0, 0, SZC
        JMP  @XMT

AXMT:  LDA  0, SFLAG, 3
        MOVL#  0, 0, SZC
        JMP  AT
        LDA  0, MEMI, 3
        LDA  1, @MANV
        SUBZL#  0, 1, SNC
        JMP  MEMOK

AT:  LDA  0, ENVS
     STA  0, 1, 2
     LDA  0, MKEY
     LDA  1, @MPWC
     STA  1, MWC, 3
     SUBZL  1, 1
     STA  1, @MSW
     LDA  1, @MSTK
     IXMT
    HALT
        LDA  2, BMEM
        JMP  MSTWT+1

114
8XMT: LDA  0, SFLAG, 3
       MOV  0, 0, SZC
       JMP  BT
       LDA  0, MEMI, 3
       LDA  1, @MNU
       SUBZL  0, 1, SNC
       JMP  MEMOK
BT:   LDA  0, BENU
       STA  0, 1, 2
       LDA  0, MKEY
       LDA  1, @ANWC
       STA  1, MWC, 3
       SUB  1, 1
       STA  1, @MSW
       LDA  1, @MSTR
       INC  1, 1
       IXMT
HALT
       LDA  2, AMEM
       JMP  MSTR+1
AMEM: GADV  ARAY, 0
       BMEM: GADV  BRAY, 0
       ANWC: GADV  BUF, 0
       BMWC: GADV  BUF, 1
       MSW:  GADV  BUF, 4
       MNU:  GADV  BUF, 2
       MNU:  GADV  BUF, 3
       MKEY: GADV  RAD, 0
       MSTK: GADV  KAY, 3
       MAG:  GADV  SPD, 0
       MEMOK: INC  2, 2
       V0AS  2, MET
       STA  2, MEMO, 3
       JMP  MKEYAV

MSTR: LDA  2, MEMO, 3
       LDA  3, LUCF
       LDA  3, LBLK, 3
       LDA  0, C6M
       DO8  0, MET
       LDA  0, MWC, 3
       DO8  0, MET
MCONT: V0AS  2, MET

STA  2, MEMO, 3
MREAD:   SUB 2, 2
          MOV 2, MET
          VIC 1, MET
          STA 1, STAT, 3
          LDA 0, CONT, 3
          VDC 0, MET
          INC 2, 2
          MOV 2, MET
          VIC 1, MET
          STA 1, XJOY, 3
          LDA 0, XVEL, 3
          MOVZL 0, 0
          MOVZL 0, 0
          LDA 1, OMAG
          MOV# 1, 1, SNK
          JMP M2 5
          MOVZK# 1, 1, SNK
          JMP M25
          MOVZK# 1, 1, SNC
          JMP M40
          JMP M100

M2 5:   MOV 0, 1
          MOVZL 0, 0
          MOVZL 0, 0
          MOVZL 0, 0
          MOVZL 0, 0
          MOVZL 0, 0
          ADV 1, 0
          JMP XOUT

M25:   MOV 0, 1
          MOVZL 0, 0
          ADV 1, 0
          JMP XOUT

M40:   MOVZL 0, 0
          JMP XOUT

M100:  JMP XOUT

XOUT:   VDC 0, MET
          INC 2, 2
          MOV 2, MET
          VIC 1, MET
          STA 1, YJOY, 3
          LDA 0, YVEL, 3
          MOVZL 0, 0
          MOVZL 0, 0
          LDA 1, OMAG
          MOV# 1, 1, SNK
          JMP N25
          MOVZK# 1, 1, SNK
          JMP N25

116
MOVZ 1,1
JMP N40
JMP N100
N25:
MOV 0,1
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
ADD 1,0
JMP YOUT
N40:
MOVZ 0,0
JMP YOUT
N100:
JMP YOUT
YOUT:
DOC 0, MET
INC 2,2
DB 2, MET
DIC 1, MET
STA 1, ZJOV, 3
LDA 0, ZVEL, 3
LDA 1, 8MAG
MOV# 1,1, SNK
JMP 22.5
MOVZR# 1,1, SNK
JMP 225
MOVZ# 1,1, SNC
JMP 240
JMP 2100
Z25:
MOV 0,1
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
MOVZ 0,0
ADD 1,0
JMP ZOUT
Z25:
MOV 0,1
MOVZ 0,0
ADD 1,0
JMP ZOUT
Z40:
MOVZ 0,0
JMP ZOUT
Z100:
JMP ZOUT
ZOUT:
DOC 0, MET
INC 2,2
DOB   2, MET
VIC   1, MET
STA   1, BUTT, 3
LDA   0, LAMPW, 3
DOC   0, MET
INC   2, 2
DOB   2, MET
VIC   1, MET
INC   2, 2
DOB   2, MET
VIC   1, MET
LDA   0, SP70, 3
DOC   0, MET
LDA   0, XWOLVD, 3
LDA   2, WH, 3
LDA   1, W, 3
AVUZ  0, 1, SZC
INC   2, 2
STA   1, W, 3
STA   2, WM, 3
LDA   0, YWOLVD, 3
LDA   2, WY, 3
LDA   1, WY, 3
AVUZ  0, 1, SZC
INC   2, 2
STA   2, WY, 3
STA   1, Y, 3
LDA   0, ZWOLVD, 3
LDA   2, ZH, 3
LDA   1, Z, 3
AVUZ  0, 1, SZC
INC   2, 2
STA   2, ZH, 3
STA   1, Z, 3
MCNT: LDA   0, XVEL, 3
STA   0, XVOLVD, 3
LDA   0, YVEL, 3
STA   0, YVOLVD, 3
LDA   0, ZVEL, 3
STA   0, ZVOLVD, 3
LDA   0, S8080, 3
MOV   0, 0, SZW
JSK   0, SIM80
ISZ    INTCT, 3
NOP
LDA    0, LKRET, 3
MOV#   0, 0, SNR
JMP    DISM
LDA    0, BUIT, 3
MOV#   0, 0, SNR
JMP    DISM
HALT
DISM:
LDA    0, DONE
ADD    3, 0
SUBZL  1, 1
IXMT
NOP
SUBZL  1, 1
LDA    3, LC1
LDA    3, LBLK, 3
LOAD   2, AC2, 3
LDA    3, AC3, 3
UIEX

LC1:    LC1T
SIM80:  SIM80

DONE:   DONE

LINIT - IDENTIFIES SCANNER TO OPERATING SY

CALLING SEQUENCE:

CALL LINIT (BLOCK)

BLOCK - CONTROL BLOCK ADDRESS

VUSK     CNT=-166

2
LINIT:
JSK    @ CPYL
LDA    0, MET
LDA    1, DCTAD
LOAD   2, PHAGE
SSTHM
IVEF
JMP    E1
LDA    0, MET
SSTHM
DE8L
NOP
LOAD   0, XYO

119
Systm

LDA 0, CPU
Systm

LDA 3, FSP
LDA 2, BLOCK, 3
LDA 3, LDC1
STA 2, LBLK, 3
LDA 0, LENG
LDA 3, FSP
STA 0, CNT, 3
SUI 0, 0
STA 0, 0, 2
INC 2, 2
VSZ CNT, 3
JMP -3
LDA 2, BLOCK, 3
LDA 0, NWORD
STA 0, NBLOCK, 2
LDA 0, SYMFL
STA 0, SBOOBO, 2
LDA 0, CONTW
STA 0, CONT, 2
SUBZL 0, 0

LSET:
STA 0, SFLAG, 2
INTDS
JSW @ MTSV
LDA 3, FSP
LDA 2, BLOCK, 3
SUI 0, 0
STA 0, SFLAG, 2
STA 0, INTCT, 2
INTEN
JSW @ FRET

180+081+0811+111815

E1:
JSW @ WTEW
JSW @ FRET

MET: MET
XYD: XYD
CPU: CPU
MTSV: MTSV
DCTR: LCT ; +180
DPAGE: 29
LEN: ENDB-NBLOC+1
NWORD: ENDB-NBLOC+1

120
LRMV - REMOVES SCANNER FROM OPERATING SYSTEM

CALLING SEQUENCE:
CALL LRMV (BLOCK)
BLOCK - CONTROL BLOCK ADDRESS

1.

LRMV:
JSR  @ CPYL
LOA  2, BLOCK, 3
SUBZK 0, 0
STA  0, $FLAG, 2
INTOS
JSR  @ METSV
INTEN
LOA  0, $ET
SYSTM
LRMV
JMP  E2
JSW  @ FRET

188+081+40 811+11815
E2:
JSR  @ WETK
JSW  @ FRET

LSTWT - RESTARTS LOCATOR

1.

LSTWT:
JSR  @ CPYL
LOA  2, BLOCK, 3
SUBZL 0, 0
JMP  LSET

LSTOP - STOPS LOCATOR

1.

LSTOP:
JSR  @ CPYL
LOA  2, BLOCK, 3
SUBZK 0, 0
JMP  LSET

T - ASSEMBLES CONTROL WORD

ING SEQUENCE:
CALL LCONT (BLOCK, MODE, LAMP, IBIAS, IGAIN, ISCANSP, ISYSSP, I

BLOCK - CONTROL BLOCK ADDRESS
MODE - SCAN MODE (0-3)
IBIAS - BIAS CALIBRATION SWITCH (0-1)
IGAIN - GAIN CALIBRATION SWITCH (0-1)
ISCANSP - SCAN SPEED (0-15)
ISYSSP - SYSTEM SPEED (0-15)
IRW 0 DATA CHANNEL ENABLE

121
LCONT:  

JSR @ CPYL
LDA 2, @INW, 3
ADDZL 2, 2
LDA 0, @MOVE, 3
ADDZL 0, 2
MOVZL 2, 2
LDA 0, PEV
ADDZL 0, 2
LDA 0, @LAMP, 3
ADDZL 0, 2
LDA 0, @BIAS, 3
ADDZL 0, 2
LDA 0, @GAIN, 3
AUDS 0, 2
LDA 0, @SCANSY, 3
ADDZL 0, 0
ADDZL 0, 0
LDA 1, @SYSSY, 3
AUD 1, 0
AUD 2, 0
LDA 2, @BLOCK, 3
STA 0, CONT, 2
JSR @ FRET

PEV: 0
INW: 0

LREAD - READS DATA IN CURRENT MODE

CALLING SEQUENCE:
CALL LREAD (BLOCK, AKWV, MWV, BKRAY, NWV)

BLOCK - CONTROL BLOCK ADDRESS
AKWV - DATA ADDRESS
MWV - LENGTH OF DATA ARRAY
BKRAY - SECOND BUFFER ADDRESS
NWV - SECOND BUFFER LENGTH

DUSK ARRES=166
DUSR MWV=-165
DUSR BKRAY=-164
DUSW NWV=-163
LKRD:

JSW       @ CHYL
LDA       2, BLOCK, 3
LDA       0, AWAY, 3
STA       0, MEMO, 2
VDA       0, MET
LDA       1, @MWD, 3
ADU       1, 0
STA       0, ENUM, 2
LDA       1, K256
SUB       1, 0
STA       0, @ HEND
LDA       1, @MWD, 3
NEG       1, 1
STA       1, MWC, 2
STA       1, @ AMWC
LDA       0, K6
VDB       0, MET
VCC       1, MET
LDA       2, AWAY, 3
SUB       0, 0
STA       0, 0, 2
INC       2, 2
INC       1, 1, SZR
JMP       -3
LDA       2, AWAY, 3
LDA       0, @ HEND
STA       0, 0, 2
LDA       0, BKW, 3
LDA       1, @MWD, 3
ADU       1, 0
LDA       1, K256
SUB       1, 0
STA       0, @ MWD
LDA       1, @MWD, 3
NEG       1, 1
STA       1, @ AMWC
LDA       2, AWAY, 3
SUB       0, 0
STA       0, 0, 2
INC       2, 2
INC       1, 1, SZK
JMP       -3
LDA       2, AWAY, 3
LDA       0, @ HEND
STA       0, 0, 2
SUB       0, 0
STA       0, @ MSW
JSW       0, WET

BCLR:

LDA       0, CHYL
LDA       2, BLOCK, 3
LDA       0, AWAY, 3
STA       0, MEMO, 2
VDA       0, MET
LDA       1, @MWD, 3
ADU       1, 0
STA       0, ENUM, 2
LDA       1, K256
SUB       1, 0
STA       0, @ HEND
LDA       1, @MWD, 3
NEG       1, 1
STA       1, MWC, 2
TYPE SEARCH
C
SUBROUTINE SEARCH (BLOCK, JX, JY, JXVEL, NWAIT, NSWATH,
PARAMETER IXVEL=4, IYVEL=5, IDONE=28, IZVEL=6
PARAMETER IX=22, IY=23, IFULL=27, IZ=24
PARAMETER IBUTT=17, ICONT=3
C
SUBROUTINE SEARCH (BLOCK, JX, JY, JXVEL, NW, NS, IRET)
C
EXTERNAL IOV, JOY, KOV, LOV
INTEGER BLOCK (0:45), ARRAY (0:2048), BARRAY (0:2048)
INTEGER NAMEBUF (6)
INTEGER JARRAY (0:15, 0:63), KARRAY (0:3, 0:63), LARRAY (0:63)
INTEGER CRARRAY (0:1024), MARRAY (0:15, 0:63)
INTEGER XLIST (2, 20), YLIST (2, 20)
INTEGER XLFIRST (2, 20), YLFIRST (2, 20)
INTEGER XLAST (2, 20), YLAST (2, 20)
INTEGER XLIMIT, YLIMIT, ZINC
INTEGER IORDER (4), IXFOC (4), IYFOC (4), IZFOC (4)
COMMON /COBLK/ MODE, LRVS, IBIAS, IGAIN, ISCANSP, ISYSSP, I!
COMMON /SCN/ ISCAN, JSCAN, NWAIT, NSWATH
COMMON /GCNT/ ICNT, JCNT, KCNT
COMMON /DISP/ IDISP, JDISP, KDISP
COMMON /RAO/ MKEY, MSCR, IPT, MSTR
COMMON /ARY/ MWD, NWD
COMMON /BARRAY/ BARRAY
COMMON /BUFFER/ MMWC, NMWC, MBND, MSW
COMMON /ARRAY/ ARRAY
COMMON /CXLIMIT ARRAY
COMMON /CYLIMIT ARRAY
COMMON /CYWID/ YINC
COMMON /CFOC/ IORDER, IXFOC, IYFOC, IZFOC
COMMON /CRARRAY/ CRARRAY
COMMON /NAMEBUF/ NAMEBUF
EQUIVALENCE (MARRAY (0, 0), ARRAY (0))
EQUIVALENCE (JARRAY (0, 0), CRARRAY (0))
EQUIVALENCE (KARRAY (0, 0), CRARRAY (0))
EQUIVALENCE (LARRAY (0), CRARRAY (0))
C
COMPUTE ABSOLUTE VELOCITY
C
IV = IAABS (JXVEL)
IF (IV.EQ. 0) RETURN
TYPE "WHAT FILE NAME=
READ (11, 100) NAMEBUF (1)
100 FORMAT (S8)
JSCAN = ISCAN + 1
GOTO(110,150,180,190),JSCAN
CALL FOYLD(2,IOV,0,IERR)
ICNT=0
JCNT=0
KCNT=0
ICLR=-2048
IDISP=ICLR
JDISP=ICLR
KDISP=ICLR
NSWATH=4
NWAIT=64
GOTO 200

150 CMLL FOYLD(2,JOV,0,IERR)
JCN=0
ICNT=0
KCNT=0
ICLR=-4096
NSWATH=8
NWAIT=128
IDISP=ICLR
JDISP=ICLR
KDISP=ICLR
GOTO 200

180 CALL FOYLD(2,KOV,0,IERR)
ICNT=0
JCNT=0
KCNT=0
ICLR=-8192
IDISP=ICLR
JDISP=ICLR
KDISP=ICLR
NSWATH=16
NWAIT=256
GOTO 200

190 CALL FOYLD(2,LOV,0,IERR)
ICNT=0
JCNT=0
KCNT=0
ICLR=-16384
IDISP=ICLR
JDISP=ICLR
KDISP=ICLR
NSWATH=32
NWAIT=512
GOTO 200

200 IV=IABS(JXVEL)
CALL DIR("DP1",IERR)
CALL RESET
IF (IERR. NE. 1) TYPE "DSKERR=" , IERR
IPT=1
CALL FOPEN (IPT, NAMEBUF)
IF (IV EQ 0.0) RETURN

COMPUTE LIMITS, TIME, AND TRAJECTORY FOR BASIC TURN AROUND MANE
ASSUME SEARCH TRAJECTORY ALWAYS PASSES THROUGH (0, 0).

JMODE = 1
IDAT = .TRUE.
ISWATH = 0
KXEND = IMPY (NWAIT, IV, 0) ; ENDING X COORD
KYEND = IMPY (NSWATH, YINC, 0) ; ENDING Y COORD
NSWATH=N NSWATH+2

CALL LACCEL (0, 0, 0, IV, 0, 0, 0, 0, 0, NXRUNS, XLIST, NYSRUNS, YL

CALL LACCEL (0, 0, 0, 0, 0, YINC, 0, NXRUNS, XLIST, NYSRUNS, YLIST, YT

CALL LACCEL (0, 0, 0, KYEND, 0, 0, 0, NXRUNS, XLIST, NYSRUNS, YLAST, YLIST, NTL

SUCCEEDING SWATHS -- ACCELERATE, WAIT

IZFOC(3)=IORDER(1)
IZFOC(4)=-IORDER(2)
BLOCK(I2)=IZFOC(1)

500 IF (IDAT) BLOCK(ICON) = ISYSSP
IF (IDAT) CALL REC(BLOCK(IDONE), MES)
CALL LSTEP (BLOCK, NXRUNS, XLIST, NYSRUNS, YLIST, NTIME, JM
IF (NWAIT .LT. 1) GO TO 1000

C

IF (IDAT) BLOCK(ICON)=100000K. OR. MODE. OR. ISYSSP
BLOCK(IZVEL)=IZFOC(3)
DO 600 I = 1, NWAIT
CALL REC(BLOCK(IDONE), MES)
CONTINUE

BLCK(IZVEL)=0
IF (IDAT) CALL LSTOP(BLOCK)
IF (IDAT) BLOCK(ICON) = ISYSSP
IF (IDAT) CALL REC(BLOCK(IDONE), MES)
IPLACE =-JPLACE
CALL LSTEP (BLOCK, NXRUNS, XLIST, NYSRUNS, YLIST, NTIME, IPLACE, 1
IF (IDAT) GOTO 1000
IF (JSCAN. NE. 9) GOTO 900
BLOCK(IZVEL)=IZFOC(4)
DO 910 I=0, 64
CALL REC(BLOCK(IDONE), MES)
CONTINUE

910 BLOCK(IZVEL)=0
GOTO 1000
BLOCK(IZVEL)=IZFOC(4)
CALL REC(BLOCK(IDONE), MES)
BLOCK(IZVEL)=0
CALL LSTEP(BLOCK, NXFIRST, XLFIRST, NYFIRST, YLFIRST, NTFIRST, 

C
C SET UP FOR NEXT SWATH, CHANGE VELOCITY SIGN
C
1000 ISWATH = ISWATH + 1
BLOCK(ICONT)=LRVS
CALL REC(BLOCK(IDONE), MES)
IF (ISWATH .GE. NSWATH) GOTO 1500
IF(IDAT) GOTO 1001
GOTO 1010
1001 GOTO (1100, 1100, 1200, 1200, 1200), MSTR
1100 GOTO 500
1200 IDISP=ICLR
JDISP=ICLR
CALL LREAD(BLOCK, ARRAY, MWD, BRRAY, NWD)
KIDISP=KIDISP+1024
1010 JMODE = -JMODE
IZFOC(3)=-IZFOC(3)
IDAT=.NOT.IDAT
GO TO 500
GOTO 1500
1500 CONTINUE
IDISP=ICLR
JDISP=ICLR
KIDISP=ICLR
CALL LSTEP(BLOCK, NXLAST, XLLAST, NYLAST, YLLAST, NTLAST, 1, 1)
GOTO(1510, 1520, 1530, 1540), JSCAN
1510 CALL FOVRL(IOV, IERR)
IF(IERR.NE.1) TYPE "DSK ERR", IERR
GOTO 1550
1520 CALL FOVRL(JOV, IERR)
IF(IERR.NE.1) TYPE "DSKERR="", IERR
GOTO 1550
1530 CALL FOVRL(KOV, IERR)
IF(IERR.NE.1) TYPE "DSKERR="", IERR
GOTO 1550
1540 CALL FOVRL(LOV, IERR)
IF(IERR.NE.1) TYPE "DSKERR="", IERR
GOTO 1550
1550 ENDFILE IPT
CALL DIR("DP0", IERR)
IF(IERR.NE.1) TYPE "DSKERR="", IERR
RETURN
END

128
TYPE RADOUT
PARAMETER IMEMI=11, IMEM0=1, IENDM=26, IMWC=9, ISTAT=13, I6
TASK RADOUT
EXTERNAL IOV, JOV, KOV, LOV
INTEGER BLOCK(0:45)
COMMON /SCN/ ISCAN, JSCAN, NWAIT, NSWATH
COMMON /NAMEBUF/ NAMEBUF
COMMON /DISP/ IDISP, JDISP, KDISP
COMMON /BLOCK/ BLOCK
COMMON /RAO/ MKEY, MSCR, IPT, MSTR
COMMON /CRRAY/ CRRAY
COMMON /GCNT/ ICNT, JCNT, KCNT
COMMON /ARRAY/ ARRAY
COMMON /ARRAY/ BARRAY
INTEGER ARRAY(0:2048), CRRAY(0:1024), JRRAY(0:15, 0:63), 8
EQUIVALENCE (JRRAY(0, 0), CRRAY(0))
CALL OVOPNC(2, "GRAIN.OL", IERR)
5 CALL REC(MKEY, MSCR)
GOTO(10, 20, 30, 40, 50, 50, MSCR
10 CALL WRBLK(1, IPT, ARRAY, 8, IERR)
   IPT=IPT+8
   GOTO 5
20 CALL WRBLK(1, IPT, BARRAY, 8, IERR)
   IPT=IPT+8
   GOTO 5
30 CALL JAUTO(BLOCK, ARRAY, IDISP, JDISP, KDISP)
   IDISP=JDISP
   GOTO 5
40 CALL JAUTO(BLOCK, BARRAY, IDISP, JDISP, KDISP)
   IDISP=JDISP
   GOTO 5
50 CALL DIR("DP1", IERR)
   IF(IERR .NE. 1) TYPE "DSKERR=", IERR
   CALL RESET
   CALL FOPEN(IPT, NAMEBUF)
   REWIND IPT
   GOTO(200, 600, 600, 600), JSCAN
200 DO 105 K=0, 64
   DO 101 I=0, 1024
101 CRRAY(I)=0
   CONTINUE
   READ BINARY (1, END=600) CRRAY
   DO 90 I=0, 15
   DO 95 J=0, 63
   K=9
91 IVAL=JRRAY(I, J)/(1.5)**K
   IF(IVAL.GT.0) GOTO 92
   K=K-1
   IF(K.GE.0) GOTO 91
90 CONTINUE
GOTO 95
JRRAY(I,J)=K
CONTINUE
CONTINUE
WRITE(10,100) ((JRRAY(I,J), J=0, 63), I=0, 15)
100 FORMAT (1X, 64I1)
105 CONTINUE
GOTO 5
600 CALL FCLOS(IPT)
TYPE "SCAN DONE"
CALL DIR("0P0", IERR)
IF (IERR .NE. 1) TYPE "VSERR=", IERR
GOTO 5
END
.TITLE JSCAN
.ENT JOV
.ENT JAUTO
.EXTO CPYL, FRET, IOCAT
.NREL
.DUSR XYD=23
.DUSR MET=25
.DUSR BLOCK=-167
.DUSR ARRAY=-166
.DUSR IDISP=-165
.DUSR JDISP=-164
.DUSR KDISP=-163
5.

JAUTO:  JSR @CPYL ; GE1
SUB 0, 0
STA 3, POINT
LOA 0, ZVAL
DOC 0, XYD
LOA 2, BLOCK, 3
LOA 1, ARRAY, 3
LOA 2, @IDISP, 3
STA 2, XOFF
LOA 0, @KDISP, 3
MOV 0, 3
STA 3, YOFF
STA 1, ADDR2

DNXT:  LOA 0, @ADDR2
ISZ ADDR2
MOV 0, 1
INC 1, 1, SNR
JMP NLINE
INC 1, 1, SZR
JMP +3
DORC 2, XYD
JSR @FRET
LOA 1, C176
ANDS 0, 1
MOVZR 1, 1
MOVZR 1, 1
LOA 2, XOFF
ADD 2, 1
MOVZR 1, 1
MOVZR 1, 1
MOVZR 1, 1
DOR 1, XYD
LOA 2, @CNT
INC 2, 2
STA 2, @CNT
JMP +1

131
COMOL 1, 1, SZC
MOV 1, 1, SZR
JMP .+2
JSR @.FRET
LDA 1, C177
AND 0, 1
MOV 1, 0
MOVZR 0, 0
MOVZR 0, 0
MOVZR 0, 0
MOVZR 0, 0
MOVZR 0, 0
MOVZR 0, 0
LDA 3, YCNT
ADD 0, 3
LDA 0, 0, 3
INC 0, 0
STA 0, 0, 3
JMP .+1
LDA 3, YOFF
ADD 3, 1
MOVZR 1, 1
MOVZR 1, 1
MOVZR 1, 1
DOBS 1, XYD
JMP DNXT

NLINE:
LDA 0, C64
LDA 2, XOFF
ADD 0, 2
LDA 3, POINT
STA 2, @JOISP, 3
STA 2, XOFF
LDA 0, YCNT
LDA 1, YEND
SUBZ# 0, 1, SNC
JMP NOW
LDA 1, C16
ADD 1, 0
STA 0, YCNT
JMP DNXT

NOW:
LDA 0, YSTR
STA 0, YCNT
LDA 1, @IPT
MOVZL 1, 1
LDA 2, IOCAT
ADD 1, 2
LDA 2, 0, 2
LDA 1, C377
AND 1, 2

132
LDA 1, @JCNT
.SYSTM
.WRR CPU
HALT
LDA 1, @JCNT
INC 1, 1
STA 1, @JCNT
SUB 0, 0
LDA 2, YCNT
STA 0, 0, 2
INC 2, 2
ISZ K64
JMP -.3
LDA 2, SK64
STA 2, K64
JMP DNXT

ADDR2: 0
C377: 000377
SK64: -100
K64: -100
JCNT: .GADD GCNT, 1
C177: 001777
C176: 176000
C16: 20
C64: 100
ZVAL: 40
POINT: 0
XOFF: 0
YOFF: 0
IPT: .GADD RAD, 2
YSTR: .GADD CARRAY, 0
YEND: .GADD CARRAY, 57
YCNT: .GADD CARRAY, 0
CNT: .GADD GCNT, 0
.END
TYPE METAPROM
PRG 2000H ;
XCNT: EQU 0403H ; X PORT CONTROL ADDRESS
YCNT: EQU 0803H ;
ZCNT: EQU 0C03H ; Z PORT CONTROL ADDRESS
XOUT: EQU 4003H ; PORT ADK
YOUT: EQU 4008H ;
ZOUT: EQU 4013H ;
MASK: EQU 22H ; TEST PC1 OR PC5 BITS
FLAG: EQU 0402H ; READ XPORT FOR DOC TEST
XVEL: EQU 0400H ; VELOCITY PORTS
YVEL: EQU 0800H ;
ZVEL: EQU 0C00H ;
XPOSN: EQU 2400H ; RAM POSITION STORAGL
YPOSN: EQU 2402H ;
ZPOSN: EQU 2404H ;
CLEAR: DI ; HARDWARE RESET INTERUPT
MVI C. 60H ; SET INTERNUPT COUNT
MVI A. 8FH ; SET CONTROL I/O
STA XCNT ;
STA YCNT ;
STA ZCNT ;
LXI H. 0000H ; CLEAR RAM POSITION STORAGE
SHLD XPOSN ;
SHLD YPOSN ;
SHLD ZPOSN ;
SHLD XVEL ;
SHLD YVEL ;
SHLD ZVEL ;
LXI H. 248FH ; RAM ADR FOR STACK
SPHL ;
LP: LDA FLAG ; WAIT FOR FOR DOC PULSE
ANI MASK ;
JZ LP ;
EI ; ENABLE INTERRUPTS
WT: MOV A, A ; NOOPERATION
JMP WT ; WAIT FOR INTERRUPT
ORG 2038H ; START OF INTERRUPT SERVICE PROGRAM
DI ;
VCR C ; TIMER FOR DEADMAN
JZ CLEAR ; NO DOC SO CLEAR
LDA FLAG ; LOOK FOR DOC
ANI MASK ;
JZ XLOAD ;
MVI C. 60H ; INIT DEADMAN L. 5(75HZ)
XLOAD: LHLD XVEL    ;LOAD DOUBLE PRECISION VELOCITY
XCHG ;
LHLD XPOSN    ;LOAD TRIG XPOSITION POINTER
LAD D    ;POSITION PLUS DELTA POSITION
SHLD XPOSN    ;UPDATE POSITION POINTER
MOV L, H    ;HIGH POSITION INTO LOW TABLE
MVI H, 21H    ;HIGH TABLE
MOV D, M    ;GET XSIN VALUE
MVI H, 22H    ;COS TABLE
MOV E, M    ;LOAD XCOS VALUE
XCHG ;
SHLD XOUT    ;OUTPUT XSIN AND XCOS

YLOAD: LHLD YVEL    ;LOAD Y VELOCITY
XCHG ;
LHLD YPOSN    ;
LAD D    ;
SHLD YPOSN    ;
MOV L, H    ;POSITION HIGH TO TABLE LOW
MVI H, 21H    ;TABLE HIGH
MOV D, M    ;
MVI H, 22H    ;COS TABLE
MOV E, M    ;
XCHG ;
SHLD YOUT    ;OUTPUT Y SIN COS

ZLOAD: LHLD ZVEL    ;LOAD Z VELOCITY
XCHG ;
LHLD ZPOSN    ;
LAD D    ;
SHLD ZPOSN    ;
MOV L, H    ;POSITION HIGH TO TABLE LOW
MVI H, 21H    ;TABLE HIGH
MOV D, M    ;
MVI H, 22H    ;COS TABLE
MOV E, M    ;
XCHG ;
SHLD ZOUT    ;Z SIN COS OUTPUT
EI    ;ENABLE INTERRUPTS
RET    ;RETURN TO WAIT LOOP
END ;
TYPE HISTBLK
   INTEGER ARRAY(0:15, 0:3)
   INTEGER BARRAY(0:31), CARRAY(0:31), DARRAY(0:31)
COMMON /ARRAY/ ARRAY
COMMON /BARRAY/ BARRAY
COMMON /NAMEBUF/ NAMEBUF(8)
TYPE "WHAT FILE NAME?"
READ(11, 100) NAMEBUF(1)
100 FORMAT(68)
   CALL DIR("DP1", IERR)
   CALL FOPEN(1, NAMEBUF)
   REWIND 1
   ISREC=0
   NREC=1
   IDX=16
   IDY=16
   IVAL=20
   X1=0
   X2=0
   X3=0
   X4=0
   TOTAL=0
5   READ BINARY (1, EN=200) ARRAY
   DO 10 J=0, 3
   DO 20 I=0, 15
      TOTAL=TOTAL+1
      XTMP=ARRAY(I, J)
      X1=X1+XTMP
      X2=X2+XTMP**2.0
      X3=X3+XTMP**3.0
      X4=X4+XTMP**4.0
   N=ARRAY(I, J)
   BARRAY(N)=BARRAY(N)+1
20   CONTINUE
10   CONTINUE
   GOTO 5
200   X1=X1/TOTAL
   X2=X2/TOTAL
   X3=X3/TOTAL
   X4=X4/TOTAL
   MEAN=X1
   STDEV=(X2-X1**2.0)**(0.5)
   SKEW=(X3-3.0*X1*X2+2.0*(X1**3.0))/STDEV**3.0
   KURT=(X4-4.0*X1*X3+6.0*X2*(X1**2.0)-3.0*(X1**4.0))/STDEV**4.0
   SKURT=KURT/STDEV**2.0
   BSKEW=SKEW/(STDEV**3.0)**0.5
TYPE "MEAN= ", MEAN
TYPE "STANDARD DEVIATION= ", STDEV
TYPE "SKEW= ", SKEW
TYPE "NORMALIZED SKEW= ", BSKEN
TYPE "KURTOSIS= ", KURT
TYPE "NORMALIZED KURTOSIS= ", BKURT

DO 210 I=0, 30
CRAY(I)=BARRAY(I+1)-BARRAY(I)
210 CONTINUE
DO 220 I=0, 30
DARRAY(I)=CRAY(I+1)-CRAY(I)
220 CONTINUE

TYPE " ", "BIN ", "VALUE ", "ISTOIF ", "2NDOIF
DO 230 I=0, 31
TYPE I, BARRAY(I), CRAY(I), DARRAY(I)
230 CONTINUE
CALL DIR( "DP0", IERR)
END
TYPE DBOX

10 IU=25
   IV=25
   IX=0
   IV=0
   IX=0
ACCEPT "SIZE=", SIZE
   SIZE=SIZE+5-1
   IV=0
ACCEPT "CLEAR I/O", ITM
   IF(ITEM.EQ.0) GOTO 20
   CALL DCLK

20 DO 100 I=0, SIZE
   IV=1(I)
   CALL DGBK(IVAL, IX, IV, IOX, I0Y)
   IX=IX+IOX
100 CONTINUE

200 DO 300 I=1, SIZE
   IV=2(I)
   CALL DGBK(IVAL, IX, IV, IOX, I0Y)
   IV=IV+I0Y
300 CONTINUE

300 DO 400 I=2, SIZE
   IV=3(I)
   CALL DGBK(IVAL, IX, IV, IOX, I0Y)
   IX=IX-IOX
400 CONTINUE

400 GOTO 10
END
TYPE DISPBLK
INTEGER DARRAY(0:15, 0:1024)
COMMON /NAMEBUF/ NAMEBUF(8)
COMMON /DARRAY/ DARRAY
10 TYPE "WHAT FILE NAME?"
READ(11, 100) NAMEBUF(1)
100 FORMAT(S8)
TYPE "BACKGROUND NUMBER=
ACCEPT IBACK
TYPE "AUTO SCAN NUMBER=
ACCEPT JSCAN
JSCAN=JSCAN+1
GOTO(110, 120, 130, 140), JSCAN
110 NWAIT=64
 IWAIT=4
 IOX=16
 IOY=16
 GOTO 150
120 NWAIT=128
 IWAIT=8
 IOX=8
 IOY=8
 GOTO 150
130 NWAIT=256
 IWAIT=16
 IOX=4
 IOY=4
 GOTO 150
140 NWAIT=512
 IWAIT=32
 IOX=2
 IOY=2
 GOTO 150
150 CALL DIR("OP1", IERR)
CALL FOPEN(1, NAMEBUF)
CALL DCLR
REWIND 1
IY=0
IX=0
II=1024
DO 90 K=1, IWAIT
IF(II.LT.1024) GOTO 60
II=0
READ BINARY (1, END=60) DARRAY
60 IY0=IY
IY1=IY
IX0=IX
DO 80 I=1, NWAIT
DO 70 J=0, 15
IVAL=10*(DARRAY(J, II)-IBACK)
IF(IVAL.LT.0) IVAL=0
CALL DBLK(IVAL, IX, IY, IUX, IDY)
IY=IY+IDY
70 CONTINUE
IY1=IY
IY=IY0
IX=IX+I0X
II=II+1
IF(II.GT.1024) II=1024
80 CONTINUE
IY=IY1
IX=IX0
90 CONTINUE
CALL FCL05(1)
CALL DIR("DP0", IERR)
GOTO 10
END

TYPE DBLK
.TITLE DBLK
.ENT DCLR
.ENT DBLK
.EXIT .FRET, CPYL
. NREL
.DUSR XY0=23
.DUSR IVAL=-167
.DUSR IX=-166
.DUSR IY=-165
.DUSR IDX=-164
.DUSR IDY=-163
S

DBLK: JSR @ CPYL
LDA 0, 0IDX, 3
COM 0, 0
STA 0, DX
STA M, DXS
LDA 0, 0IDY, 3
COM 0, 0
STA 0, DY
STA 0, DYS
LDA 0, 0IX, 3
LDA 1, OIY, 3
LDA 2, @IVAL, 3
DCC 2, XYO
LDA 3, C512
ADD 3, 1
ADD 3, 2
ADD 3, 0

YRST: LDA 3, DYS
STA 3, DY
SUB 3, 3
ADD 1, 3

XRST: LDA 2, DXS
STA 2, DX
SUB 2, 2
ADD 0, 2

SCAN: DDA 2, XYO
DOB 3, XYO
JMP +1

MOS XYO
INC 2, 2
ISZ DX
JMP SCAN
INC 3, 3
ISZ DY
JMP XRST
DOAC 0, XYO
JSR @ FRET

CINT: 400
DINC: -4
DTIME: 0
DY: 0
DX: 0
DXS: 0
DYS: 0

OCLR: JSR @ CPYL
LDA 1, DINC
SUB 0, 0
STA 0, DTIME
LDA 0, CINT
DOCP 0, XYO
ISZ DTIME
JMP -1
INC 1, 1
MOV 1, 1, SZR
JMP -4
JSR @ FRET

C512: 177000
 END