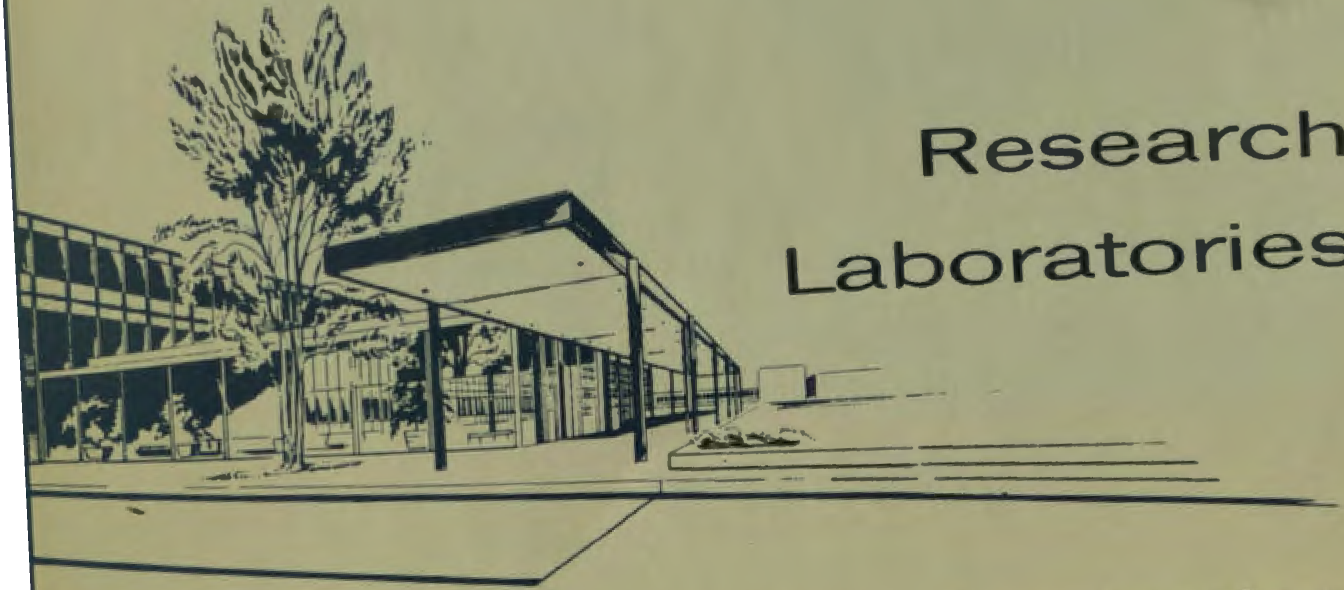
GMR-430

THE GM DAC-I SYSTEM DESIGN AUGMENTED BY COMPUTERS

●

Five papers presented at
The 1964 Fall Joint Computer Conference

Computer Technology Department
October 28, 1964



Research
Laboratories

General Motors Corporation

THE GM DAC-I SYSTEM

DESIGN AUGMENTED BY COMPUTERS

In mechanical design, graphics, e.g., drawings, are necessary for the development and communication of design ideas throughout the conception and implementation of any new device. In order to learn how computers can be used to assist man in his design work, in 1963 GMR placed in operation a special laboratory for the study of graphical man-computer communication.

This laboratory consists of a large digital computer, devices for direct graphic input to and output from the computer, a graphic display console for man-computer communication and control, a data file for storing the current state of the design, and an extensive library of computer programs.

The hardware and software, combined to form a system for Design Augmented by Computers has become known as DAC-I. A first report on the DAC-I system is contained in the following reprints of papers presented at the 1964 Fall Joint Computer Conference.

10/28/64

CONTENTS

A Laboratory for the Study of Graphical Man-Machine Communication	EDWIN L. JACKS	343
Operational Software in a Disk Oriented System	M. PHYLLIS COLE PHILIP H. DORN RICHARD LEWIS	351
Image Processing Hardware for A Man-Machine Graphical Communication	BARRETT HARGREAVES JOHN D. JOYCE GEORGE L. COLE ERNEST D. FOSS RICHARD G. GRAY ELMER M. SHARP ROBERT J. SIPPEL THOMAS M. SPELLMAN ROBERT A. THORPE	363
Input/Output Software Capability for a Man-Machine Communication and Image Processing System	T. R. ALLEN J. E. FOOTE	387
A Line Scanning System Controlled from an On-Line Console	F. N. KRULL J. E. FOOTE	397

THAT IS ALL.

A LABORATORY FOR THE STUDY OF GRAPHICAL MAN-MACHINE COMMUNICATION

Edwin L. Jacks
Research Laboratories, General Motors Corporation, Warren, Michigan

INTRODUCTION

Engineering has evolved rapidly during the last fifteen years as analysis techniques geared to the computational power of a slide rule and desk calculator have been replaced by techniques which make extensive use of computers. During these years, however, graphical techniques for conversion of design ideas to final products have not changed significantly, nor has the role of drawings in engineering design changed. The drawing plays a vital role in each phase of the evolution of a product. The original design proposals, the engineering analysis, the design compromises, and the prototype product fabrication all depend on graphical communication among engineers and designers. Whether the product is to be machined, assembled, stamped, wired, welded or hand modeled, a drawing is made so that a two-dimensional representation of the product may be reviewed by the engineers concerned with the product. Prior to the final product drawing, many ideas are exchanged by the use of sketches, drawings, plots, and engineering reports.

The drawing board is used as the basic mechanism for resolving problems in design packaging. For instance, "Where can part B be located if part A is made larger?" and "Can part A be assembled to part C?" As the design evolves, many decisions are made by the engineer while the drawing representing the design is being produced by the draftsman. The questions are endless, and, in many design problems, are not finally resolved until after early prototypes of a product are made.

Two key points in this process are: (1) the engineer is an integral part of the graphic design process, and (2) the draftsman is doing a task that requires considerable attention to detail and mechanical precision. In many mechanical design situations the two functions of engineering and preliminary product drafting are done by the same man. A drawing serves as his way of exploring design ideas.

The dependence of engineering design on graphical techniques is fundamental to the design process. Graphics serves as a language of communication among design personnel and, as outlined above, as a mechanism for design evolution.

The General Motors Research Laboratories had been using digital computers for engineering and scientific analysis for several years dating back to a card-programmed calculator in 1952, but notably absent from the applications were problems relating to graphical design. In the late 1950's, the Research Laboratories addressed the question, "Could computer techniques significantly improve the design process?". To answer this question, a study was started on the potential role of computers in the graphical phases of design. Prototype hardware and software components were developed to investigate the problems of image processing. A 740 cathode ray tube recorder attached to the 704 computer was already being used to plot results of engineering computations. It satisfied the requirement for graphical output. The associated 780 display unit provided a graphical on-line display which, along with a simple switchboard, became an elementary man-machine console. A program-controlled film scanner was devised using the 740 recorder; a photocell detector was substituted for the film magazine, and its output signal was connected to a computer sense switch. With this breadboard setup, lines on film could be digitized under program control. Programs were written for graphic input and output and for the manipulation of images in three dimensions. These early software and hardware components were integrated into an operating system that demonstrated the feasibility of using the computer as an aid in the graphic design process.

On the basis of this early feasibility demonstration, the decision was made to establish a more comprehensive laboratory for graphical man-machine communication experiments. The facilities were to permit the computational power of a large-scale digital computer to be brought to bear on the problems of graphical design in a manner which fully recognized the importance of the man in design. This project on Design Augmented by Computers has become known as DAC–I.

The initial goal of the Design Augmented by Computers project was the development of a combination of computer hardware and software which (a) would permit "conversational" man-machine graphical communication and (b) would provide a maximum programming flexibility and ease of use for experimentation. This goal was achieved in early 1963. This paper gives a broad outline of the computer technology which was developed to meet the above goal. Other papers [1,2,3,4] present approaches to solutions and examples of performance of the various hardware and software components of the system.

The present hardware complex consists of an IBM 7094 computer and an IBM 7960 Special Image Processing System. The Image Processing System was designed and built by IBM to specifications provided by the General Motors Research Laboratories (GMR).[2]

The supporting software was developed by the GM Research Laboratories Computer Technology Department and includes a multiprogramming system, an algebraic compiler (NOMAD), a data channel command compiler (MAYBE), a dynamic storage assignment procedure, and extensive facilities for the storage, retrieval and editing of programs and data stored on a disk storage device.[1]

Each major portion of the DAC–I system—the 7960 Special Image Processing System, the computer with its attached disk memory, the multiprogramming trap control system, the DAC–I monitor system, the programming languages used for system development and the disk filing programs—have, via their design criteria, all contributed to the system's flexibility and ease of use for experimentation.

DESIGN OBJECTIVES OF THE IMAGE PROCESSING HARDWARE

The over-all objective of the image processing system was to achieve the equivalent of what is possible with graphical man-to-man communication while utilizing drawings. In establishing systems specifications, four types of man-machine communication were sought.

The *first* type of drawing communication desired was static. The machine should be able to produce a hard copy drawing for engineering use. Conversely, it should be able to accept a drawing and be able, under computer control, to read the drawing. Because of the nature of automobile design, it was necessary that the DAC–I system be able to accept free form curves, i.e., curves which are constructed without consideration of particular mathematical representations. Furthermore, to provide compatibility with existing design procedures, precision input and output of such curves was needed. These requirements ruled against a "sketchpad" type of operation.[5] The drawing input-output functions have been achieved in the image processor of the 7960 System by: (a) using a high resolution cathode ray tube (CRT) under computer control to record drawings onto 35 millimeter film and (b) using a second and similar CRT as a computer-controlled flying spot scanner to scan 35mm film images of drawings. The image processor has built into it the ability to photo-seconds have the film ready for optical scanning. The output drawings are also ready for viewing 30 seconds after film exposure.

The *second* type of drawing communication desired was dynamic. The system should simulate the type of man-to-man communication where one man is drawing or pointing at a particular part of a drawing while another man is observing or discussing details of the design with the first man. This capability was provided in the graphic console of the 7960 through the combination of a 17-inch display tube and a device called a position-indicating pencil. When a designer touches the pencil to the glass plate in front of the display tube, the computer program can detect what position on the tube face is being pointed to and, hence, can react to any comments the man may wish to make about the indicated portion of the display. Thus, after the computer generates a picture on the display tube, either the man (by pointing at the display) or the computer (by placing an "x" on the displayed picture) can in effect say to the other, "Consider this portion of the picture."

The *third* drawing communication objective was simulation of the comparison function. The system should allow the overlay of two pictures to permit comparisons of the differences and similarities in the information.

This feature was provided by having the image processor designed such that pictures can be recorded on two separate film trains and then projected automatically onto a common view screen. This feature allows, for example, overlay of scanned data with the original film source for verification. By programming techniques, the graphic console can also be readily used to compare drawing information.

The *fourth* design objective was to achieve man-machine communication of non-graphic information. The system should provide, via the graphic console, a convenient means of communicating (a) alphabetic and numeric information to the computer, (b) multiple choice decision responses to the computer, and (c) permissible actions by the man. For alphanumeric information, a 36-position keyboard with upper and lower case and a slow-speed card reader is part of the graphic console. For communication of gross actions, the console has 36 program control keys and 36 message lights; the computer receives a signal when a program control key is depressed by the man, and inversely the man receives a visual signal when the computer turns on a message light.

A detailed description of the 7960 Special Image Processing System is contained in the paper by B. Hargreaves, J. D. Joyce and G. L. Cole, et al.[2]

OBJECTIVES OF THE COMPUTER HARDWARE COMPLEX

Studies at the GM Research Laboratories in 1959 and 1960 were made to estimate the computing facilities required to adequately support the DAC–I project. Considered in the studies were the number of instructions required to support the experiments, execution time for the required programs, and man-machine response rate. These studies indicated that approximately 200,000 to 500,000 instructions would be programmed for the graphic communication experiments. The computation required for these experiments was estimated in terms of central processing unit use per hour and amounted to 6 minutes of 7090 time for each hour of console use.

The response rate considerations were stated in terms of system objectives. We wanted the designer to be essentially working on-line and in "real time." The measure of real time was that the man and machine could carry on a meaningful conversation about a design at a rate satisfactory to the man. The response consideration then required a real-time approach to receiving and handling data arriving from the man. But the computer programs and hardware did not need to have a fail-safe time limit approach to sending a response to the man. For this reason the system is best described as online console system rather than a real-time system.

Another more independent consideration was the computing requirements of the GM Research Laboratories. In 1959 and 1960, a 704 was in use between two and three shifts per day, and it was forecast that a 7090 or equivalent computer would be reqired by 1961 to satisfy the continuing needs for an engineering graph 22 x 22 inch drawings and within 30 and scientific computing facility at GM Research.

The combination of the above requirements, we believed, could be met by a 7090 computer *. The speed of the 7090 would adequately handle the computational load and, if properly multiprogrammed, the machine would effectively be able to give the response time desired for console communication and computational purposes without wasting the estimated 54 minutes per hour of non-console use. The requirement for multiprogramming implied that the computer would need to be modified such that two independent programs could reside in its core memory with a minimum risk of either program modifying the other program. For this purpose, a core memory protection system was designed which prevents instructions from storing into program-specified 4K blocks of the memory.

Multiprogramming also implied that for accounting purposes a clock be attached to the computer so that proper timekeeping could be performed during the switching from program to program. A clock was built by the Delco Radio Division of General Motors for GMR with a millisecond as its basic interval of time.

The requirements for 0.5 x 10^6 words of program storage could be satisfied by having a disk memory on the computer. The original 7090 configuration had a 1405 disk connected via a 1401 and a direct data connection to the computer. The current facility uses a 1301 disk and three drum storage units for the program and data library. The computer complex resulting from the above set of specifications is shown in Figure 1.

OBJECTIVES OF PROGRAMMING SYSTEMS SUPPORT

The combination of the IBM 7090 and the IBM 7960 system as described above was to provide an experimental graphical communication hardware facility. To support this system from the software standpoint, it was decided that an investment would be made in programming techniques which would minimize the time from the conception of a man-machine communication experiment until the required programs were operating.

Figure 1 shows the computer as a central processing unit with five attached data channels. That is how the machine appears to the hardware man. To the people responsible for programming graphical communication experiments, however, the machine was to have an entirely different appearance. These people were to be in a programming position in which a large library of procedures were at the "finger tips" of their programs.

The programs were to be able to conveniently display situations to the man. If the man was expected to require more than a millisecond to respond, the programs were to be able to say to a control program, "Control, I am in standby status now and when the man answers my question or takes other action, return control to me."

For programming convenience, the programmer should be able to do all his programming in a higher level language (higher than an assembly type language at least) including the programming of the data channel driving the 7960 System, the loading of programs by name from the disk, and the analysis of all data coming from the image processing or graphic console equipment. In short, he should be able to program all of his graphical communication experiments in a language similar to FORTRAN or ALGOL. An algebraic complier (NOMAD) and a data channel compiler (MAYBE) were developed for this purpose.

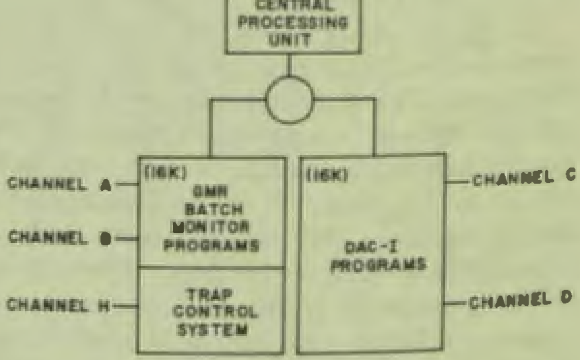


Figure 1. Computer Configuration.

---

* In 1963, the originally installed 7090 was upgraded to a 7094.

The specifications of the programming system revolved around three broad statements of facility operational policy. First, for programming purposes, the 32K computer memory was to be considered as two 16K blocks of memory *. This is represented symbolically in Figure 2, with 16K assigned to the DAC–I console support programs and 16K assigned to the standard batch monitor operation.

Second, all input/output programs in both the DAC–I operation and the batch monitor operation must use the trapping hardware built onto the 7090 and all trap program operations must be compatible with a trap control system (TCS) developed at GM Research.

The third policy statement indicated that the batch monitor's use of the computer was limited to channels A and B while the DAC–I console program's use was limited to channels C and D. This condition was imposed to prevent conflicts in hardware use and means, for instance, that tapes in use by the batch monitor could not be used by DAC–I during multiprogramming. One major exception to this rule was that the use of the disk was permitted by programs being executed under batch monitor control for purposes of compiling or checking out programs being developed as part of the DAC–I project.

TRAP CONTROL SYSTEM SPECIFICATIONS

Based on the above operation policies and on the programming system's objectives, specifications for the trap control system's performance in multiprogramming and the programming techniques for DAC–I were developed.

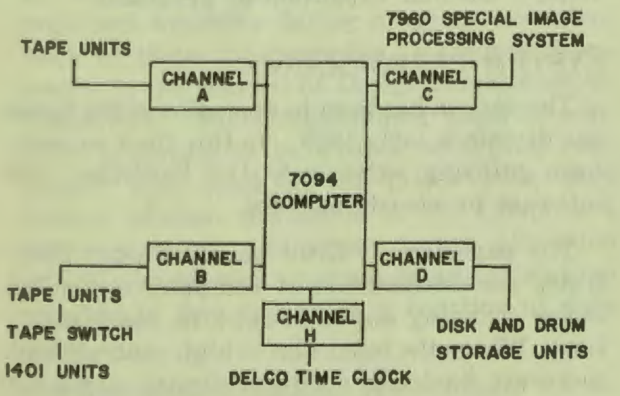


Figure 2. Split-Memory Operation.

The trap control system (TCS) was to meet the following broad specifications: (Refer to Fig 2)

1. DAC–I channel traps terminate monitor job central processing unit (CPU) operation. Machine status is saved by TCS prior to transfer of control to the appropriate DAC–I program.
2. If DAC–I has CPU control, monitor job channel traps are saved for later action. Monitor job traps are processed immediately upon return of CPU control from DAC–I.
3. TCS must switch memory protection regions as CPU control is switched.
4. TCS must honor DAC–I requests for additional memory space by a dump and memory protect release of all remaining core (except TCS). Restore and restart procedures for monitor jobs must be provided.
5. DAC–I channel traps should not be inhibited unless absolutely necessary.

MONITOR SYSTEM'S OBJECTIVES

With the above type of trap control system, the batch monitor and DAC–I program sequencing monitor each had distinct operating objectives.

1. The batch monitor should be a general purpose batch processing monitor and should be able to execute any program as long as the program was compatible with the trap control requirements and the 16K core limitations.
2. The DAC–I monitor was to accept from the graphic console card reader a single card containing job accounting information and a program name. The program name was to be the name of any subroutine stored on disk. The monitor loads the subroutine and turns control over to it.
3. Any program being executed on the DAC–I side should be able to enter a job into the batch monitor job stack. This permits conventional printed and punched

---

* As of March, 1964, the 7094 was expanded to two 32K memories.

output from a DAC–I program without using channels A and B at the time the output is generated. The disk is used as the temporary storage space for the output while waiting for a between job break in the monitor operation.

4. The DAC–I programming system should be duplicated within the scope of the batch monitor. This permits debugging of programs independent of the 7960 System and prior to execution in the DAC–I environment. When operating in batch monitor mode a basic monitor facility must be provided for the compiling of DAC–I programs in the NOMAD language and the subsequent filing of the resulting object program in the disk file.

THE DAC–I MONITOR

The DAC–I operating monitor was developed around the idea of a disk program library where all system functions and program execution sequences are built up from a) the basic operations of storage, retrieval, and updating of a library, and b) the allocation by the currently running program of memory core space for program and data.

Basic concepts of the system are:

a) The basic unit of a program is a subroutine which has a name, an entry point name, and a disk file area name.

b) Data for a subroutine may be either global or transmitted to it by a standard subroutine calling sequence. A global variable is declared at subroutine compile time, but no memory locations are assigned until the subroutine is loaded into core.

c) Program execution involves the loading of subroutines from disk as they are needed. It is the function of the loaded subroutine to assign subsequent locations within memory for whatever additional subroutines and global variable assignments are required for the program's task.

Based on the above concepts, the DAC–I monitor was required to provide:

a) A table which contains the location and size of each subroutine in memory.

b) A table which contains the location and size of all global variables in memory.

c) The basic codes required to retrieve programs and data stored in the auxiliary disk and drum memories.

d) A relocation program which, when given data in the form of a subroutine in memory, will relocate the subroutine and assign memory addresses to its global variables based on the subroutine and global variable tables mentioned in a) and b).

The basic facility ground rule was that given essentially the above codes any program written in NOMAD could then at execution time do its own storage allocation. The paper by M. Phyllis Cole, Philip H. Dorn, and C. Richard Lewis[1] describes the procedure for storage allocation (itself written in NOMAD). The method is basically program subroutine selection at execution time and it allows a programmer to make decisions in his programs as to which is the best method for handling the storage assignment for a given data set at execution time. He may either keep a large block of data in memory and pass programs by the data, or keep all his programs in memory and pass the data by his programs. In practice, for small data sets, the programmer keeps all of his program in memory. As the data set becomes larger, initialization, computation, and post-processing subroutines are cycled by the data.

With the combined facility of disk program retrieval by subroutine name and memory storage allocation at execution time, a very useful feature develops—any alphabetic data can be viewed as a subroutine name. This permits convenient modular expansion of programs.

SYSTEM PERFORMANCE

The system has been in operation eight hours per day since early 1963. In this time we have been utilizing extensively the hardware and software previously outlined.

The paper by F. Krull and J. Foote[4] illustrates the combination of computer-controlled image scanning and man-machine communication. Where the input film is high contrast and there are basically no uncertainties, a simple computer program rapidly solves the problem of conversion from graphics to binary data.

When uncertainties, such as arise when scanning low contrast film, become the dominant problem, then the man, as referee, can obtain control. One can argue that for each uncertainty a program can be written to analyze the situation and then the man is not needed to aid the process. The strong point of man-machine communication via graphic consoles is that for any given problem, one may now ask which parts of the problem are easily solved by the computer and which parts are best solved heuristically by man. This results in programs being written which have decision points in them at which the man at the console can be asked for advice. Many of the past discussions of man-machine communication have been based on the concept of "let the man get to the computer" so he can directly ask questions of the computer program. Experience at GMR to date has been that the payoff from consoles comes not from asking the computer a question but assigning the computer a task from which the response is one of the following: "What is my next job?" "Here is the answer; what next?" or, "I don't understand; and here is my analysis of the situation."

From the standpoint of a laboratory facility, the system is performing excellently. We are learning that man and machine can communicate readily via graphical means.

SUMMARY

The software development for the graphical man-machine communication laboratory has incorporated three major departures from conventional higher level language programming: 1) multiprogramming, 2) source program storage allocation control, and 3) a disk library of programs available during program execution. Each of these programming techniques is essential to the concept of Design Augmented by Computers. Multiprogramming permits computer programs to be written such that, even though they work at the man's pace, they achieve efficient utilization of the computer's processing unit. Program storage allocation control allows each program to adjust storage assignment dynamically as a function of data needs.

A disk library available at execution time allows a control subroutine to view other subroutines as black boxes which required certain inputs and produce outputs. The size and name of the black box does not need to be known at programming time and, in fact, are data at execution time, for the control subroutine. This feature allows continued growth of the design support programs with no change to control programs.

The above three software techniques combined with the flexibility of NOMAD, permitted a fourth major departure from conventional programming techniques. Ninety percent of the DAC–I programming system was written in NOMAD. The trap control system and the basic subroutine relocation programs were the major exceptions to the above. With the new laboratory facilities at GM Research, the process of man-machine communication for design can now be explored with both formal experiments (direct comparisons of methods with planned testing) and informal experiments (let's try something to see how it works).

ACKNOWLEDGEMENTS

The DAC–I software system is the product of many people at the GM Research Laboratories. The four papers (1, 2, 3, 4) associated with this paper reflects the contributions of their authors. The work on the programming system was directed by Charles S. Gerrish, and the batch monitor system was developed under the guidance of George F. Ryckman. The trap control system was developed by Floyd Livermore, and the linkage system to the DAC–I operation was done by Theodore J. Theodoroff.

Throughout this project Professor Bernard A. Galler of the University of Michigan has served as a consultant to the Research Laboratories on the development of the DAC–I software support program.

REFERENCES

1. Cole, M. P., Dorn, P. H., Lewis, C. R., "Operational Software in a Disc Oriented System," *Proceedings of the Fall Joint Computer Conference,* San Francisco, California, October 27–29, 1964, (this issue).
2. Hargreaves, B., Joyce, J. D., Cole, G. L., et al., "Image Processing Hardware for a Man-Machine Graphical Communication

System", *Proceedings of the Fall Joint Computer Conference,* San Francisco, California, October 27–29, 1964, (this issue).

3. ALLEN, T. R., and FOOTE, J. E., "Input/Output Software Capability for a Man-Machine Communication and Image Processing System," *Proceedings of the Fall Joint Computer Conference,* San Francisco, California, October 27–29, 1964, (this issue).
4. KRULL, F. N., and FOOTE, J. E., "A Line Scanning System Controlled from an On-Line Console," *Proceedings of the Fall Joint Computer Conference,* San Francisco, California, October 27–29, 1964, (this issue).
5. SUTHERLAND, I. E., "Sketchpad, A Man-Machine Communication System," *Proceedings of the Spring Joint Computer Conference,* Detroit, Michigan, May 21–23, 1963.
6. PREISS, R. J., "Design Automation Survey—A Report to the AIEE Membership," AIEE Winter General Meeting, New York, New York, January 29–February 3, 1961.

OPERATIONAL SOFTWARE IN A DISK ORIENTED SYSTEM

M. Phyllis Cole, Philip H. Dorn and C. Richard Lewis
Research Laboratories, General Motors Corporation, Warren, Michigan

1.0 INTRODUCTION

This is one of a series of papers which describes the General Motors Research Laboratories DAC–I (Design Augmented by Computers) System.(1, 2, 4, 5) For a summary of the overall system objectives and organization, the reader is referred in particular to the paper "A Laboratory for the Study of Graphical Man-Machine Communication" by Edwin L. Jacks.[1]

In using the DAC–I system, the man at the console wants to perform the following types of tasks in solving his problems—

1. Introduce data rapidly and accurately to the computer.
2. Operate on this data.
3. Observe the results of these operations and have the ability to modify them while still in the on-line environment.
4. File the original data and final results for future references.

To accomplish these tasks requires the interplay of the 7960 Special Image Processing System,* the 7094 Computer Complex, and the man.

It is the purpose of this paper to discuss the systems software developed in support of this interplay.

Requirements of this software system include:

1. Establishing efficient storage and retrieval methods for handling large numbers of data arrays and subroutines.
2. Creating an environment within the computer which would allow subroutines and associated data to be brought into memory, processed, overlayed, filed, etc. based on the operational demands made by the man at the console.

A disk oriented software system (hereafter referred to as the D-System) has been implemented in order to provide the basic software support. The objectives of the D–System are to provide compiler level accessibility to the new hardware devices and minimize the impact of disk usage on the general programmer. The D–System provides all of the necessary subroutines required in using the disk for data storage and retrieval, for scratch space during intermediate computation, and for the loading of subroutines based upon program needs. The D–System is accessed through a batch monitor system in order to generate subroutines and data for storage on disk. Program execution may be accomplished through the batch monitor or the on-line console.

Establishing an operational computer environment also required the development of compilers to satisfy needs for both the object and system codes which were to be written. The NOMAD compiler was used extensively

---

* The IBM 7960 Special Image Processing System, consisting of a graphic console, an image processor, and a modified data channel, was designed and built by IBM to specifications provided by GM[2].

for central processing unit (CPU) codes and the MAYBE compiler was developed for 7960 data channel codes. Processors for disk bookkeeping and a complete system for loading and overlaying subroutines in memory were also required.

This paper considers first the organization and maintenance of a disk file and then describes an operating system based on a disk with both system and execution time features detailed. A brief discussion of the on-line D–System operation is included. Finally, certain conclusions are drawn based on experience with a disk-oriented system.

2.0 THE ORGANIZATION AND MAINTENANCE OF A DISK FILE

2.1 *Introduction*

During 1961, a small disk file was attached to the computer permitting experiments in disk file usage. Although the file was both small and slow, its capacities were sufficient to allow meaningful simulation of the ultimate system. When the final D–System (a disk oriented software system) was being designed, certain design principles learned during simulation were applied.

It was ascertained that when planning storage allocation for a random access device, the following design criteria should be followed:

1. An absolute integrity must be maintained between the data to be stored and the programs which deal with the data.
2. The programming system shall not impose restrictions on the quantities of data retrieved from or stored on disk.
3. The system must be free of fixed locations on the disk file. The optimum is to provide one fixed track location and reference all other locations from this one.
4. The design of object time programs should not be concerned with the manner in which data is physically stored on disk.
5. The number and time duration of disk arm movements to reach a particular piece of information must be kept to a minimum even at the cost of inconveniences to the operating system.
6. Efficient disk back-up procedures are necessary.

In planning the storage allocation, four different types of storage areas were noted. These were: first, processor areas where subfunctions of the operating system reside; second, subroutine areas where object time subroutines are permanently stored; third, data areas; and fourth, scratch and temporary working areas. Each of these areas is handled in a unique manner and the techniques therein involved will be discussed below.

2.2 *Storage of System Processors*

A processor is a logical portion of the operating system which may be called upon to perform a unique function. Examples of processors are the compilers, assembler and a processor which files subroutines on disk.

Processors* are stored in absolute form. Filing is handled by a separate non-system program which blocks relocatable subroutines into an absolute package. During the filing process, a dictionary is developed which contains the following information for each processor:

1. processor name
2. disk location
3. length of processor
4. processor entry point
5. processor loading location

Since both processor storage and retrieval is by absolute blocks, it may be seen that speed in loading is a feature of the processors. The system (using a cylinder reading technique) loads each processor as an absolute block into the location specified in the dictionary. Control is transferred to the entry point as specified by the processor dictionary. Since no system processor exceeds the capacity of a disk cylinder, no provision has been made for handling overflow to adjacent cylinders.

2.3 *Storage of Subroutines*

The disk area reserved for subroutine storage is divided into logical areas with each

---

* All processors are stored on disk except for the basic compiler and assembler. Although there is no logical reason why they could not have been placed on the disk, as a convenience factor they were left on the System Master Tape.

uniquely named area assigned on a project basis. Each logical area consists of a block of physically contiguous tracks bounded for minimum access time.

A separate dictionary is maintained for each disk area. Each dictionary entry contains the following information for each subroutine within that area:

1. subroutine name
2. disk address of the subroutine
3. length of the subroutine
4. date the subroutine was filed on disk

Subroutines are stored on disk in a blocked, relocatable binary form. Output from compilers and the assembler is post-processed into blocked form and then filed in a two record per track format. Each record contains a pointer to the following record except the last record which has a null pointer. When subroutines are being retrieved, the pointer becomes a command to the loading routine to access the next record.

Allocation of a particular location to a subroutine is made from a subroutine space assignment table. This table contains the following information for each disk area:

1. area name
2. location of area dictionary
3. next available dictionary track
4. first and next available tracks in the area
5. number of available records in the area
6. next available record in the area.

In addition, the subroutine assignment table contains the location of a track "pool" to be used by an area whose basic space allocation is exhausted. Should this occur, subroutines enter the "pool". The area limits may be enlarged during edit time *. Subroutine areas and their respective indexes are located by referencing this table; should it become necessary to redefine any or all areas, only the assignment table need be changed. The system refers to track assignments through the table; references to absolute disk locations are not permitted.

---

* The edit function is explained in Section 2.6 entitled, "Disk File Maintenance Procedures".

2.4 *Permanent Data Storage*

All permanent data is stored in uniquely numbered files assigned on a project bais. Each file may contain four different data types where a data type is defined as one of the four different record sizes maintained on the disk. The types are:

type 1: 25 word record—stored 15 records per track

type 2: 111 word record—stored 4 records per track

type 3: 229 word record—stored 2 records per track

type 4: 465 word record—stored 1 record per track

To establish a data file, the user must indicate how many records of each type his file will contain *.

A directory table is maintained for each data file permanently stored on disk. The directory tables are stored on disk in numeric sequence beginning with file number 1. The table contains information as to whether this particular file number has been assigned, and if assigned, where the first track of each data type may be located. The location of the first file directory table is maintained in memory so the operating system may access the directories quickly. Given a file number and the number of directories per track, it is simple arithmetic to compute the disk address for a file directory which in turn points to the data file.

Disk space allocation for permanent data is made from a table stored on disk. This table has an entry for each group of uniquely formatted tracks.** A group of tracks is called an area and the entry for each area contains:

1. the first track in the area
2. the currently available track
3. the next available record on the currently used track

---

* Space allocations, if not sufficient, may be changed after the file has been assigned space. This process is not, however, performed automatically.

** The IBM 1301 requires a strict formatting of any given cylinder for the size of the records to be maintained on the tracks within that cylinder. The file can be reformatted under program control but this procedure is not generally available to D-System users.

4. number of available tracks in the area
5. number of available records in the area

System subroutines have been provided to store and retrieve permanent data. In general, data files are referred to at execution time as record N of type M.

2.5 *Scratch Data Storage*

A scratch file is available to programmers for temporary storage of data during execution. To use the scratch area, a request is made for n tracks of 465 words to a system subroutine. Assignment is on a "last-in, first-out" basis. If the space is available, the system returns a key word containing the first and last track of the area assigned. Any subsequent storage and retrieval is by record number with the key being used as a base address for computation of the physical track referenced.

When the programmer is finished, he executes a "closing" subroutine and the tracks which were made available to him will be returned for future assignment.

2.6 *Disk File Maintenance Procedures*

Because the disk file may have occasional mechanical, hydraulic or electronic failures, a back-up procedure was developed to insure the preservation of the permanently resident information. Since both data and subroutines change from day to day, a working procedure was developed which consists of writing a disk save tape(s) daily. These areas are the only disk areas saved. Disk back-up tape(s) are retained for several days before being released.

Both subroutine and data areas are periodically edited by separate programs. Editing is a clean-up procedure; information is physically moved to designated areas and space made available by deletions is returned for reassignment.

Subroutine editing is basically the process of constructing a new area dictionary by omitting entries for deleted subroutines. The subroutines are taken off disk and placed on a scratch tape for temporary holding. At this point in the edit process any change to the boundaries of an area may be made. The subroutines are then assigned new locations within the area, the area dictionary is updated and the subroutines and dictionary written back onto the disk. At completion of the edit, all subroutines within an area are sequentially assigned and remaining space is available for future changes.

The data editing process is basically similar to the subroutine edit although the data edit program has the additional attribute of being able to operate upon any one type of data (or all types if desired) in a given machine run.

Data and subroutine restore programs reverse the save procedure and reload the disk from the last save tape(s). This procedure normally is used only in the event of major machine failures. A "cold start" procedure in the event of complete catastrophe includes subroutine and data restoration as well as reformatting, rewriting of home addresses, reloading processors and rewriting certain system control information.

3.0 THE D–SYSTEM

3.1 *Introduction*

The D–System is a switchbox which examines the input stream of requests, loads the appropriate processor and links processors by passing parameters. A pointer is positioned by the D–System so that a processor may know the current position in the input stream. The D–System is also responsible for collecting timing information to be used for installation billing operations.

The basic functional breakdown of the D–System is outlined in Figure 1 *:

Each of these three areas—input, bookkeeping, and execution processing—is described in succeeding sections.

3.2 *Input Processing*

Input processing is that set of system functions which results in object programs being

---

* The execution of the D-System as well as other systems operated at GM Research is controlled by the General Monitor Program. The General Monitor decides which system to call in, processes accounting information, signs tapes on and off the machine, and provides a means for operator communication to the programming systems.

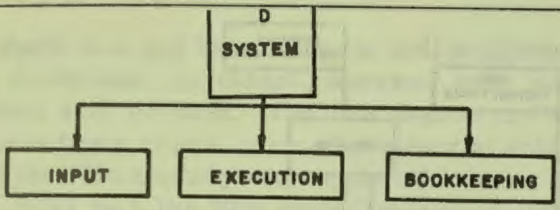


Figure 1. D-System Functions.

placed on the disk. Object programs are placed in one of two disk areas—either checkout or scratch, depending on the programmer's request. Subroutines filed in scratch disappear at the end of the job while those filed in checkout remain on disk for approximately one month.

The input processors are shown diagrammatically in Figure 2.

3.2.1 *The NOMAD Compiler*

NOMAD is an algebraic compiler adapted from the MAD * language to meet the special needs of this installation. It is a high speed compiler which permits a wide latitude of generality in expressions. Since MAD was implemented in a highly modular fashion, there was little difficulty redefining code generation sequences, adding new operators and enlarging the statement repetoire. Although MAD is of the ALGOL '58 family of languages, it has been modified considerably since its original design.

The NOMAD dialect developed at GM Research Laboratories has four basic areas of difference from MAD:

1. additional operators
2. new variable types
3. new relocation scheme
4. real-time statements

Thirteen new operators were added to the language to permit a full set of logical operations. Of special note are three bit detection operators that seek the first, last, and number of "1" bits in a variable, and a set of address/decrement packing and unpacking operators. These operators were especially designed to permit coding of system subroutines in the NOMAD language.

A new class of variables, the *GLOBAL VARIABLE*, was introduced. A global variable is defined as a variable, single or array, to which storage is not assigned at compile time. Unlike the COMMON variables in a FORTRAN program, global variables not used in a program do not have to be declared merely for the information of the loader. No special ordering of global variable declarations is necessary.

For each global variable used in a NOMAD program, the compiler generates an entry into a list attached to the program card information **. Each occurrence of a global variable results in special relocation bits being produced to indicate that one or more fields of this instruction are global. Additional bits indicate the slot in the list to which the global variable is assigned. If the global variable is subscripted, the numeric subscript is placed in the field normally assigned to the address.

The relocation scheme is based on a variable number of bits assigned to each instruction type (e.g., absolute, relocatable address with absolute decrement, etc.). The most frequently used class of instruction is described by one bit, the second most frequent by two bits, the next most frequent by three bits, etc. The first "0" bit reached acts as a delimiter. Comparisons made to other schemes have shown non-trivial operating efficiencies as well as considerable core and disk space savings.

The real time statements within the NOMAD compiler seek to acknowledge the presence of man in the program loop. Since a console for display of graphical information is part of the hardware configuration, system users output data onto the console display screen rather than the system output tape. Because the

---

* The *M*ichigan *A*lgorithmic *D*ecoder (MAD) is an algebraic compiler based upon ALGOL '58. It was originally developed by Arden, Galler, and Graham of the University of Michigan Computing Center for use on the IBM 704. The language and compiler have been updated through a series of revisions for the 709 and 7090. The NOMAD compiler springs from the earliest 7090 version circa 1961.

** A NOMAD subroutine contains on its program card(s) data relating to the global variable(s), the entry point(s), the program length and the number of program cards. When the subroutine is actually filed on disk, additional information is added during the blocking process such as the program's checksum and the length of the transfer vector.

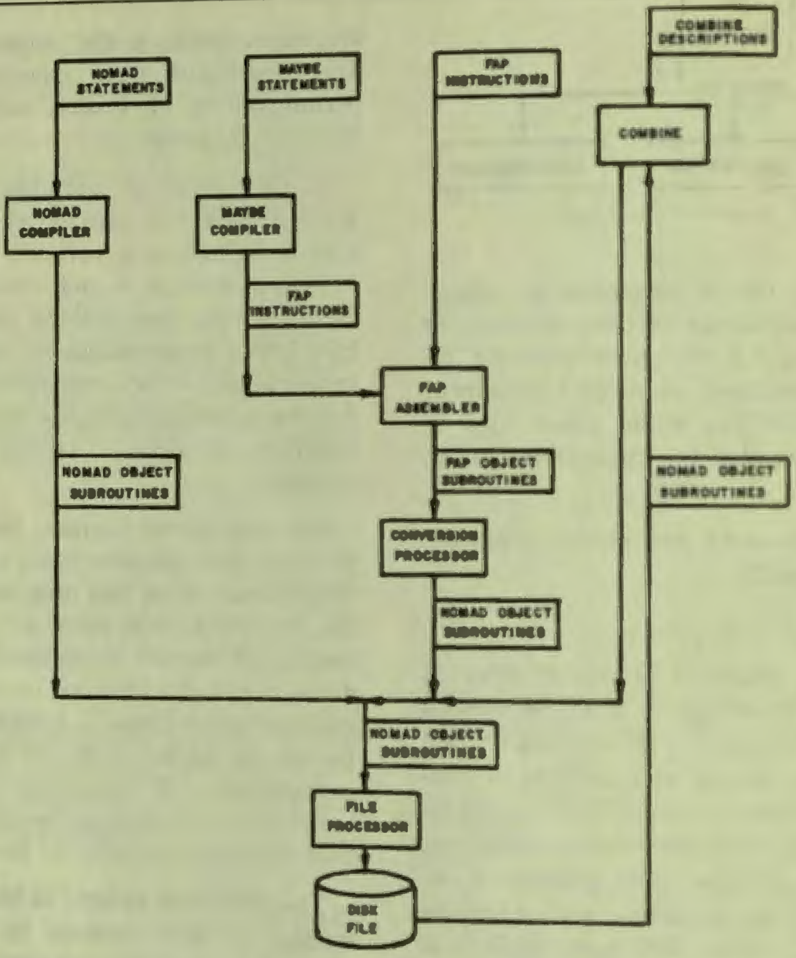


Figure 2. D-System Input Processors.

NOMAD statement for writing tape output has the form:

PRINT FORMAT F, List

where F is the name of a format and List is the data to be written, it is appropriate and logical to give the statement:

DISPLAY FORMAT F, List

for displaying information on a screen. A similar statement exists for production of hard copy output through the recording CRT:

RECORD FORMAT F, List

where F and List have the same meanings as above.

These statements represent steps toward development of a language which recognizes parallel processing in a large scale computer. Since the IBM 7094 has the capability to drive multiple channels in parallel, it is essential to permit the direct use of the full capabilities of the machine at the source language level.

Other features of NOMAD which contribute to its selection for use in the D-System are more conventional but still important. NOMAD permits a completely general subscription expression, a generalized iteration statement, multiple entries and exits from subroutines, the manipulation of statement labels, the use of internal procedures, nested conditional statements and the use of certain elementary push down list facilities.

3.2.2 *The MAYBE Compiler*

The I/O devices of the 7960 Image Processing System are connected to the central computer through a modified 7909 data channel. The data channel is capable of performing simple iteration loops, full and partial word substitution and byte testing as well as driving I/O devices. It cannot, however, add, subtract, shift or mask. The data channel can be viewed as a special purpose processor attached to the 7094 memory and capable of running in parallel with the 7094 central processing unit. It was necessary that means be provided to program this special purpose computer in a higher level language.

The MAYBE compiler was designed and implemented to provide the instructions, commands and orders for operation of the data channel and 7960 I/O devices. In addition, MAYBE automatically produces the necessary system linkages to process the data channel interrupts and central computer traps. Figure 3 shows the relationships among the computer, the data channel and the 7960 system.

Each subroutine generated by MAYBE consists of prologue instructions and a main body of data channel commands and orders. Stripped of its frills, the MAYBE compiler is a macrogenerator which feeds symbolic input to the standard assembly program. MAYBE was coded in NOMAD and utilizes standard system I/O routines.

The MAYBE language includes approximately 75 declarations and statements divided into the following classes:

1. storage allocation declarations
2. replacement statements
3. iteration statements
4. control and linking statements
5. device manipulation statements

MAYBE declarations are essentially the same ones found in the NOMAD compiler and provide a means to utilize local and global storage for data variables. The replacement statements allow substitution of data variables even when they are positioned in non-standard fields *. Iteration statements permit loops within the data channel and testing of the loop control index. Control statements permit the transfer of control within MAYBE subroutines or externally to other MAYBE or NOMAD subroutines. Device manipulation statements permit the starting, continuance and stopping of I/O devices attached to the 7960 system.

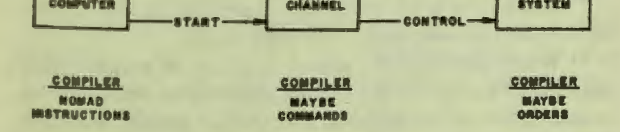


Figure 3. 7909 Relationships.

Linkages generated by the MAYBE compiler permit MAYBE subroutines to interrupt operations and transfer control to NOMAD (main frame) subroutines. In this way, a nesting of alternate MAYBE and NOMAD subroutines is achieved. The maximum depth of this nesting operation is the programmer's ability to remember where he is; there is no system specified limit. At each step, the data channel and the main frame will be jointly interrupted and their respective status saved. Thus, no matter how deep the nesting, the machine status will be restored upon return to each higher level.

MAYBE is essentially a compiler for use of system programmers. Most users operate devices through NOMAD statements (such as DISPLAY or RECORD FORMAT). Subroutines coded in MAYBE provide the I/O commands to drive the requested device.

3.2.3 *The Combine Processor*

The COMBINE processor permits the D–System user to reduce a set of NOMAD object level subroutines into one physical subroutine. This installation's programming standards emphasize subroutinizing as a checkout technique. Advantages may be gained by using a combined package of subroutines since the number of disk accesses at load time is sharply reduced as are the bookkeeping tasks during subroutine execution.

In addition to producing one relocatable subroutine from many, the COMBINE processor has the following features:

1. Global variables used for communication between the set of subroutines to be com-

---

* Instruction fields for 7960 commands and orders vary from the address and decrement fields normal to 7094 instructions. Provisions had to be made to handle fields as small as three bits.

bined may be drawn inside the package and assigned local storage or left as external global variables at the user's option.

2. A merged external transfer vector is produced for the combined package. Internal connections between the set of subroutines involved are handled through an internal transfer vector. References between one subroutine of the set and another within the set do not produce an external transfer vector entry.
3. Any entry point or set of entry points associated with the subroutines of the set may be retained, discarded or renamed at the user's option. A discarded entry point has no external meaning and does not exist outside the scope of the set.

Since all D–System processors have themselves been through COMBINE, there remains only the single merged transfer vector to be set during processor loading.

3.2.4 *The Assembly Program*

A standard 7094 assembly program, FAP, has been built into the D–System. Although it is available for general use, programmers are discouraged from using it except for highly special cases where extreme speed or efficiency is required, or for special purpose utility routines (such as number conversion and input/output). The bulk of the D–System and the processors, as well as over 95% of the applications programs, are coded in NOMAD.

3.3 *The Bookkeeping Processors*

The bookkeeping processors exist for the purpose of allowing the programmers to add, replace, and delete subroutines within the disk area to which their project has been assigned. Since the system permits retention of many subroutines with the same name but allows only one version to be in any area at any time, users' are responsible for having the right version in the right place at the right time.

The following bookkeeping processors are available in the D–System:

1. Move
2. Delete
3. Move and Delete
4. List Areas

Subroutines are initially filed in either checkout or scratch. Programmers generally leave new subroutines in checkout until debugging is completed. A facility exists for overriding normal calls for this particular subroutine * so that the new version may be debugged as either an isolated unit or within the total operating environment. Subroutines may remain in checkout for one month after they are compiled. After that date, if not moved from checkout, they will be discarded at edit time. MOVE is the programmer's means of sending a subroutine to a permanent area.

DELETE allows the programmer to take a version of a subroutine off the disk when it has outlived its usefulness.

MOVE AND DELETE performs the dual function of moving a subroutine to a new area and deleting it from the old area.

All these processing functions do not physically move the subroutine involved but merely make entries (or delete entries) in the subroutine dictionary for the area under consideration. Actual moving is done at edit time. Should a programmer move a subroutine to an area which already has a same named subroutine on file, the new version will automatically override the old version.

LIST AREAS permits a programmer to obtain a full listing of all subroutines in a specified area(s) along with their lengths and filing date. The physical disk address is also printed out at this time **.

3.4 *The Execution Phase*

D–System exception philosophy diverges from execution logic in most systems. Taking a standard FORTRAN batch processing moni-

---

* This is the USE processor described in the section on subroutine execution. (Section 3.4.2)

** While printing the actual location of a subroutine may seem to violate the system criteria of not allowing a user to know the physical location of anything on the disk, it actually is virtually useless information since the physical layout of an area will change at each edit pass. Those responsible for the correct operation of the hardware need this information occasionally after a machine failure.

tor for comparison, the following major differences may be pointed out:

FORTRAN SYSTEM	*D–SYSTEM*
Subroutine loaded from tape	Subroutines loaded from disk
Core loads	One subroutine loaded at a time
Ping-pong and overlay	Continuously changing core configuration
COMMON variables loader assigned	Global variables not assigned until needed

3.4.1 *Dynamic Loading*

While grouping subroutines in a core load is reasonably efficient when using magnetic tape as an input medium, this procedure becomes wasteful when a random access on-line device is available. Subroutines can be loaded individually from disk *as they are needed* without the burden of tape spacing and rewind time. Variables placed in COMMON in a FORTRAN system are assigned locations at compile time. D–System global variables are assigned locations at execute time when the actual core availability determines the location assigned. This floating quality of a global variable is essential to maintaining a flexible core arrangement.

Since multiple versions of any subroutine may exist, the D–System loader must receive a specification of the area in which the subroutine is located. When a global variable is first referenced by a subroutine, the dimension of the variable is needed to assign space. A D–System programmer defines the scope of his program by supplying area information for his subroutines and dimensions for his global variables. The D–System performs the function of an interface between the programmer and the loader to initiate execution. The system accepts the name and area of one subroutine and the dimensions of a set of global variables and passes this information to the loader as "starter" parameters. Execution commences with one "starter" subroutine which may be located anywhere on disk. The disk areas of other subroutines and the dimensions of other global variables are dynamically passed to the loader as execution proceeds.

When a program is in execution, the status of individual subroutines may vary as indicated in Figure 4. The status of a given subroutine is one of the following:

1. Undefined—This status is included for completeness and indicates the basic state of all subroutines on the disk.
2. Not In—The subroutine has been located on disk and may be loaded as needed.
3. Active—Whenever a subroutine is executed for the first time, it is loaded from disk and becomes "Active".
4. Inactive—A subroutine declared "Inactive" by the programmer remains in core in anticipation of later use. However, if additional core storage is required, it is returned to "Not In" status making the core location which it occupied available.

When subroutines have served their purpose, they are declared "Out" and return to undefined status.

The status of global variables may vary as shown in Figure 5. A global variable is assigned storage when the first subroutine which references it is loaded from disk. The storage is released when all subroutines referencing the global variables are declared "Out".

Availability of core space is maintained by the loader as subroutines and global variables change status. Therefore, the functions of the loader may be summarized as follows:

1. Change status of subroutines as directed by the executing program.

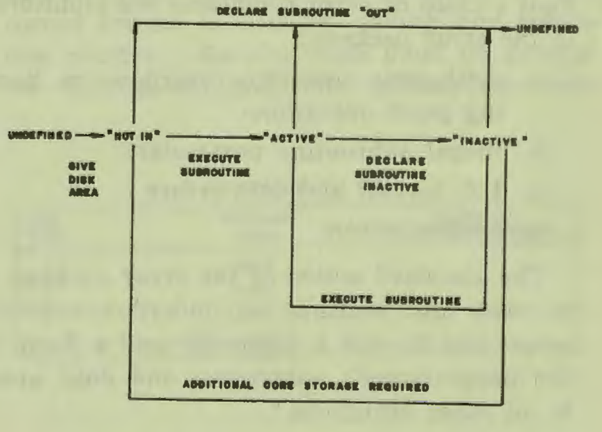


Figure 4. Subroutine Status.

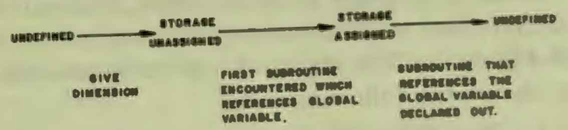


Figure 5. Global Variable Status.

2. Change status of global variables as directed by the executing program.
3. Maintain storage availability.

3.4.2 *Dynamic Subroutine Definition*

A D–System feature allows the checkout of a new version of a subroutine without disturbing the production execution of a previous version of that subroutine. This is done by placing the newer version in the scratch or checkout area. The production programs anticipate finding the subroutine in a permanent storage area. However, the loader may override their instructions and find the subroutine from scratch or checkout. This is accomplished by means of the USE processor which transmits a user's request to the loader. All requests to load this subroutine will be trapped and the version from scratch or checkout loaded instead of the production version.

3.4.3 *Error Procedures*

The D–System error package operates in one of two modes. The programmer may elect to monitor all software error conditions and take corrective action, or he may allow the error package to perform standard procedures. Four classes of error conditions are monitored in the error package:

1. arithmetic underflow/overflow in floating point operations
2. illegal subroutine parameters
3. I/O format and data errors
4. loading errors

The standard action of the error package is to reset and continue on underflow/overflow errors and to emit a diagnostic and a dump of the programmer's subroutine and data areas in all other situations *.

When the programmer is monitoring the error conditions, he may reenter the error package to obtain the following actions:

1. reset and continue on underflow/overflow
2. reset and continue on loading errors caused by insufficient core storage availability. Prior to re-entering the error package, the programmer must make additional core storage available by releasing space taken by either global variables or subroutines. If the same loading error occurs a second time, the execution will be terminated by a dump.
3. job termination with a memory dump of either the programmer's core area or all of core.

Since core is constantly changing during a D–System run, a full or selective core dump is always accompanied by a core map. The map contains the following information:

1. the name, location, length and status of all defined global variables.
2. the name, location, entry points, length and status of all defined subroutines.
3. the location and length of remaining available core space.

3.4.4 *Data Handling*

Data that is to be stored permanently is always in array form where word one of the array is the data name and word two contains the length of the array.

The data name is encoded to contain a file number, a key to which record within that file is being referenced and a record revision number. Upon the first reference to a file, the directory table ** is brought into memory and is used with the data name to compute the track and record address for the data request.

Should a data array exceed the record length of the indicated data type, an additional record is assigned and chained to the base record indicated by the data name.

At the beginning of each record is a control word, CW, containing three flags defined as follows:

flag 1 = 1 if revision has been filed, = 0 otherwise

---

* The system is not normally dumped.

** The directory table is described in section 2.4 dealing with data storage assignment.

flag 2 = 1 if data has been filed, = 0 otherwise

flag 3 = 1 if data has been deleted, = 0 otherwise

An example will best illustrate the control and chaining techniques used. Assume a data array, TORQ, 129 words long. Assume also the record revision number is 0. The first two words of TORQ are as follows:

Assume the directory table for this file to be:

The encoded name in TORQ(0) references the fifth record of data type 2 in file #19. Data type 2 (stored 111 words per record, 4 records per track) begins on track 1280 for this file. Therefore, TORQ(0) points to record 1 of track 1281. Since the length of the array is greater than 111 words, the first record will be chained, as shown on the next page, to an additional record taken from a pool of available records.

If it is desired to store a revised version of the array TORQ at the same time retaining the original data, a slightly different procedure is followed. First, the data stored on track 1281, record 1, is moved and the original record of 106 words on track 1281 is chained to the new location. The remaining 23 words on pool track 1682, record 2, do not move. The revised data is stored in a newly assigned record and another chain in record 1, track 1281 is set to point to the location of the revised data.

The name given in TORQ(0) identifies revision number 1 of the data. As before, record 1 of track 1281 is to be referenced.

Under this addressing scheme, the programmer need only give the name of a data array to retrieve it—the length of the array is stored in the array itself and will govern the number of words transmitted from the disk.

4.0 ON-LINE EXECUTION

In addition to execution within the batch monitor system, a D–system program can be executed on-line from the graphic console. In this operation the batch monitor is restricted to half core, and the other half of core is made available to on-line operation. A simplified system exists for on-line execution to act as an interface between the man at the console and a slightly modified D–system loader. The man at the console indicates to the system the name and disk area of the subroutine which initiates execution.

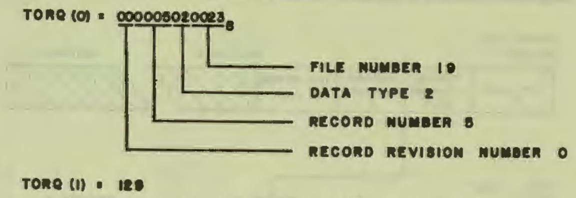


Figure 6. Data Name Encoding.

While there is no difference in basic philosophy between batch monitor and on-line execution, implementation is quite different since tapes are not used directly when operating on-line. Graphical output is used whenever possible to replace normal output tape functions. When printed output is required, as in producing core dumps, the information is placed on the disk and inserted in the output stream of the batch monitor system between jobs. The on-line program has the ability to insert jobs into the batch monitor system via the disk. A circular file is used to pass data to these jobs. The program in execution must also use the disk rather than tape for scratch space.

5.0 CONCLUSIONS

Our experience indicates it is feasible to operate from a disk and gain rapid access to large amounts of information, thus attaining considerable on-line capability. To obtain this on-line capability, users must pay a penalty in several areas. Core memory space must be reserved for an in-memory loading and relocation routine. Machine time must be granted for disk bookkeeping and editing functions.

DISK TABLE FOR FILE — 19

DATA TYPE	BEGINNING TRACK	NUMBER OF RECORDS RESERVED
1	0080	25
2	1280	10
3	2880	12
4	6800	8

Figure 7. File Directory Table.

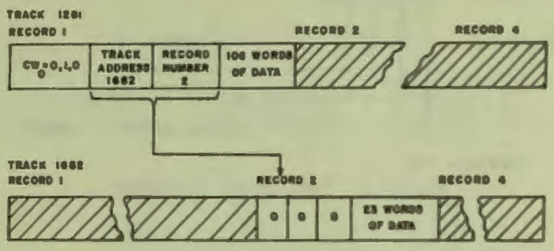


Figure 8. Data Track Layout.

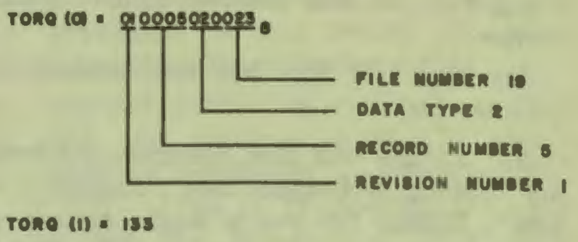


Figure 9. Encoded Revision Name.

Compatibility with other installations is completely lost.

In return for this investment, the system allows access to an enormous library of routines without having to deal with an object level deck. Large quantities of data are stored on-line and may be added to, modified, deleted or used with no difficulty. Both subroutines and data are always available; run preparation time is sharply reduced.

REFERENCES

1. JACKS, E. L., "A Laboratory for the Study of Graphical Man-Machine Communication," *Proceedings Fall Joint Computer Conference,* San Francisco, California, Oct., 1964, (this volume).
2. HARGREAVES, B., JOYCE, J. D., COLE, G. L., FOSS, E. D., GRAY, R. G., SHARP, E. M., SIPPEL, R. J., SPELLMAN, T. M., THORPE, R. A., "Image Processing Hardware for a Man-Machine Graphic Communication System," *Proceedings Fall Joint Computer Conference,* San Francisco, California, Oct., 1964, (this volume).
3. IBM 7090/7094 Programming Systems FORTRAN II Assembly Program (FAP) IBM Form C28-6235-2.
4. KRULL, F. N., FOOTE, J. E., "A Line Scanning System Controlled From an On-Line Console," *Proceedings Fall Joint Computer Conference,* San Francisco, California, Oct. 1964, (this volume).
5. ALLEN, T. R., FOOTE, J. E., "Input/Output Software Capability For a Man-Machine Communication and Image Processing System," *Proceedings Fall Joint Computer Conference,* San Francisco, California, Oct. 1964, (this volume).

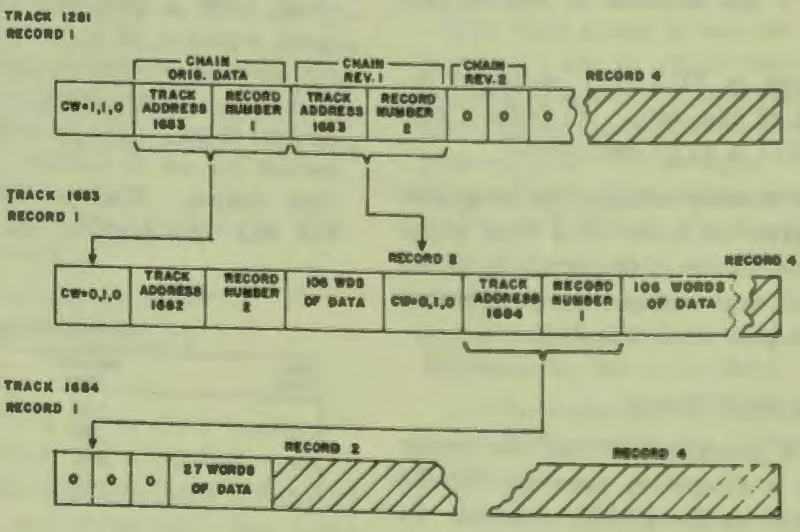


Figure 10. Revision Track Layout.

IMAGE PROCESSING HARDWARE FOR A MAN-MACHINE GRAPHICAL COMMUNICATION SYSTEM

Barrett Hargreaves, John D. Joyce and George L. Cole
Research Laboratories, General Motors Corporation,
Warren, Michigan
and
Ernest D. Foss, Richard G. Gray, Elmer M. Sharp, Robert J. Sippel, Thomas M. Spellman and Robert A. Thorpe
Data Systems Division, International Business Machines Corporation,
Kingston, New York

INTRODUCTION

The General Motors Research Laboratories (GMR) obtained the IBM 7960 Special Image Processing System in order to provide a laboratory for the study of graphic data processing and related man-machine communication problems. The IBM 7960, designed and built by the IBM Data Systems Division to specifications provided by GMR, consists of:

a) A graphic console which includes a display tube, control buttons and lights, a card reader, an alphanumeric keyboard and a position indicating pencil.

b) An image processor which permits computer-controlled scanning of film images and computer-controlled recording on 35mm film.

This paper is divided into two parts. Part I, written by the IBM authors, describes the design of the special image processing components and the integration of these components into the system. The main functional requirements for these components are computer compatible image generation speeds and high image quality. The design shows how the diverse technologies of analog circuits, cathode ray tubes, optics and film processing were successfully combined to provide a new type of image processing system.

Part II, written by the GM authors, is a review of GM's experience with the hardware as a component in the General Motors Research Laboratories' DAC–I (Design Augmented by Computers) System. Other papers in this series (1, 2, 3, 4) cover various aspects of this system.

The hardware is a working model of auxiliary computer equipment for designers. Previously, experiments have been conducted on individual components of equipment such as man-machine consoles or light pens, or image digitizers or image recorders or plotters. These experiments have pointed out possibilities for future developments in computer-aided design equipment. Now, all the necessary hardware components have been developed and put to use as a complete DAC–I hardware system.

In addition to having demonstrated the capabilities of this equipment for man-machine relationships in design experiments during the past one and one-half years, GM Research has created and used extensively new types of programs that both improve the effectiveness of the hardware by calibration and evaluate and display the status of the hardware for the user or maintenance engineer. One test program is described briefly as an example.

PART I—ENGINEERING DESIGN

FUNCTIONAL DESCRIPTION

A block diagram of the 7960 Special Image Processing System is shown in Figure 1–1. The attachment of this system to the central processing complex is through data-channel logic. In relation to the central processing system, the Special Image Processing System appears almost identical with any of the other data channels which may be attached to the system. In fact, the special data channel is an IBM 7909 Data Channel, modified slightly to make it better suited to the particular tempo of data flow that exists with this system.

The 7960 system comprises three basic units. The display adapter unit performs such functions as control of the basic system, control unit selection, and digital-to-analog conversion.

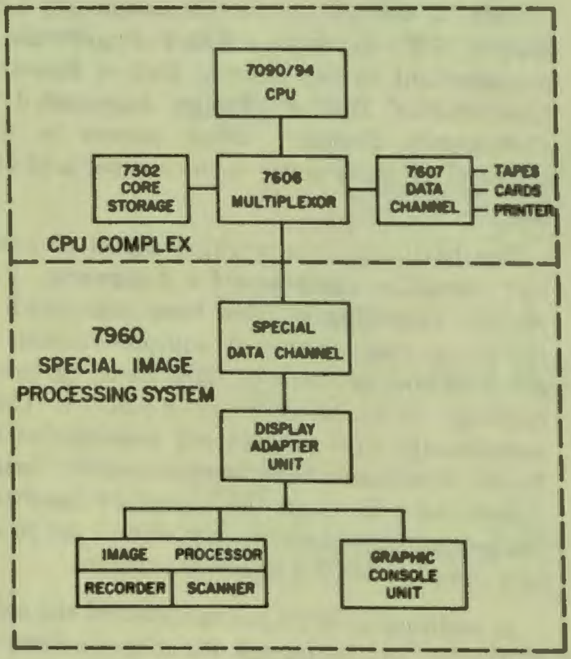


Figure 1-1. System Block Diagram.

The image processor provides for the input and output of data in graphic form. The unit contains a CRT photo recorder, projectors, CRT photo scanner, an input camera (for photographing drawings and documents), and rapid film-processing equipment. (Figures 1–2 and 1–3).

The graphic console is the primary system control point (Figure 1–4). It contains a CRT display, graphic pencil input, alphanumeric input from keys and punched cards, special function inputs from keys, and status and program status indicators. Information may be entered or modified in the system through the use of the graphic pencil, the program control keys, and alpha-numeric keys, or the card input. The results of calculations are displayed on the cathode ray tube or indicated on the status lights. Detailed descriptions of each of these units follow.

Display Adapter Unit

The display adapter unit controls the transmission of data, unit control information, and unit status information, and the sequencing and synchronizing of the various units in the system. In addition, digital data received from

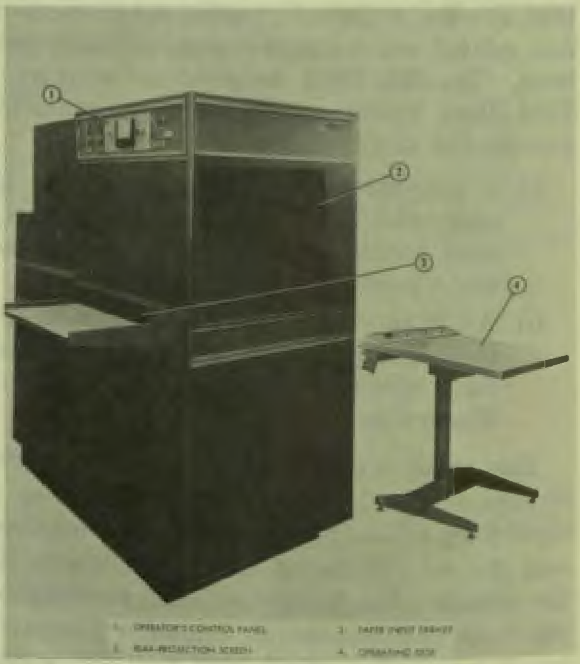


Figure 1-2. Image Processor Unit.

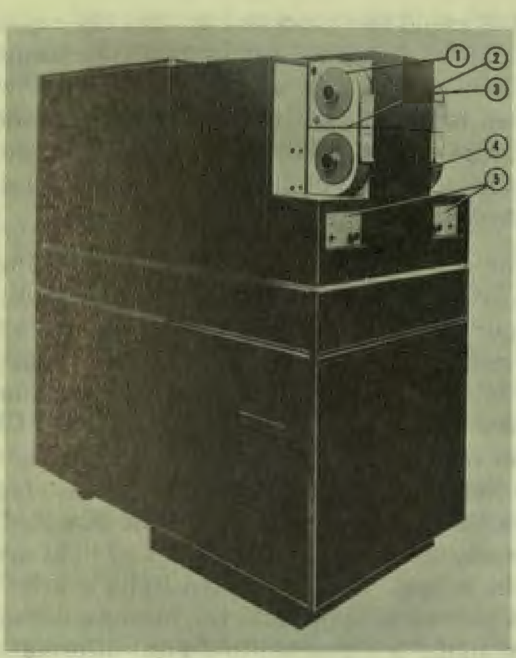


Figure 1-3. Image Processor Unit.

computer storage is formatted for deflection commands for the CRT devices.

The display adapter unit and the data channel recognize five states that may exist. Four standard commands (control, write, read, and sense), perform all data movements in the system. These commands serve to transfer data in 6-bit bytes, which contain the encoded information associated with the operation to be performed. The fifth state, an interrupt signal, is used to notify the data channel when operator action requires a branch in the program.

Image Processor

The image processor provides the input and output of data in graphic form. The unit contains two photographic-film transport units which are similar in operation but which differ in the functions that they perform. For convenience, they are designated transport A and transport B. Figure 1–5 is a simplified representation of the film transport and optical system.

Transport A:

a. Exposes film from a high-resolution recording cathode-ray tube.
b. Exposes film from a paper input station.
c. Processes the film (develops, fixes, washes, and dries).
d. Scans processed film for computer input at the read station using a high-resolution scanning CRT.
e. Projects the processed film from the read station to a 20 x 22-inch rear-projection screen located at the front of the unit.

Transport B:

a. Exposes film from the record CRT.
b. Processes the exposed film.
c. Projects processed film.

Both transports can be operated independently and simultaneously, within the limits imposed by the optical and shared data paths for the CRT's. For example, exposing film from the record CRT involves a mirror which directs the image to the selected film transport; therefore, only one film may be exposed at a time.

Image Input

The source document for an image-processing design system is normally graphic information on paper. The paper documents will include engineering drawings, sketches, or graphs

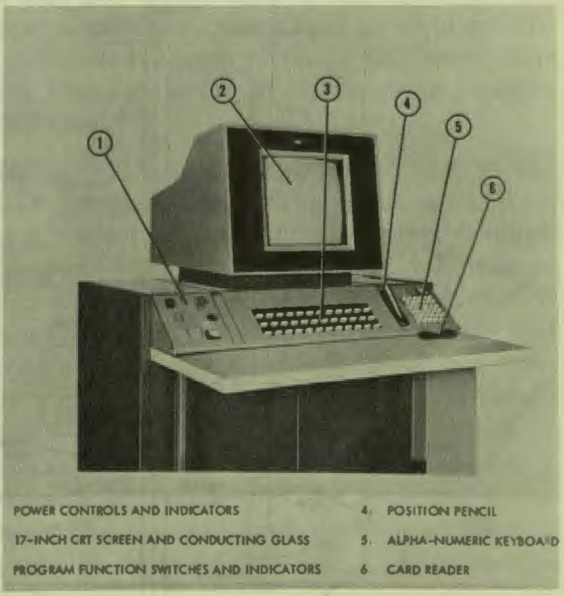


Figure 1-4. Graphic Console Unit.

which must be converted by a digitizing process into a form acceptable to the computer. In the 7960 System, paper documents are photographed on 35mm film and the film is used as the media for the image input process, as shown in Figure 1–5.

Film was chosen as the media to be scanned because of the need of being capable of processing a range of paper sizes and image qualities. The reduction of various document sizes to a standard image size for scanning permits higher scanning speeds and the ability to control image quality.

Paper documents up to 22 inches square are reduced to a 1.2-inch square on sprocketless 35mm film by the paper input camera. To expose the paper it is positioned in the paper-input drawer and held flat by means of a vacuum. An array of eight flood lamps illuminates the paper. The intensity of the illumination is under operator control and exposure can be varied to adjust for differences in image density and contrast on the paper. A paper-input shutter provides a timed exposure.

The use of 35mm silver film as the image input media requires an on-line-computer-controlled, rapid-film processor so that the exposed film can be developed for immediate scanning. A three-station cup application process was chosen to provide an image with uniform density, high-image stability and a resolution comparable to that obtained by hand-processing methods.

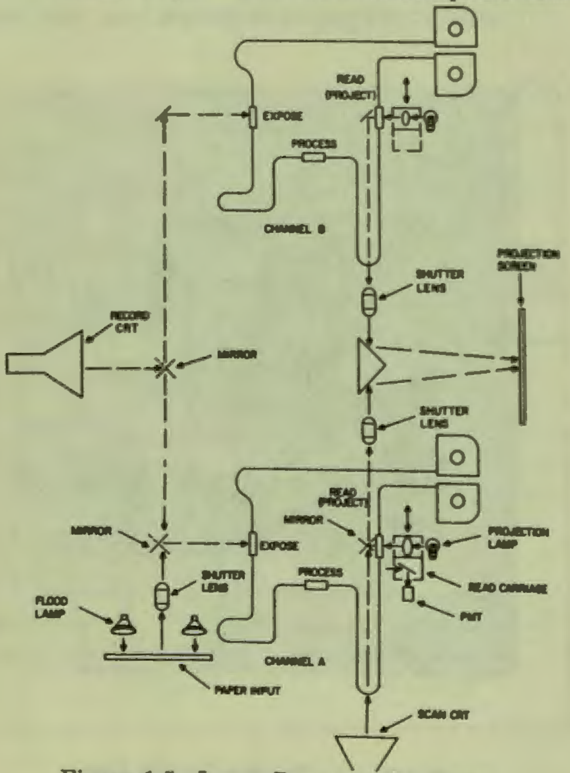


Figure 1-5. Image Processor Schematic.

The processed film image is digitized under computer control by a flying-spot, CRT scanner (Figure 1–5). A CRT scanner was chosen for the reduction of graphic data to digital computer data because of the CRT scanner's computer-compatible speeds and flexibility. The CRT beam can be scanned over the 1.2-inch square film image area under program control. Light from the CRT passes through the film and is intensity-modulated by the dark and light areas of the image. The modulated light is detected by a photo-multiplier and the amplitude-modulated signal is converted to digital information.

The primary considerations in the design of the scanner were high scanning speed and accuracy. Accuracy includes both the reliability of detected data and the high relative positional stability over the period that the image is being scanned. The CRT beam is moved from point to point over the image by vectors composed of straight-line segments of varying length. This method of beam positioning is called an end-point vector method because, regardless of the length or direction of a vector, only the new end-point must be specified. This results in minimum computer data to control the scanning vector.

The data required to draw one vector is given by the computer in 12 bits for X position and 12 bits for Y position. These digital values are converted to analog voltages which determine the deflection current applied to the deflection yoke of the CRT. The scanning beam can be positioned over 4096 x 4096 addressable positions. An effective increase in positional resolution of the scanner is obtained by the use of a constant-time-scan vector system. In the course of a vector, the light passing through the film intercepts a line or lines of the image contained in the frame. Each line interception is sensed by the PMT and is called a strike. With a constant time vector system, it is possible to divide a vector into time segments which can be related to position. A strike occurring during a particular time segment can be related to the position on the image corresponding to the position of the scanning vector at that time. Scanning response resolution to a fractional part of the vector length can be obtained by this method.

In the scanner, the time required to draw a vector is either 32 or 256 usec. The 32-usec-scan-vector time is used for short vectors; the 256-usec-time for long vectors. To accommodate a wide range of images with varying image density, contrast and line widths, it is necessary to provide program controlled scanner detection sensitivity. The detection threshold of the strikes or hits can be varied by commands so that optimum detection sensitivity can be selected by the program.

Image Output

After an image is processed by the computer program and modified and verified by the operator at the graphic console, it is frequently necessary to produce a permanent output document of the processed image. Film is used for the output image media, as for the input media, because of its compatibility with computer speeds, flexibility of use, and high image quality.

CRT recording on silver film permits images to be generated by computer control at computer-data-channel speeds. The CRT used for output image recording is similar to the type used for input image scanning. As in the scanner, a CRT image is formed by vectors drawn on the CRT screen. The vector trace is visible when the beam is unblanked, in order that the beam trace can be recorded on the film. When the beam is blanked, the trace is not visible and the beam can be positioned over the image without exposing the film. An image—for example, a drawing—is divided into straight line segments. Core storage contains the X and Y coordinates of the end point of each vector. Vectors may vary in length from the very short traces required to form a smooth arc to longer traces which are used for straight line segments of the drawing.

The deflection circuits used for CRT recording are the same as those used for CRT scanning and graphic console display. Therefore, two fixed vector times are available for image recording: 32-usec vector time, generally used for drawing short vectors, and 256 usec generally used for long vectors. Four vector line widths are selectable: basic, 2x basic, 4x basic and 6x basic. A constant vector time system for recording results in a beam velocity that varies with vector length. If not compensated, this would result in varying film exposure and therefore, varying density on the recorded image. To provide even exposure for all vector lengths and vector widths, dynamic intensity control is used. This analog circuit provides continuous compensation for beam speed and line width.

Other analog circuit corrections must be used to obtain a high quality, high resolution, linear image on film. Continuous beam focus compensation is required to maintain a perfectly focused beam over the CRT face. The flat screens of the record CRT and scan CRT require focus compensation that is a maximum when the beam is deflected to the edges of the screen, and follows a parabolic function which gradually diminishes to zero compensation when the beam is at the center of the screen.

The X and Y signals, which cause beam deflection at the record and scan CRT's must be corrected to prevent "pin cushion" effect. This effect is inherent in flat CRT screens and causes the sides of a square to become concave arcs and the overall area to be enlarged. An analog pin cushion corrector is used to modify the deflection current to make the beam position a linear function of the angular position of the beam.

The output image is exposed to film at the expose station of either Transport A or B of the CRT recorder. The film is pulled through the expose station in one-frame increments, and is exposed to an image from the record CRT as for the paper input. The exposed film is accumulated in a storage loop until a sufficient quantity of film is available for rapid processing. A loop of film is maintained in front of the expose station so that the frame can be quickly pulled down by the drive mechanism. The processing of the exposed film is identical to that which occurs after paper input exposure.

After the output image is processed, it may be immediately viewed by the operator by projecting the film image onto a 20-inch by 22-inch rear projection screen (See Figure 1–5). At the projection station the film may be advanced or backspaced one frame at a time under computer control or advanced or backspaced incrementally under operator control. By utilizing both of the film transports, each with a projection station capable of projecting a film image onto the common projection screen, it is possible to compare two images or to produce 3-dimensional effects on the screen.

The large screen projector permits the operator to study the output image off line from the computer. The image is larger and of higher quality than can be obtained on the graphic console and the image can be studied and compared with drawings or other graphic console images.

Graphic Console

The graphic console (See Figure 1–4) provides primary system control. The man-machine-communication components of the graphic console are:

1. A 10-inch-square CRT display surface, and a position–indicating pencil.
2. Thirty-six program-status lights (with a message overlay) and 36 program control keys.
3. An alphanumeric keyboard.
4. A card reader.

The 10-inch-square display surface is a CRT display with a phosphor coating designed to control flicker and improve viewing comfort. As with the scanner and recorder, the display is created by having the computer specify the end points of the vector to be drawn. The dynamic CRT display utilizes a transparent conductive screen with an impressed voltage gradient and a voltage-pickup position pencil to aid in operator modification of the displayed image. Basically, the data read under program control notifies the tracking routine whether the pencil position is to the left or right and above or below a particular (X,Y) position.

Thirty-six program control keys are provided for use in addition to the input pencil. The function of a particular program control key is assigned by the program and can be changed from program to program. Replaceable overlays (See Figure 1–6) are used to identify the function of each key for each application. The descriptive labels on the overlay can be illuminated by the program-status indicators which are also under program control. An alphanumeric keyboard, consisting of 36 keys arranged in a 6-by-6 key matrix pattern enables the operator to enter data at the console. When the keyboard is operated in conjunction with an upper and lower case switch, alphabetic, numeric and special character codes can be generated. There is, in addition, a manually-fed, card-reader input for the entry of limited amounts of data.

The code generated by either the keyboard or the card reader can be interpreted by the program as desired, giving additional flexibility to these devices.

ENGINEERING DESCRIPTION

Film Transport Control

A film transport consists of the following units (Figure 1–7):

1. A film supply cassette and takeup cassette.
2. Drive motors, clutches, film guides and other controls that move the film from the supply cassette.
3. An expose station.
4. A process station with its associated processor - applicator - elevating mechanism.
5. A read station for projection or scanning.

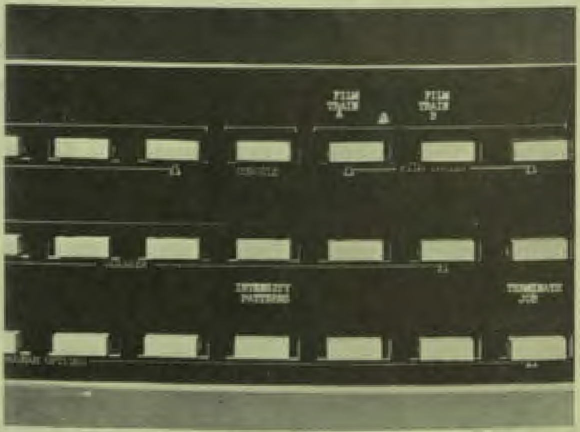


Figure 1-6. View of Graphic Console Overlay.

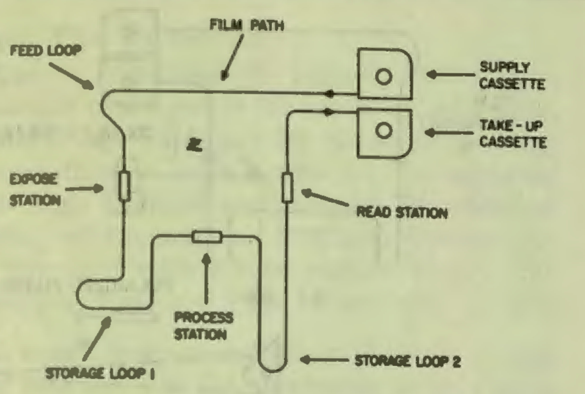


Figure 1-7. Film Transport.

Film is fed from a supply cassette capable of holding 400 feet of 35mm unsprocketed film. The film is moved through the channel to a takeup cassette by means of a friction drive.

Initially, the film motion controls thread the film through the film transport. Independent motions at each of the three stations (expose, process and read) are permitted within the limits imposed by film storage loops between stations.

A small loop of film (Loop Feed) is maintained prior to the expose station so that the film can be quickly pulled down by the drive mechanism without encountering excessive drag. The film is pulled through the expose station in one-frame increments at a rate of approximately one frame every 210 ms by the expose station drive motor. The exposed film is accumulated in Storage Loop 1 until processing is desired. The maximum capacity of Storage Loop 1 is 20 frames before processing must be performed. When the computer signals the channel to process the exposed film, the film is moved through the process station at a rate of 31 inches per minute by the processor drive motor.

Storage Loop 2 (between the process and read stations) accumulates the processed film until the computer advances the film to the read station. If Storage Loop 2 becomes full and processing is still going on, film is forced to advance through the read station. In this way, processing is not interrupted to prevent film from being ruined through over-development or under-development. Film can be backspaced at the read station into Storage Loop 2; its maximum capacity is 20 frames. At the read station, film may be advanced or backspaced one frame at a time under computer control at a rate of approximately one frame every 170 ms or advanced or backspaced under operator control at one of two speeds: ¼ inch or one inch per second as seen on the screen.

Photographic System

Optical Elements

There are five essentially independent optical systems in the image processor:

1. Input camera system
2. Scanner system
3. Recorder system
4. Projection system
5. Alignment system

The first four systems are used in the operation of the image processor while the fifth is a maintenance aid. All of the optical paths (except alignment) are shown schematically in Figure 1–5.

The input camera system consists of a 22-inch-square, paper-input drawer (See Figure 1–2) with a vacuum-actuated platen to keep documents flat; a series of tungsten, line-filament, light sources located above the drawer; a series of reflecting mirrors; a 4-inch, f/4.0 lens with an electrically-actuated shutter and an expose station on channel A.

In operation the drawer would be extended outside the image processor, a document placed in the drawer, and the hold-platen actuated. Upon depression of a control switch, the drawer with the document on it will automatically return to its normal position, the light sources will be turned on, the shutter will be opened, and an exposure made on 35mm film at a reduction of 18.3X. Both the intensity of the light sources and the shutter timing are variable to provide for flexibility of exposure. The resolution of the optical system at the film plane is approximately 150 lines/mm.

Film images at the channel A read station can be projected for visual examination or can be scanned electronically by means of the flying spot scanner. Optical switching is used to obtain the functional selection (See Figure 1–5). A 9-inch, f/4.5 CRT lens was designed to reduce the CRT presentation 2.5X to scan the 35mm film. The lens, corrected for the P16 phosphor emission band, yields a resolution at the image plane of 160 lines/mm with a distortion of less than 0.1%.

A collector lens is placed behind the film image which images the exit pupil of the scanner lens on the sensitive cathode of a photomultiplier tube (PMT) detector. This lens serves to uniformly distribute the light passing through the film over the PMT cathode.

Dual-channel recorder optics are provided to expose film on either channel A or channel B (See Figure 1–5) using a common CRT source with optical switching. Optical switching is also used to select the exposure source from either the paper input or the recording CRT on channel A. The recording operation utilizes a lens similar to that used in the scanner optics except that it is corrected for the P11 phosphor emission band.

The simultaneous projection of film images in the read stations of the two channels to a common viewing screen is provided (See Figure 1–8). The same lens as that used in the paper input camera was selected for projection. Off-axis projection, using the displaced image plane technique, gives maximum screen illumination with minimum distortion. Superposition of the two projected images is achieved by moving the film along the vertical axis of the screen and by racking the channel B projector lens along the horizontal axis.

Color filters are provided in the projector lamphouses to aid in differentiating the two images at the screen while projecting simultaneously into the projectors to provide stereoscopic or 3-D viewing. When these filters are introduced by electrical command into the projector light path, the projected images are selectively polarized. Complementary polarizing glasses must be worn by the viewer. The capability of advancing and backspacing the film in the read station is provided. A magnification of 18.3X is used in projecting the film on the 20 by 22 inch projection screen.

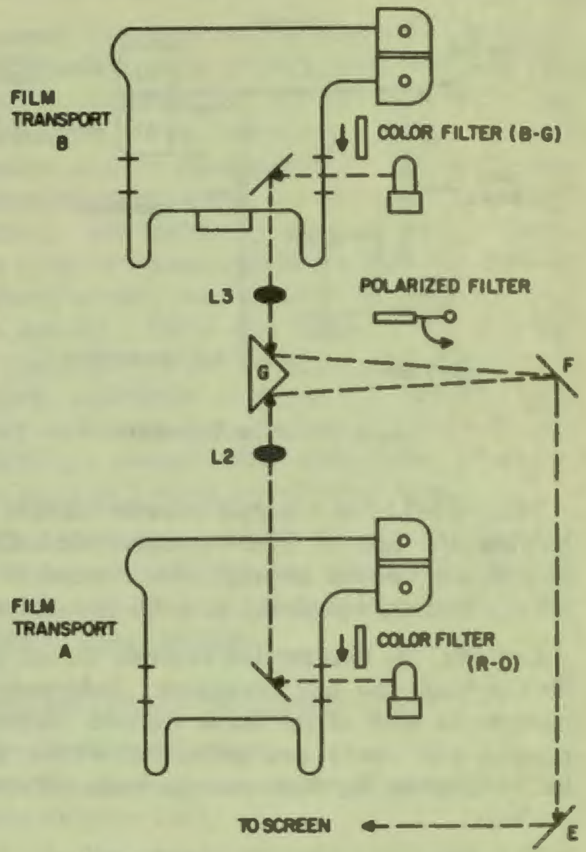


Figure 1-8. Simultaneous A-B Projection, Simplified Diagram.

The CRT's internal alignment optics serve as a reference to which the recorder and scanner cathode ray tubes are aligned. An illuminated reticle is brought into visual superposition with either CRT display. Switching from the recorder CRT display to the scanner CRT display is accomplished by manual rotation of a beam splitter. Variable magnification of the alignment system allows for both gross and detailed inspection of the CRT displays.

Recording Media

The selection of photosensitive material required the weighing of several desired characteristics. For document-recording, a film in the microfilm resolution class was desired. For CRT recording, a film with an extremely high sensitivity was desired. An additional design criterion was that of elevated temperature, short-time film processing. The silver halide film emulsion designed for this unit has a high blue sensitivity, a reasonably high resolution—130 lines/mm, medium contrast and capability of withstanding the rigors of a high-temperature process.

Rapid Film Processing

Upon a command, the exposed film is automatically processed in the rapid-film processor, which brings chemicals for processing at high temperatures in contact with the film emulsion. This high temperature increases the chemical activity of the solution such that the film can be processed with a total contact time in the developer, fixer and rinse of only five seconds.

A negative-pressure, cup-application method was selected for the processing (See Figure 1–9). The compliant rubber lips on the solution-applicator cavities form a seal with the film emulsion. Pumps situated in the heated-fluid container draw the solution from the container through the hose to the applicator cavities and back to the pump. The negative-pressure fluid system greatly reduces liquid spillage hazards since any leakage in the liquid circuit results in air being drawn into the circuit. This is particularly important when a process such as this one is integrated into a complex electronic device. The rubber lips on the applicator act as squeegees between the sequential processing steps to minimize contamination of solutions. The exit lip of the applicator removes the surface moisture from the film to minimize drying time. The film dryer directs high-velocity heated air against the film emulsion which is thus dried in approximately one second. Vent valves located in the return line to the pump are used when the processor is moved away from the film. When these valves are actuated, air enters the fluid line allowing for gravity drain of the applicator cavities. The applicator can then be lowered away from the film, without spilling any chemicals.

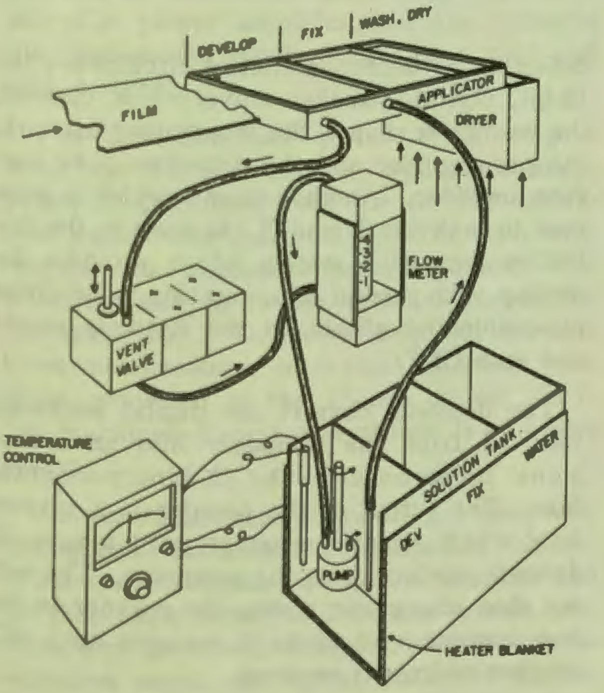


Figure 1-9. Film Processing System.

The processor processes film at a 31 inches/minute rate. The compartmented solution tank has a volume to accommodate the processing of 400 feet of film. The solutions in the tank are maintained at 130° F through the use of a blanket heater and a temperature controller. The one solution tank and two sets of solution pumps supply the solution to the two rapid film processors, one for each of the two channels.

Analog System

General Description

The analog system, which controls the scan, display, and record CRT's, scan detection, and position-pencil operation, is shown in Figure 1–10. As can be seen, a single set of analog circuits is used to control all three CRTs. Switching between tubes is done with relays and is under computer control.

The control circuits relating to the CRTs perform three basic functions. The deflection-control system precisely controls the position of the electron beam on the face of a given CRT as a result of a sequence of digital X, Y addresses supplied by the computer through the control unit. The focus control provides a uniform (in size), round CRT beam over the entire usable area of the flat-face record and scan CRTs. Without this control, the CRT beam would increase in size and become astigmatic (oval) as the beam position moved off-axis. A farther requirement of this control is to provide for four program-selectable line widths (CRT beam sizes) for film recording. The intensity control is required to maintain constant beam brightness in the system CRTs, independently of beam size (line width) and beam velocity.

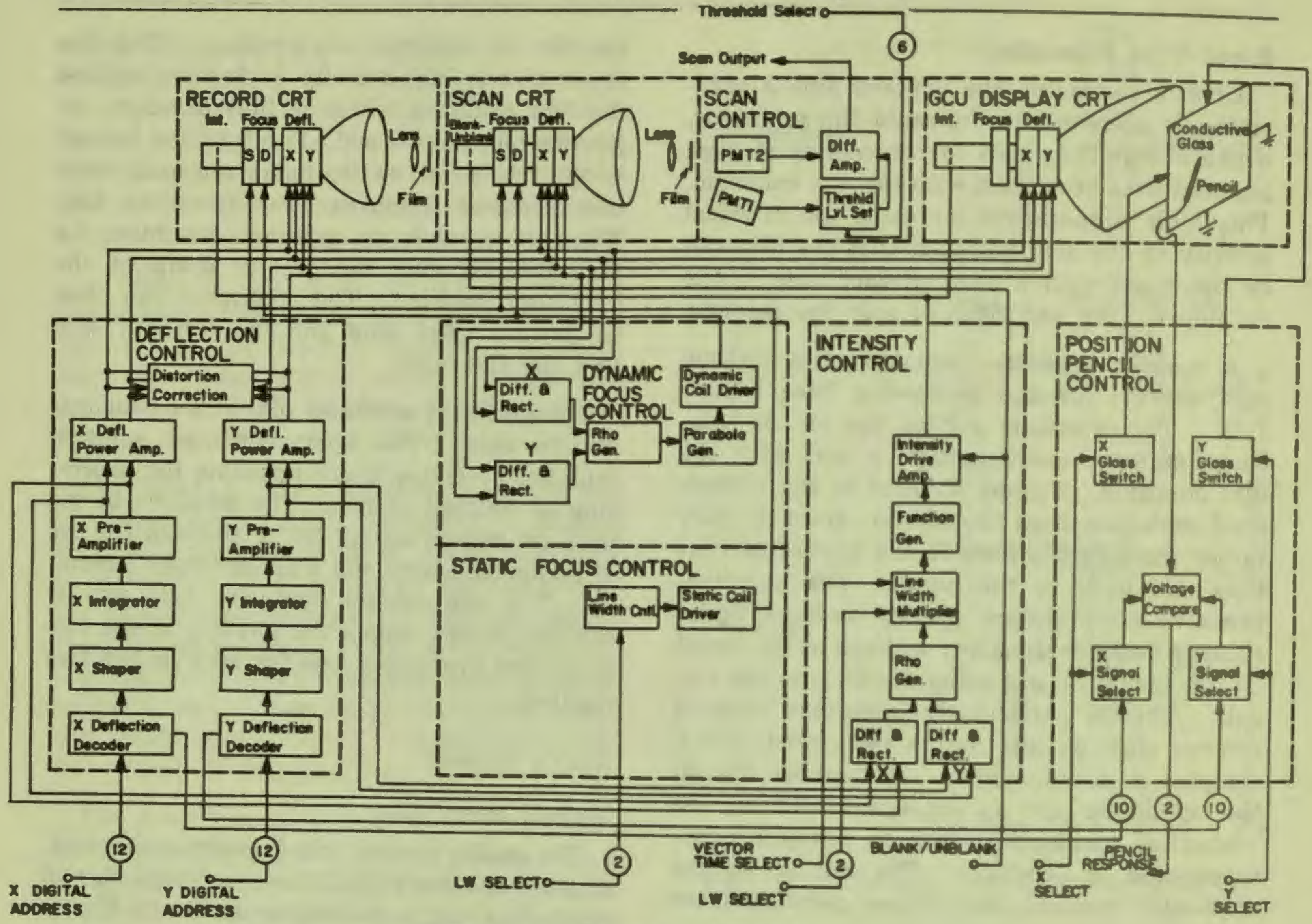


Figure 1-10. Analog System Block Diagram.

The position-pencil control system allows the pencil, when in contact with the conductive glass screen of the GCU, to be located by the computer such that the CRT beam (either blanked or unblanked) appears at the pencil location. The scan detection system senses the light output of the scanner CRT which is modulated by the film image being scanned and correlates the amount of light received at a particular time to a position of the film image. There are 64 program selectable threshold levels representing image transmissivities from 0 to 100%.

The following sections describe each of these major control systems.

Deflection Control

The deflection control circuits utilized in the 7960 System are shown in block diagram form in Figure 1–10. Note that the circuit configuration is identical for both X- and Y- deflection channels. The main elements of the X- deflection Y- deflection control circuits are the 12-bit, digital-to-analog converter or decoder, the waveform shaper, the integrating network, the pre-amplifier and the deflection yoke current amplifier. Another circuit which is common to both the X and Y channels is the distortion correction system which provides deflection yoke current compensation to minimize pin-cushioning effects on the flat-face record and scan CRTs.

The decoders convert the digital addresses received from the computer into an analog signal proportional to the 12 binary-weighted bits. The output of the decoder is a current level which remains constant until a new address is received from the computer. The output then changes in a step-like manner to the new current level where it remains until still another address is received.

The decoder output is then fed into the waveform shaper network which converts the current steps into a voltage waveform. As may be recalled, the 7960 System operational characteristics require a so called constant-time, end-point, vector-generation mode in which the CRT beam is deflected from a previous end-point address to a new end point address in a straight line and in a constant time, T, regardless of the distance between the points. A further requirement imposed by the scanning system was that the beam move between the points at a rate linear with time. In other words, the beam would move one quarter of the distance between points in a time T/4, half the distance in T/2, etc. In order to achieve these two objectives, it is necessary to generate a deflection waveform in which the change in current or voltage from one level to the next takes place in a constant time T and at a linear rate. The shaper output, when integrated, provides such a waveform. The time period, T, is program selectable to be either 32 or 256 microseconds. Restrictions on dynamic ranges of the circuits limit the maximum positional change in any one cycle to ⅛ of the total X, Y positions in the 32-microsecond mode, and ¼ of the total X, Y positions in the 256-microsecond mode. The output of the integrator is then fed into a preamplifier which provides an impedance match with the deflection-yoke power amplifier and also converts the single-ended input signal into a push-pull output signal.

The deflection-yoke power amplifier provides the current into a high-performance push-pull deflection yoke for driving the 5-inch record and scan CRTs. The yoke was selected for maximum perpendicularity and linearity, and minimum residual magnetism (or hysterisis). The power amplifier drives a lower performance, push-pull deflection yoke when connected to the 17-inch display CRT. The maximum display area utilized is 10 inches square.

The record and scan CRTs are provided with optically-flat faceplates for utilization with an optics system. Because of the flat faceplate and the fact that a change in deflection current produces a proportional change in the sine of the deflection angle, an optical distortion known as pin-cushion is observed at the faceplate. This distortion can be explained best with the aid of Figure 1–11. If the face of a CRT had a radius of curvature equal to the distance from the center of the deflection coil to the screen, the deflection distance A would be proportional to the sine of the deflection angle and thus to the deflection current. The image thus produced by independent X, Y deflection would appear, when viewed from a distance, to be undistorted as indicated by the inside box in the figure.

With flat-faced CRTs, however, the deflection distance, A^{1}, is proportional to the tangent of the deflection angle and thus proportional, non-linearly, with deflection current. The effect of this is that deflection distance increases somewhat faster than the current. Under these condtions, the X and Y deflection components interact, producing the pin-cushion pattern shown in the diagram. For the maximum angles of deflection utilized in the record and scan CRTs, the maximum displacement error at a corner of the image would correspond to approximately +6% (proportionate distance between B and B^{1}). To correct for this error and meet the requirements for positional accuracy, a distortion correction circuit is utilized to provide correction which can be expressed mathematically as follows:

$$\Delta X = - KX (X^2 + Y^2)$$

$$\Delta Y = - KX (X^2 + Y^2)$$

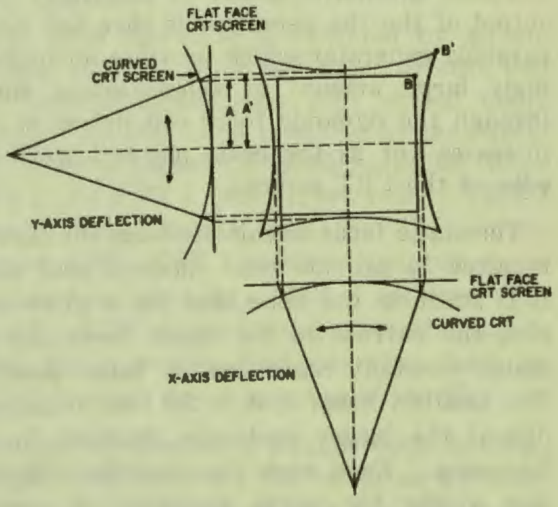


Figure 1-11. CRT Pin-Cushion Distortion.

where ΔX and ΔY are the deflection correction currents, K is a proportionality constant which is a function of the system geometry, and X and Y are the deflection currents.

Focus Control

The focus control system provides two separate and concurrent functions:

1. A dynamic control to compensate for beam defocussing as a function of beam position.
2. A static control to generate the correct size of the CRT beam as determined by the programmed line-width selection.

The requirement for dynamic focus compensation is, like pin-cushion, caused by the optically flat faceplate of the CRT. As the beam is deflected off-axis, the distance to the screen increases and, for a constant value of focus current, the beam defocusses. By changing the current in the focus coil, and thus the magnetic field, the beam can be refocussed at any point on the CRT screen.

Because the CRT geometry is radially symmetrical, the focus error function is also radially symmetrical and mathematically approximates a parabolic function. The block diagram of the focus control is shown in Figure 1–10.

The rho generator produces a signal which is the approximate vectorial addition of the X and Y deflection components referenced to the electrical geometric center of the CRT. The output of the rho generator is then fed into a parabola generator which provides an increasingly large amount of compensating signal through the dynamic focus coil driver as rho increases (or as the beam moves toward the edge of the CRT screen).

The static focus circuit produces the current required to provide four different spot sizes. It is static in the sense that for a given spot size, the current in the static focus coil remains constant regardless of beam position. The smallest beam spot is the true focus condition; the larger spots are obtained by defocussing. Each spot size provides different line widths for image recording of vectors. The line widths are provided in the ratio of 1:2:4:6 where the minimum spot size or line width relative to the 1.2-inch square film image is less than 0.001 inch. Line-width control is provided by relay selection under computer control.

Intensity Control

Intensity compensation is required to maintain constant beam brightness in the CRT's. Two operating conditions account for this requirement.

1. Beam sweep speed. Because the vector time, T, is constant (either 32 or 256 microseconds) regardless of vector length, the beam sweep speed or velocity varies. In order to maintain constant brightness, the CRT beam current must be increased proportionately with vector length.
2. Line width. As line width is increased, the beam current is effectively spread out over a larger area. In order to maintain equal brightness over all line widths, beam current must be increased proportionately to the increased line width.

The requirement for intensity compensation applies only to the record and display CRT's: the record CRT to provide an even exposure of the film; the display CRT to provide an evenly illuminated display. The beam velocity range of the scan CRT is much more restricted and only the basic line width is utilized. Thus, no compensation is required for the scan CRT.

Dynamic beam intensity compensation is accomplished by determining the status of the three variable quantities:

1. Vector length (continually variable from zero to ¼ full image size).
2. Line width (basic, 2X basic, 4X basic, 6X basic).
3. Vector time (32 or 256 microseconds).

A block diagram of the intensity control circuits is shown in Figure 1–10.

The length of each vector is determined by sampling the X, Y deflection signals, differentiating and rectifying these signals, and then feeding them into a rho generator. The rho generator produces an output signal which is the approximate vectorial addition of the change in the X and Y deflection components.

The output of the rho generator is thus a signal proportional to vector length or velocity. This signal is fed into a circuit called a line-width multiplier. This circuit is controlled by the status of two digital lines from the computer which relate to the particular line width selected at any given time. The signal out of the line width multiplier is proportioned in the same ratio as the line width options (1:2:4:6). For example, if the 4X basic line width is selected, the output is four times as great as when the basic line width is selected for a vector of a given length. Another input to this circuit is the vector time selection. The signal level, as described above, is further modified as a function of the vector time, 32 or 256 microseconds.

The CRT beam intensity is, for all practical purposes, linearily proportional to the beam current. The intensity control circuit changes grid voltage to control this beam current. However, the CRT grid voltage-to-cathode (beam) current transfer charactertistic is not linear. Therefore, it is necessary to provide a non-linear function generator circuit which compensates for the CRT charactertistic such that the intensity compensating signal, as derived from the output of the line width multiplier, provides a non-linear grid voltage signal in such a way as to provide the proper level of beam current. The final block is the intensity drive amplifier which is a gated amplifier controlled by the blank/unblank line from the computer.

The intensity control is capable of providing an intensity level for film recording that permits lines of varying lengths and thicknesses to maintain a density tolerance of ±0.1D about a nominal level of 0.7D after rapid processing.

Position Pencil Control

The position-pencil control system allows the pencil, when in contact with the conductive glass screen of the graphic console, to be located by the computer and a CRT beam (blanked or unblanked) to appear at the pencil location. By sampling at a rate high in comparison with the motion of the pencil, the position-pencil control system can maintain the position of the beam under the pencil, making it appear that the pencil traces an image on the CRT screen.

The block diagram of this system is shown in Figure 1–10. The conductive screen is a piece of glass 14 inches square, coated with a thin transparent layer of tin oxide and placed directly in front of the display CRT. A voltage is alternately applied to the screen through the X, Y glass switches causing a voltage gradient to develop on the screen, which is oriented left to right (X) or top to bottom (Y). The pencil when in contact with the screen will thus alternately detect a voltage which is proportional to the distance that the pencil is from the left side of the screen or the top of the screen. This voltage is fed into one side of the voltage comparator. The opposite side of the comparator is supplied with a signal alternately from the X and Y deflection decoder. A comparison is thus made at a given time of the Y position of the pencil and the Y position of the CRT beam or the X position of the pencil and the X position of the CRT beam. The output of the comparator is two digital lines to the computer which indicate that the pencil coincides with the CRT beam, is to the right or left (for X) or above or below (for Y) the CRT beam, or that the pencil is not touching the conductive screen.

Programs have been written which use the interpretation of the digital output of the comparator to direct the CRT beam to coincide with the location of the pencil to accomplish pencil tracking.

Scan Control

The scan operation is initiated by an unblanked (visible) beam deflection, or scan vector, written on the scan CRT screen under program control. Light produced by the sweeping beam is sensed by photomultiplier tubes PMT 1 and PMT 2 (see Figure 1–10). PMT 1 receives light directly from the CRT screen. Consequently, PMT 1 receives light whenever the beam is sweeping. PMT 2 receives light from the CRT screen through the film image. Thus, PMT 2 receives light only when the beam sweeps through the clear areas of the film.

The object of writing a scan vector is to intercept the lines on the film image as the beam sweeps. Thus, assuming that the film image is composed of black lines on a clear background (positive mode), the beam light sensed by PMT 2 will be momentarily interrupted when the sweeping beam intercepts each line. Assuming that the film image is composed of clear lines on a black background (negative mode), PMT 2 will sense light momentarily when the sweeping beam intercepts the clear lines.

However, the PMT 2 output depends on the relationships of beam speed, beam spot diameter, and width of the intercepted line (Figure 1–12). Section A shows positive-mode scanning, in which the line width is greater than the beam-spot diameter. In this case, when the beam intercepts the line, PMT 2 senses total darkness and, accordingly, produces a maximum amplitude level. Section B shows the same beam spot intercepting a line whose width is smaller than the beam diameter. In this case, PMT 2 does not sense total darkness but a degree of light dimming called *gray level*. The PMT 2 output then reflects this gray level with a signal of proportional amplitude. Sections C and D illustrate negative mode scanning where line width is greater and smaller than beam spot diameter.

In practical operation, the most common occurrence is that interceptions with lines of average width produce different gray levels. The PMT 2 signal indicates a gradual transition from light to dark and dark to light. This corresponds to the gradual dimming of the beam in either transition.

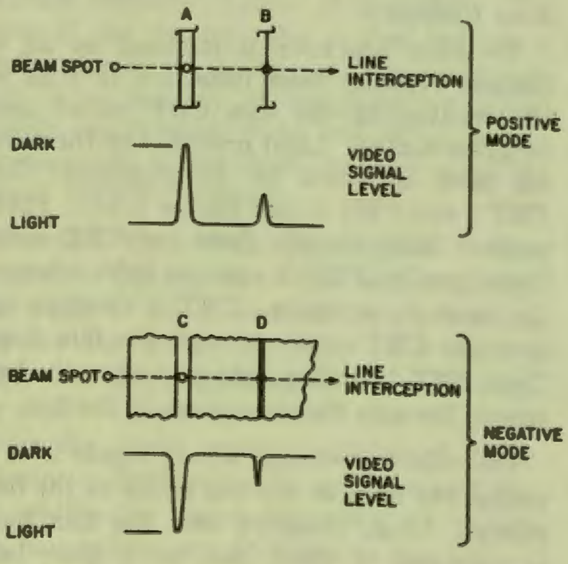


Figure 1-12. Scan Signal Generation.

Since line width is sensed by PMT 2 in terms of varying gray levels, a search is required to make sure that all interceptions, or meaningful levels, are accounted for in the scan process. This search is done by comparing the signal level with a reference level which can be varied by predetermined values. Figure 1–13 illustrates this comparison with examples of typical vector scan operations. In this figure it is assumed that the positive and negative versions of the same image are being scanned and that the sweeping beam makes two line interceptions.

Section A of the figure shows that the reference level, which lasts for the length of the scan vector, can be set to any one of 64 threshold levels by computer program.

Section B shows (1) the superimposed signal and reference levels as they are placed for comparison at the differential amplifier (Figure 1–10) and (2) the threshold levels which are lowered until they cross the higher level signal. Thereafter, the search process requires an additional scan operation with a lower reference threshold to cross the lower-level signal.

Section C indicates the output signal transferred to the control unit to indicate a strike in positive-mode scanning.

Sections D and E indicate the search for signal levels in the negative mode in which the program raises the threshold until it crosses

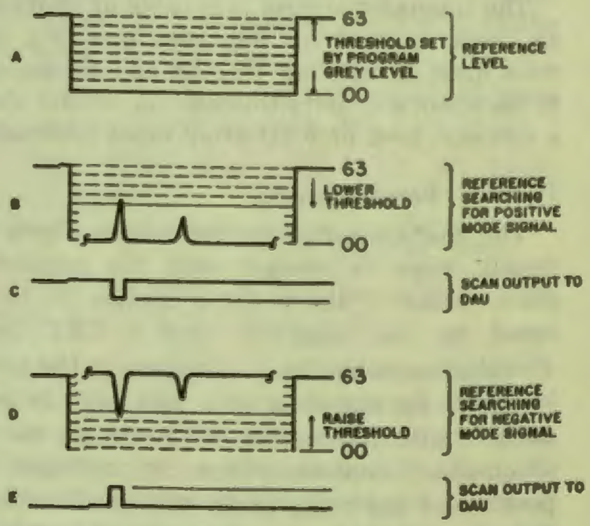


Figure 1-13. Scan Output Generation.

the higher-level signal; an additional scan operation with a higher reference threshold is required to cross the lower-level signal.

In the section on deflection control, it was stated that a requirement of the deflection signal was that it be linear with time. The reason for this is that the strike or output data from the scanner is sampled at 16 equal intervals during one vector time. Thus, in the 32 microsecond mode, the scan output is sampled and stored every 2 microseconds; in the 256 microsecond mode, the output is sampled every 16 microseconds. By making the deflection signal, and therefore the scan vector, linear with time, the time segment during which an output was sensed from the scanner can be correlated to beam position. This is extremely useful in either one of two ways. It provides the equivalent of 16 separate vector scans in one scan vector time period thereby increasing the effective scan rate by 16, or it can be considered to increase the resolution capability of the scanner for a scan vector of a given length by a factor of 16. The example shown in Figure 1–14 illustrates this point. A scan vector is generated which traverses from point A to point B, intercepting four lines on the film image. If the scan output was sampled only once per scan vector, the only information which would be retrieved would be that a line or lines were located somewhere between points A and B. By sampling 16 times during the generation of the single scan vector, not only was it determined that there were four lines but the locations of the lines relative to points A and B can be computed to within one part in sixteen of the distance between the points.

The maximum length of the scan vector (and therefore the CRT beam velocity) is restricted by the band width of the scan deflection circuits which are in turn restricted in order to minimize noise. The resolution of the scanner is primarily a function of CRT spot size. The scanner can resolve lines 0.0005 of an inch thick separated by 0.0015 of an inch relative to the 1.2-inch-square film image.

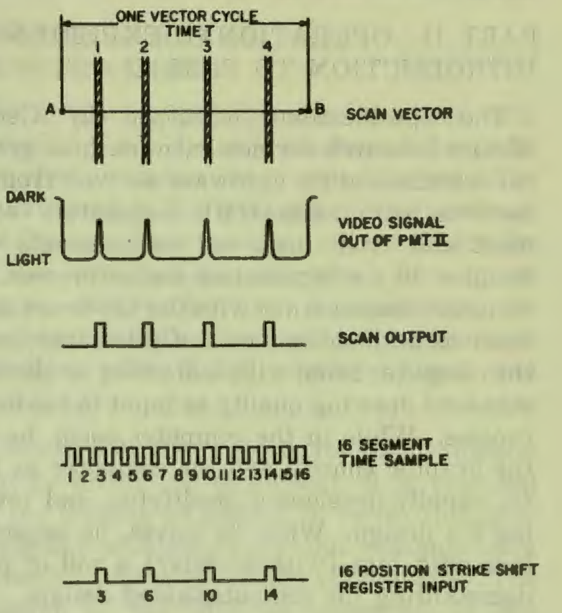


Figure 1-14. Scan Output Time Sampling.

Cathode-Ray and Photomultiplier Tubes

The CRT's utilized for scanning and recording are identical except for their phosphors. Each is a 5-inch, round, high-resolution CRT with an optically flat faceplate. Both the deflection and focus is magnetic. Nominal spot size at the CRT screen is 0.001 inch.

The scan CRT employs a P16 phosphor which was selected because of its ultra-fast light-decay characteristic. It has a spectral energy distribution which peaks around 3800 Å (violet and near UV). The scan lens is color-compensated for this spectrum and the scan detection photomultipliers employ an S11 photo cathode which is quite sensitive to the P16 spectrum.

The record CRT employs a P11 phosphor which was selected because of its high light-output capability, which is required for exposing film. It has a spectral distribution which peaks around 4600 Å (blue) and is considered medium fast relative to its decay characteristic.

The display CRT is a 17-inch, rectangular CRT with a curved faceplate and requires magnetic focus and deflection. The nominal spot size is 0.020 inch and the P19 phosphor peaks around 5900 Å (orange). This phosphor was selected for its long persistance characteristic. The usable display area is 10 inches square. The focus control is not connected to the display CRT. Because of the larger spot size and the curved faceplate, a permanent magnet focus was determined to be adequate.

PART II OPERATIONAL EXPERIENCE

INTRODUCTION TO PART II

The specifications developed by General Motors Research for new man-machine graphical communication hardware evolved from experience with early GMR Laboratory equipment and from projected requirements of a designer in a computerized design process. The designer, unconcerned with the hardware specifications outlined in part I of this paper, enters the computer room with a drawing or sketch of standard drawing quality as input to the design process. While in the computer room, he uses the graphic console and the computer as tools for rapidly developing, modifying, and reviewing his design. When he leaves, he expects to take with him (without delay) a roll of paper documenting his computer-aided design.

The remainder of this paper evaluates the degree to which the hardware has approached the requirements of a designer in the process of design. In addition, some programming techniques for improving the effectiveness of the hardware by testing and calibration are described.

THE GRAPHIC CONSOLE AS A MAN-MACHINE INTERFACE

Continued use of the graphic console by a large number of people has led to some conclusions regarding the equipment as a tool in a design environment.

The CRT and pencil have proven to be a highly successful man-machine interface in spite of the small screen size (10″ x 10″). Except for a period of two months when a lower persistance phosphor tube was tested in the unit, there have been no serious operator complaints directed toward use of the display tube. The lower persistance phosphor caused a highly objectionable flicker when large amounts of data were displayed. The operator antagonism toward this flicker was sufficiently severe in some cases to keep the individual from the equipment.

The position-indicating pencil has been found to be functionally very suitable. One reason for this is the positive action-reaction; that is, the program is alerted whenever the pencil touches the screen. Contrast this action with a light pen which must be placed close to a specific area of interest and requires in addition some type of manual switch to alert the program. Another advantage of the pencil compared to a light pen is that pencil position can be determined in milliseconds when the pencil is pointing at a void portion of the image area. This quick position location is essential in an on-line problem solving environment. Users of the pencil must be able to quickly locate in real dimensions any point of the display area. With a light pen it is necessary for the user either to wait for a searching operation or to point at part of a display and then track the pen to an area of interest.

The speed and positive action characteristics make the pencil a comfortable tool in an alphanumeric display-and-correct mode in which the user points to a character of the display and uses the alphanumeric keyboard for correction. The user of the pencil need only touch the screen in the general vicinity of a particular character and the program can determine which character is to be replaced. Users have adapted very rapidly to using the pencil as a pointer.

One disadvantage of a position-indicating pencil when compared to a light pen is that the pencil does not establish a positive reference to a particular line of the displayed composite line image. It is necessary to compare pencil position with the position of each of the displayed lines to determine at which line the pencil is pointing.

From these experiences with the use of the pencil, we have learned that the functional characteristics of a device for pen-pencil man-machine communication should include modes such that:

1. The control program is alerted each and every time the pencil touches the display surface.
2. The control program is able to determine within milliseconds the position of the pencil on the screen independently of the presence or absence of any display.

3. The control program is able to establish a positive identification by the pencil of a particular line of a displayed image.

The use of a man-computer pen or pencil for the input of graphic data is an interesting topic in itself. When analyzing the tracking mode, however, it is difficult to separate the characteristics of the hardware, the programming and the application. While the pencil or a light pen could be used for entering images consisting of connected straight lines, a highly sophisticated program would be required to track and accurately digitize drawings of curves which are connected to form an image. In the GMR DAC–I System, precision graphic data input is entered through the paper input facility of the image processor.

The program status lights and control keys have also proven to be excellent both for the programmers and for the man at the console. The alphanumeric keyboard, on the other hand, has been an object of much discussion from which the following controversial points have arisen:

1. The keyboard should be a standard typewriter keyboard to take advantage of the speed of those who know how to type.
2. The typist-designer intersection is small; therefore, use a keyboard that is arranged in some order to minimize learning.
3. The typewriter should be installed immediately in front of the display tube to maximize display feedback of a typed message.
4. The space in front of the display tube should be reserved as a work area for listings, drawings, note pads, etc.
5. The typewriter would be of great value for long message input.
6. No long messages should be entered at the graphic console.

This type of discussion can be and has been carried out to great lengths by human factors people. Our experience has shown that an alphanumeric keyboard is difficult to cope with and not a natural device for man-machine communication. Until a pencil entry device accompanied by character recognition is available, however, the keyboard must remain an important part of the man-machine interface.

EVALUATION OF THE IMAGE PROCESSOR

The graphic console provides the user with the facility for a dynamic display of comparatively low accuracy and repeatability. The voltage pencil is also dynamic in its functional capabilities and not intended to be used for the entry of precision graphic data. The image processor on the other hand is expected to provide the more static man-machine graphic communication with higher precision input and output.

When establishing criteria for the evaluation of image processing equipment, it is essential that the criteria be defined in terms that give an overall evaluation of the unit, encompassing electronic circuits, optics and photography. The resolution of the CRT as measured by the shrunken raster method, for instance, is of little interest to the user of an image recorder. For Design Augmented by Computers the user wants to know, first of all, the line width and resolution as measured on the paper of the hard copy output from the recorder. If he must supply his own hard copy machine, the user wants the specifications as measured on the processed photographic film. From the user's standpoint, all specifications must be based upon the response of the unit in the final output state and the specifications must be measurable at that final output state.

What follows is an important subset of image processing evaluation criteria and programs resulting from daily use of the image processor since early 1963. The criteria are specified in a manner that allows rapid evaluation whenever possible. The programs are intended to provide versatility for the evaluation of the equipment over a wide range of conditions and requirements.

Accuracy

The user of image processing equipment for Design Augmented by Computers wants to know the accuracy of the positioning of a point on the output image when referred to any other point on the image. He wishes to measure with a rule the distance between two points and know his confidence limits. Accuracy to the user is then defined as the maximum error in the distance between any two points on an image. It is measured by recording a pattern of vectors with known spacings. The distance between vector endpoints is measured on hard copy if provided, on the film image with an optical comparator, or on the film image using the image scanner. The maximum error encountered in measuring the distance between any two points is the accuracy characteristic of the unit. With a little experience, the user learns the particular pattern and vector endpoints that demonstrate the maximum error.

Figure 2–6 shows the accuracy errors of various points (referred to the center point) on the scanner image magnified by a constant factor. Note that although the accuracy figure is stated in terms of per cent of image size, the error measured is an absolute value that includes pincushion error (distortion corrected) and all other electronic, photographic and optical errors. Note also that this method of measuring accuracy results in a figure nominally twice that of those techniques that measure accuracy as the error of one point referred to an origin. This method does, however, yield a value of accuracy such as a designer would measure with his rule.

Users of the equipment are not completely satisfied with the accuracy (approximately 1%) resulting from uncalibrated scanning input and output recordings in spite of the fact that the equipment represents an advanced state of the art. Calibrated input and output accurate to 0.2% barely meets the requirements of static man-machine communication.

Stability and Repeatability

To the user of DAC–I image processing equipment, stability means freedom from drift of analog components including power supplies and deflection circuitry. This specification is important since it is essential in maintaining constant raster shape and size on a recorder and scanner CRT. Accuracy can be improved by calibration procedures, if and only if the hardware stability is such that the calibration runs can be spaced at practical intervals of time. A practical interval of time might be approximately one hour unless some form of automatic calibration is provided.

Repeatability is the degree of capability of the hardware to exactly duplicate an output condition after an intervening number of random input conditions. Other definitions of repeatability which are dependent on repeating a sequence of inputs are of little or no value to the users of image processing equipment.

Imperfect vector endpoint repeatability shows up dramatically in circles (see Figure 2–1) which do not close properly and at intersections of a number of lines which are all supposed to pass through the same point. Another example of repeatability error is the case where two or more lines are spaced close together. The lines may touch or even cross if the repeatability is poor. Although this type of error may be aesthetically less troublesome than poor repeatability at intersections and at closing points of circles, it would be important if a user were trying to evaluate positions of edges of parts which are to fit together.

It has been observed that it is the natural tendency of the human eye to notice the worst case of repeatability on recorded output even if all the remainder of the drawing represents high-quality output in terms of repeatability and accuracy. The width of the lines affects the aesthetic appearance of the output. The observer immediately spots repeatability errors at the junction of thin lines while the same errors with thicker lines are overlooked. Since thin lines are normally more desirable on recorded output, some compromise is frequently required.

Figure 2-1. Circle Pattern for Demonstrating Repeatability.

The pattern in Figure 2–1 has proven to be excellent in showing the limit of the recorder repeatability for quick approximations. The errors in the closing of the circles can be observed in terms of line widths and then converted to per cent of image size. The maximum repeatability error for the circles, for instance, might be approximated at one line width.

Repeatability is also an important factor in scanning operations. In line tracking, repeatability is particularly important when input lines have small changes in curvature and the positions of inflection points are important. Repeatability errors also put a limit on the effectiveness of calibrating for improved accuracy.

Experience has shown that the effective repeatability of the scanner can be improved by repeated scanning at a given region and then averaging the results of these several scans. However, this does not eliminate repeatability errors, but merely reduces the magnitude of the errors when using the scanner.

Line Width and Intensity

Recorder line width must be defined in terms of measurements on micro-densitometer recordings from the output film. The width of the line is arbitrarily defined as the width of the line at the mean light transmission level where the mean is the average of the film background light transmission and the transmission at the peak of the line. Use of the microdensitometer gives a numerical measurement of the line width independent of human observation. Line intensity of recorder output film is defined as the optical density of a line at the center point.

Variations in the optical density and line width are very noticeable to users of the image processor recorder. Line density and thickness variations with vector length frequently result in short vectors having a greater density than the longer vectors. This is particularly objectionable in areas of a drawing where information in the form of sharply curved lines is concentrated. In this case, sharply curved lines require many short vectors with a corresponding high density and loss of fine detail. In addition, small alphabetic characters, normally readable, may become illegible when line width and intensity are out of adjustment for short vectors. Variations in line density and width with vector direction and position has also proven to be very noticeable to the equipment users. Even if drawings are made accurately with only small repeatability errors, poorly controlled line widths and peak densities will make the recorded output look poor.

The pattern of Figure 2–2 was generated by a program that has proven excellent in testing for the conditions mentioned above. The position, direction, and lengths of the vectors were selected at the graphic console to show how uniform the line widths and intensities are over a wide range of these parameters.

Film Processing Time

It has been observed that any time an operator has nothing to do, or no drawings or displays to review, a forced delay of one minute or even half a minute becomes annoying. Even though the rapid film processor develops, fixes, rinses and dries film at a rate of approximately one frame every three seconds, a minimum of 25 seconds is required to process the first frame plus three seconds for each additional frame.

Whenever practical, the graphic console operator should be given a choice of other functions to perform while film is being processed for the best man-machine interaction, at least until film processing speeds increase by another order of magnitude or some other more rapid

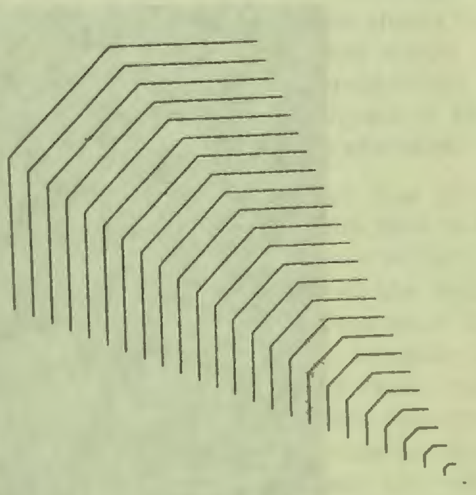


Figure 2-2. Intensity Control Pattern for Demonstrating Line Intensity and Width Conditions.

form of static image documentation is available.

Scanner Sensitivity

The sensitivity of the scanner is defined by the thinnest line the scanner can consistently detect and the range of thresholds for this detection. Typically, the scanner, using the paper input facility, can consistently detect lines .01" wide (on the 22 x 22 inch paper) over a range of three thresholds.

A considerably more complex algorithm can be used to define the sensitivity of the scanner over a range of line and background densities. This is essential for determining which photographic images can be expected to be scanned. A simple definition and measuring technique is necessary, however, for determining the hardware sensitivity.

Figure 2–3 shows what is displayed on the graphic console within two minutes after the start of the sensitivity test. This display demonstrates to the user that lines nominally 9.8 raster units wide (referred to the 0–4095 raster unit image) can be detected over a range of 17 thresholds. It further points out the line thickness that one may expect at each of the threshold levels. It is now standard practice to test the equipment daily using test programs which are highly application oriented and which can analyze the equipment and display the results for immediate review by the maintenance engineers or by users of the equipment.

TEST PROGRAMS

It is no small task to define the exact performance status of electronic systems which are comprised only of digital components. It is still harder to define the status of electromechanical units. In the first instance, it is the complexity of the system that causes the difficulty even though the system operates at a discrete level for performance evaluation. In the second case, the mechanical positioning involved may result in a series of discrete levels that must be evaluated. Defining the performance level of an analog-digital-mechanical system such as the image processor is still more difficult because of the continuous range of the operational status which is biased by the discrete digital-mechanical levels. Test programs which are understood and used by both the maintenance people and the operators of the equipment have proven essential in obtaining the maximum performance from the equipment for the operator and computer time involved.

Hardware test programs for the DAC–I system have been written primarily for the scanner and the recorder. The recorder programs basically provide a variety of test patterns to

```
   LINE SENSITIVITY TEST
      06/24/64       2027.1
   LINE WIDTH =         9.8
THRESHOLD    WIDTH   LINE CENTER
   10         0.00       0      0
   11         2.83    2432   1745
   12         1.41    2432   1745
   13         2.30    2432   1745
   14         4.24    2433   1744
   15         4.24    2432   1745
   16         4.95    2432   1745
   17         4.95    2432   1745
   18         5.30    2432   1745
   19         5.66    2432   1745
   20         5.66    2432   1745
   21         7.95    2433   1744
   22         7.07    2432   1745
   23         7.07    2432   1745
   24         6.19    2432   1745
   25         7.07    2432   1745
   26         6.63    2432   1745
   27         5.66    2432   1745
   28       178.90    2494   1684
   29       178.90    2494   1684
   30       178.90    2494   1684
```

Figure 2-3. Scan Sensitivity Test Program Display.

show recorder repeatability, accuracy, line intensity, width and smoothness. The patterns are selected to demonstrate dramatically each characteristic to be tested. This frequently allows preliminary evaluation at the projection station. Detailed evaluation is made with test instruments when necessary. Test programs for the scanner include the program for line sensitivity described earlier, line resolution, threshold characteristics and scanner and recorder accuracy and repeatability.

With a man-machine console, test programs should include the following features:

1. rapid measurements
2. rapid editing of the measured data so that only a small amount of important information will be displayed
3. rapid display of results
4. options to record results on film
5. pictorial rather than numeric display of measurements when requested
6. options for conventional printout
7. facilities to easily change and display program parameters.

Permanent records of test program results are kept to monitor and signal when a long-term decrease in hardware performance should be corrected. These records also aid in the evaluation of changes in the hardware.

Sometimes adjustments or changes are made that result in a temporary improvement in the quality of the hardware. The test program results, when analyzed over a period of time, show if the change corrected a problem or merely adjusted around it temporarily. To illustrate some of these desirable characteristics of test programs, we will outline as an example the main features of the calibration test program for scanner and recorder accuracy.

The calibration test program uses the scanner to measure the nonlinearities of both the scanner and recorder caused by electronic and optical effects. The magnitudes and positions of these non-linearities are displayed for immediate review and stored for later calibration of both scanner input data and output recordings.

The reference for linearity and size measurements is a metal plate with grid lines inked on its surface. The locations of all the intersections have been previously stored in the computer. The grid is exposed onto film through the paper input station, processed, and moved to the scan station.

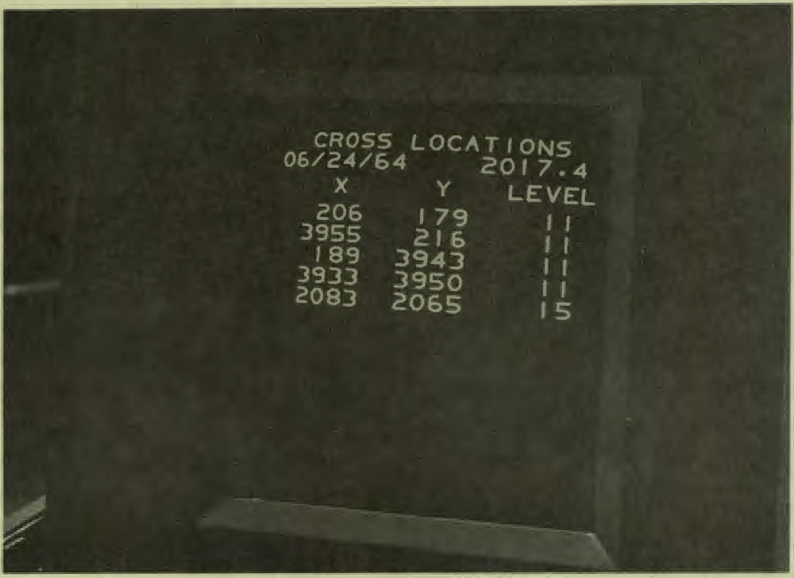
Before executing a complete scan of the grid, a preliminary scan for the four-corner crosses and the center cross is made. (See Figure 2–4) This allows the operator to check the image size and the optical alignment of the system and insures that the entire image to be scanned is aligned optically and electronically within the scan raster.

Figure 2–5 shows the type of alphanumeric information displayed after scanning every grid intersection and comparing the scanned itersection locations to the reference values. Here a large amount of data has been condensed to a few numbers giving the most pertinent information about the raster size and linearity of the raster. This reduces the question of acceptability of the machine to a matter of comparing the numbers on the graphic console to figures which are previously defined to be acceptable.

For diagnostic purposes, the information shown in Figure 2–6 is most useful. The errors associated with each point are magnified by a factor and added to the reference points to produce an exaggerated representation of the non-linearity and size of the raster (see Figure 2–6. This graphic presentation shows the non-linearity of the raster size and shape. A numerical figure for scanner repeatability is displayed by the calibration program if repeated scans of the grid pattern are executed.

Experience has shown that test programs such as the calibration program have been able to define scanning status before actual digitizing begins and have maximized the usefulness of the scanner operation. Programs like the calibration program are also valuable in separating hardware deficiencies from errors in new programs; that is, if new programs fail but the test programs run normally, then it is likely that the new programs are either in error or expecting too much from the analog hardware.

The calibration program illustrated above has resulted in improved scanning and record-



```
CROSS LOCATIONS
06/24/64        2017.4
  X        Y      LEVEL
 206      179       11
3955      216       11
 189     3943       11
3933     3950       11
2083     2065       15
```

Figure 2-4. Corner and Center Crosses on Scan Program.

ing to the extent of nearly one order of magnitude. The hardware has proven itself sufficiently stable to allow a significant improvement in scanner accuracy by a calibration procedure which uses the error data measured by the test program.

The recorded output can also be calibrated to show a significant improvement in accuracy. The scanner is used to scan recorded output and the errors of the recorder raster relative to the scanner raster are then modified by the scanner errors and stored for calibration by recording programs. Two of the most important values of these GM test programs are that they provide a good overall test of the system and they are oriented to evaluating the user's requirements.

SUMMARY

The IBM 7960 Special Image Processing System was designed and built by IBM to specifications provided by the General Motors Re-

```
SCANNER RASTER
06/24/64     2019.0

VALUES ARE RELATIVE TO
PAPER INPUT GRID

HORIZ SCALE IS TOO LARGE, PCT-
   RASTER IS TOO SMALL          0.08

VERT. SCALE IS TOO LARGE, PCT-
   RASTER IS TOO SMALL          0.40

ACCURACY FIGURE IS-    0.89  PCT.
                  X    0.44
                  Y    0.88
```

Figure 2-5. Scanner Accuracy and Raster Size.

Figure 2-6. Accuracy Errors Multiplied by a Constant.

search Laboratories. The system is the man-machine and image processing hardware for the GM Research DAC–I system.

The design shows how the functional requirements for an image processing system were implemented to achieve a new type of computer input-output system. The components were chosen for characteristics that were compatible with digital computer speed and accuracy. Technologies normally foreign to computer technology were successfully integrated by careful consideration of the interface between components and the effect of each component on the total system performance. Some of the features of the new hardware system such as the high resolution, the excellent accuracy and the rapidly processed film for pictorial input and output have been extensions of the state of their respective arts. These features have also indicated the necessary quality and speed for a graphical machine to interact with a man in the iterations of a design cycle.

The hardware has proven to be valuable as a laboratory tool for the analysis of equipment required in an online computer aided design facility.

The use of image processing equipment in a computing facility has also pointed out some of its interesting operational characteristics. The digital program-analog response characteristic of the hardware makes system testing and status documentation a necessary part of operational procedures. In addition, programming techniques that improve the apparent performance of the hardware must be used. Man-machine programs such as the calibration program described in this paper have proven to be a solution to these requirements.

REFERENCES

1. JACKS, E. L., "A Laboratory For The Study of Graphical Man-Machine Communication," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
2. COLE, M. P., DORN, P. H., LEWIS, C. R., "Operational Software In a Disc Oriented System," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
3. ALLEN, T. R., FOOTE, J. E., "Input/Output Software Capability For a Man-Machine Communication and Image Processing System," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
4. KRULL, F. N., FOOTE, J. E., "A Line Scanning System Controlled From an On-Line

Console," *Proceedings Fall Joint Computer Conference,* San Francisco, California, Oct. 1964, (this volume).

5. LAZOVICK, P. B., TROST, J. C., REICKORD, A. W., GREEN, R. S., "A Versatile Man-Machine Communication Console," *Proceedings (1961) Eastern Joint Computer Conference,* Washington, D.C., Volume 20.
6. STOTZ, R., "Man-Machine Console Facilities For Computer Aided Design," *Proceedings (1963) Spring Joint Computer Conference,* Detroit, Michigan, Volume 23.
7. LICKLIDER, J. C. R., "Man Computer Symbiosis," IRE Transactions on Human Factors in Electronics, Volume HFE, March 1960.

INPUT/OUTPUT SOFTWARE CAPABILITY FOR A MAN-MACHINE COMUNICATION AND IMAGE PROCESSING SYSTEM

Thomas R. Allen and James E. Foote
Research Laboratories, General Motors Corporation, Warren, Michigan

INTRODUCTION

Consistent with the design objectives of the General Motors Research Laboratories DAC–I (Design Augmented by Computers) project,[1] the IBM 7960 Special Image Processing System has extensive input/output (I/O) capabilities.[3,4] In order to facilitate the programmed control of this new hardware, the NOMAD and MAYBE programming languages were developed.[2] However, while these languages provided the programmer with an effective means of controlling the new hardware, they offered little assistance in meeting the hardware data format requirements. The programmer still would have had to convert his output information from his own internal format to the output format required by the hardware. Similarly, all input from the hardware would have had to be converted by the programmer back to a form suitable for his own use. Moreover, these conversion processes are typically very involved and complicated. In short, the programmer was still not in a position to make easy, efficient, and flexible use of the I/O capabilities of the new hardware system. Thus, the need to provide a layer of general purpose I/O software between the programmer and the hardware was very apparent.

The effort to develop a general purpose I/O software capability resulted in a hierarchy of utility routines. At the lowest level of this hierarchy is a set of very basic utility codes. These include numeric-to-BCD (binary coded alphanumeric information) conversion routines, BCD-to-vectors character generation routines,[5] simple display and recording routines, and basic film train operation codes, all of which permit the programmer, if he so desires, to operate very close to the basic hardware without having to understand all of the intricacies of the hardware's operation. Each of these basic codes represents an implementation of only a very small facet of the total I/O capability but, as such, serves as a building block for subsequent level codes in the hierarchy. These higher level I/O utility codes are progressively more inclusive in their total I/O capabilities and, at the same time, relieve the user of the task of dealing with the hardware on its own terms.

This effort to develop an I/O software capability was not designed to culminate in one single all-inclusive routine. Rather, the aim was to produce a set of sophisticated, general purpose, problem oriented subroutines and source languages statements, each of which would serve as a powerful tool in the utilization of the various I/O capabilities of the new hardware. The body of this paper describes five representative utility subroutines, three source language I/O statements, and some typical examples of their application.

UTILITY SUBROUTINES

This section describes five different tasks and the I/O utility subroutines which were developed to meet their requirements. The basic problem associated with all of these tasks is that of requesting various types of information from the graphic console operator. Hence, the emphasis is on the implementation of the various input devices associated with the graphic console. The graphic console display CRT (cathode ray tube) is used to indicate what type of information is required.

Alphanumeric Input

Subroutines names, variable names, titles for recorded film output, etc. constitute one type of information frequently required from the graphic console operator. This information is most conveniently handled by the programmer in the form of BCD character strings. Thus, the task here was to provide a facility for requesting and accepting these strings. Two input devices are appropriate: the alphanumeric keyboard and the card reader (see Figure 1).

In addition to the primary requirement that the I/O subroutine for this task be simple for the programmer to use, two other requirements were felt to be important. First, there was the need to give the graphic console operator immediate feedback which would permit him to verify that each character has been entered correctly. Concomitant with this was the need to provide the operator with a means of correcting erroneously entered characters.

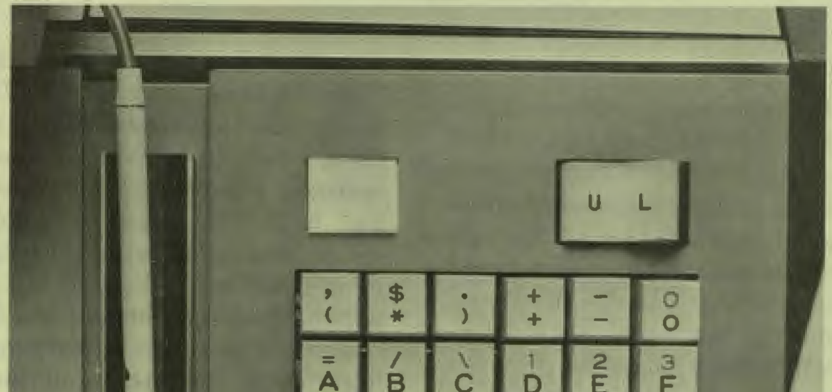


Figure 1. The 7960 Alphanumeric Keyboard and Card Reader.

The subroutine RINSE (*R*equest for INformation SubroutinE) was developed for this task. To use this subroutine, the programmer specifies three items of information: (1) the symbolic name (i.e., location in core) of an array which contains the request message in BCD format, (2) the maximum number of BCD characters which the operator is allowed to enter, and (3) the symbolic name of an array in which to store these characters. Figure 2 shows a typical set of source language statements for utilizing the RINSE subroutine. The DIMENSION declaration is used to reserve space for an array at compile time. The VECTOR VALUES declaration is used to preset an array (i.e., vector) at compile time. The EXECUTIVE statement generates a call to a subroutine.

In Figure 3 the request message in the example above is shown as it would appear on the display CRT upon the execution of RINSE.

Assume that the graphic console operator wishes to enter the subroutine name: ABC. At this point, the operator can either type in the subroutine name at the alphanumeric keyboard or insert a card in the card reader. If he elects to type in the name, each character will be added, one at a time, to the display. If he inserts a card in the card reader, characters will be added to the display sequentially as they are read from the card. The console operator can intermix these two modes of input. Figure 4 shows the display as it would appear after the subroutine name has been entered.

The operator can delete the last character in the string by depressing the BACKSPACE key or delete the entire string by depressing the RESTART key (see Figure 1). He may then enter more characters. In this manner, erroneous characters may be corrected. Once the input is satisfactory, the operator depresses the END key, the display is terminated and the BCD character string is passed back to the calling program.

```
DIMENSION REPLY(1)

VECTOR VALUES RQUEST = 5,$ENTER NAME OF/SUBROUTINE$

EXECUTE RINSE.(RQUEST,6,REPLY)
```

Figure 2. Source Language Statements for the RINSE Subroutine.

Figure 3. RINSE Subroutine Display.

Numeric Input

Another type of information which the programmer may require from the graphic console operator is numeric data. The task here is one of providing a facility for initializing and/or modifying data variables. Two subroutines were developed to meet this requirement. The difference between these two routines stems from the two basically different ways in which data variables can be defined: as elements of an array or as a set of distinct variables which may be widely scattered throughout memory. In the first case, each variable is referenced by giving the name of the data array and the subscript of the particular item. In the second case, each variable has its own unique name.

Figure 4. RINSE Subroutine Display.

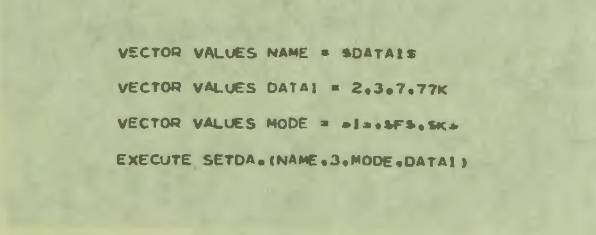
In addition to the three task requirements mentioned in connection with the RINSE subroutine (i.e., ease of use, visual feedback to permit verification, and an error correction procedure) there was the additional requirement that the operator be able to enter numeric data in either floating point, integer or octal mode. For this task, the only input device which has been implemented is the alphanumeric keyboard.

The subroutine SETDA (*SET* up *D*ata *A*rray) permits the inspection and modification of items in a data array. To use SETDA, the programmer specifies four items of information (see Figure 5): (1) the BCD name of the data array, (2) the length of the data array, (3) the symbolic name (i.e., location in core) of a second array which defines the mode (integer, floating point or octal) of each item in the data array, and (4) the symbolic name of the data array. If each item in the data array has the same mode, the mode need only be specified once for the whole array.

Figure 6 shows the display which will be generated upon execution of SETDA, as indicated in Figure 5.

At this point, the operator can modify the value of the current item, step the display to the next item in the data array, define the subscript of any item in the data array which he wishes to see displayed next, or terminate the subroutine.

The operator enters a new value for the current data item by using the alphanumeric keyboard. The visual feedback and error correction procedures are identical to those described for the RINSE subroutine. Figure 7 shows how the display would appear during this process. The operator can cause the new value of 27 to be stored in DATA(O) by depressing the END key.



```
VECTOR VALUES NAME = $DATA1$

VECTOR VALUES DATA1 = 2,3,7,77K

VECTOR VALUES MODE = $I$,$F$,$K$

EXECUTE SETDA.(NAME,3,MODE,DATA1)
```

Figure 5. Source Language Statements for SETDA Subroutine.

```
THE CURRENT ITEM IS
 DATA1 (    0)

ITS CURRENT VALUE IS
            2

ITS MODE IS
 I
------------------------
TO DEFINE NEW SUBSCRIPT,
HIT '=' KEY
```

Figure 6. SETDA Subroutine Display.

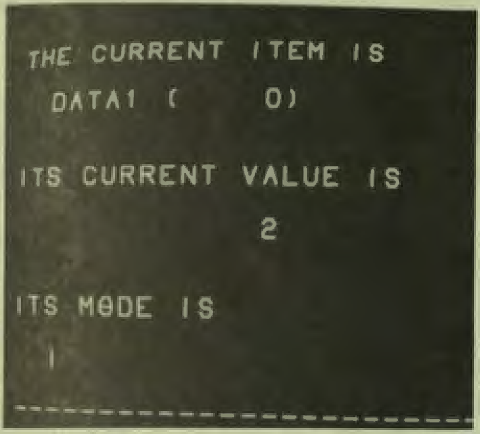
If the operator tries to store a number of the wrong mode, the display changes to that shown in Figure 8.

If the operator wishes to step the display (Figure 6) to the next item in the data array, he depresses the alphanumeric key labeled "," and the upper portion of the display changes appropriately. If, on the other hand, the operator wishes to "move" the display directly to any item in the data array, he first depresses the "=" key. The display then changes to that shown in Figure 9 and the operator enters the subscript of the desired item.

If the operator tries to enter a subscript which is too large, the display changes to that shown in Figure 10.

Control is returned to the calling program when the operator depresses the END key from the display indicated by Figure 6.

The second subroutine (SETPAR) developed to facilitate numeric input permits the inspection and modification of a set of distinct data variables. The programmer uses this subroutine in essentially the same manner as SETDA. However, the format in which the current status of the data variables is displayed was altered in an attempt to gain insight and experience in the presentation of information. The information display technique used by the SETDA subroutine can be characterized as the "player-piano" approach: the data items appear as if they were listed sequentially on a long strip of paper which is being moved back and forth past a one-item viewer under the opera-



```
THE CURRENT  ITEM  IS
  DATA1  (      0)

ITS CURRENT  VALUE  IS
                2

ITS MODE  IS
  1
-------------------------
```

Figure 7. SETDA Subroutine Display.

```
ENTER VALUE OF
NEW SUBSCRIPT
```

Figure 9. SETDA Subroutine Display.

tor's control. SETPAR uses a "page" approach as the information display technique. This technique treats the data variables as if they were listed, up to 16 to a page, in a loose-leaf notebook. As each "page" is presented to the graphic console operator, he has the options of modifying the values of any data items listed on that page, turning to the next "page", or "closing the book" (i.e., terminating the subroutine).

Figure 11 shows a typical "page" display.

If the operator wishes to change the value of any data variable in the displayed list, he uses the alphanumeric keys labeled " ↑ " and " ↓ " to move the pointer to the desired data item and then enters the new value of this item using the alphanumeric keys exactly as in the SETDA subroutine. To "turn the page" the operator depresses the "," key and to terminate the subroutine he hits the END key.

```
THE CURRENT  ITEM  IS
  DATA1  (      0)

ITS CURRENT  VALUE  IS
                2

ITS MODE  IS
  1
-------------------------
ILLEGAL MODE, TRY AGAIN
```

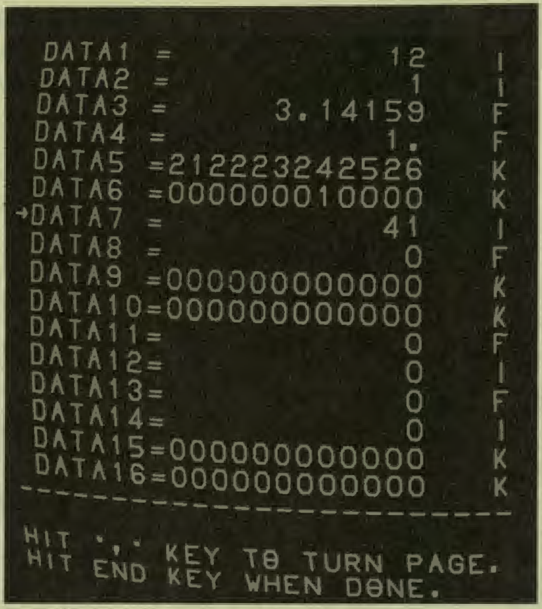
Figure 8. SETDA Subroutine Display.

Positional Data Input

Since the GM Research Laboratories DAC–I system was developed specifically to investigate the field of graphic data processing, it was necessary to provide a capability for presenting the graphic console operator with a display of graphic information and receiving positional input from him relative to this display. The position indicating pencil provided the basic hardware capability necessary to meet this need. The requirements for this task were: (1) to display the graphic information specified by the calling program, (2) to determine the position of the pencil relative to this display, and (3) to return this position to the calling program when the pencil was removed from the screen.

```
ENTER VALUE OF
NEW SUBSCRIPT




-------------------------
ILLEGAL SUBSCRIPT,
TRY AGAIN
```

Figure 10. SETDA Subroutine Display.



```
 DATA1  =            12    I
 DATA2  =             1    I
 DATA3  =       3.14159    F
 DATA4  =            1.    F
 DATA5  =212223242526      K
 DATA6  =000000010000      K
→DATA7  =            41    I
 DATA8  =             0    F
 DATA9  =000000000000      K
 DATA10=000000000000       K
 DATA11=              0    F
 DATA12=              0    I
 DATA13=              0    F
 DATA14=              0    I
 DATA15=000000000000       K
 DATA16=000000000000       K
----------------------------
HIT '.' KEY TØ TURN PAGE.
HIT END KEY WHEN DØNE.
```

Figure 11. **SETPAR** Subroutine Display.

Because of the very general nature of graphic information, no attempt was made to assist the programmer in formating his display. To increase the flexibility of the I/O utility subroutine designed for this task (PEN2) an option was added which allows the graphic console operator to terminate the display by depressing either an alphanumeric key or a program control button. In any case, however, the calling program is fully informed as to the condition which terminated the display.

Figure 12 shows the source language statement necessary to execute PEN2. The 2nd and 3rd arguments define the location and size of the array containing the graphical information. The 4th argument describes the condition which terminated the display. The 1st argument provides information pertaining to the pencil.

Upon execution of PEN2, a display of the specified information begins. At this point, the graphic console operator can depress an alphanumeric key, a program control button, or bring the position indicating pencil into contact with the display CRT. If a key or button is depressed, the display is immediately terminated and control passes back to the calling program. If the pencil is touched to the screen, a small cross appears superimposed on the display. This cross defines the position of the pencil relative to the display and provides the operator with visual feedback. The visual feedback is particularly important here because of severe parallax problems. If the pencil is moved across the screen, the cross will follow. When the pencil leaves the screen, the display is terminated and the last location of the cross is returned to the calling program. Figure 13 shows the pencil being used to point to a region on a grid.

```
EXFCUTE PEN2.(PEN,ARRAY,SIZE,RETURN)
```

Figure 12. Source Language Statements for PEN2 Subroutine.

This subroutine has proven to be very flexible and easy to use, resulting in a wide variety of applications. One such application of PEN2 was in the development of a higher level I/O utility subroutine permitting the entry of decision type information.

Decision Information Input

The task associated with the entry of decision information is to present the graphic console operator with a set of possible actions available to the controlling program and to allow him to select an ordered subset of these actions.

The I/O subroutine CHOICE was developed for this task and requires three items of information from the calling program: (1) the number of alternatives to be presented, (2) the symbolic name (i.e., location in core) of an

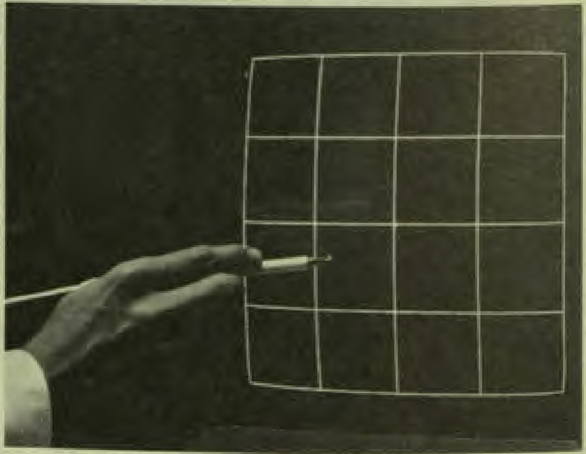


Figure 13. Positional Input Via the PEN2 Subroutine.

array containing the BCD messages describing the alternatives, and (3) the symbolic name of an array provided for the list of selected alternatives (see Figure 14).

The execution of this subroutine begins with a display of the alternatives, as shown in Figure 15.

The operator may now select his first alternative by pointing to the appropriate field with the voltage pencil and then removing the pencil from the screen. Immediate visual verification of the choice is provided by the digit 1 at the right hand side of the selected field as shown in Figure 16.

The operator proceeds in this manner to complete his selection of the desired alternatives. When he is done, he selects the field labeled "DONE" and control is returned to the calling program. If the operator makes a mistake or changes his mind, he can "erase" his last selection by selecting the field labeled BACKSPACE.

SOURCE LANGUAGE STATEMENTS

In the first section of this paper, five subroutines were described which are representative of a class of subroutines whose primary objective is to facilitate the on-line input of various types of information. In the following section, we describe three source language statements which provide the programmer with the capability of generating various types of on-line output in an easy, efficient, and flexible manner. In the development of these statements, an effort was made to conform (insofar as possible) to the precedents established by the normal I/O source statements appearing in the NOMAD language. It was felt that, in so doing, the resulting statements would be much easier for the programmer to use.

Display Format Statement

"DISPLAY FORMAT" was designed to permit the programmer to use the graphic console display CRT in much the same way as the on-line printer was used in "the good old days." The implementation of this "on-line print" capability provides the programmer with a very convenient way of presenting a wide variety of information to the graphic console operator.

The maximum amount of information which can be displayed by DISPLAY FORMAT (i.e., its unit record) is 480 characters. These 480 available character positions are in the form of 20 lines on the CRT with 24 character positions per line.

```
EXECUTE CHOICE.(ALTERS.5.SELECT)
```

Figure 14. Source Language Statement for CHOICE Subroutine.

```
THIS SPACE IS
ALLOTED TO THE
FIRST ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
SECOND ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
THIRD ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
FORTH ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
FIFTH ALTERNATIVE
--------------------------------
DONE            BACK SPACE
```

Figure 15. CHOICE Subroutine Display.

```
THIS SPACE IS
ALLOTED TO THE
FIRST ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
SECOND ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE                1
THIRD ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
FORTH ALTERNATIVE
--------------------
THIS SPACE IS
ALLOTED TO THE
FIFTH ALTERNATIVE
--------------------------------
DONE            BACK SPACE
```

Figure 16. CHOICE Subroutine Display.

The DISPLAY FORMAT statement has a list and a format associated with it. The list gives the values and items to be displayed and the format gives the display pattern to be used. The format is composed of a sequence of format specifications of the form *Cw*. The character *C* is the control character and defines the type of operation or conversion to be performed. In general, *w* refers to the number of character positions associated with the operation defined by *C*. If, in any format specification, the field width *w* is zero, that format specification and its associated list item (if any) are ignored.

There are three types of format specifications available for the conversion of numeric data. Octal and decimal integers are generated by using the *Kw* and *Iw* specifications respectively. The *Fw* specification provides for floating point variables which will be displayed as either floating point decimals (e.g., .2E–3) or fixed point decimals (e.g., .0002) depending on which form produces the fewest characters for display.

There are two types of format specifications available for the conversion of BCD information. The *Aw* specification is identical with that used in other NOMAD I/O statements. In particular, one word (six characters) will be picked up from the list and right justified or truncated on the right, depending on whether $w \geq 6$ or $w < 6$. The *Hw* specification differs from ordinary I/O statement usage in that the Hollerith information is specified in the list rather than in the format.

The specification *Xw* provides *w* blanks in the display. The specification */w* results in a downspace of *w lines.* An automatic downspace of one line results when the number of characters on a line reaches 24. The specification *Pw* (*w* has no particular significance in this case unless it is O) resets the current character position to the first character position of the first line. This specification has proven very useful when putting out columns of information which comes from several different arrays in memory.

The specification *Rw* causes the succeeding portion of the format to be repeated, if necessary, until the list is exhausted (*w*, again, has no particular significance unless it is O). If the specification *Rw* does not appear anywhere in the format, the entire format will be repeated until the list is exhausted.

The form of the "DISPLAY FORMAT" statement and a sample format is shown in Figure 17.

Perhaps the most radical departure of the DISPLAY FORMAT statement from standard NOMAD I/O statements involves the use of its second "argument" (i.e., the item called INOUT in Figure 17).

INOUT will, in general, be the first cell of a short array in which the program using DISPLAY FORMAT can:

a) specify the status light configuration during and upon termination of the display,
b) specify the input devices which the graphic console operator will be permitted to use in terminating the display.
c) receive a description of the condition which caused the termination of the display.

The graphic console display shown in Figure 18 corresponds to the DISPLAY FORMAT statement of Figure 17.

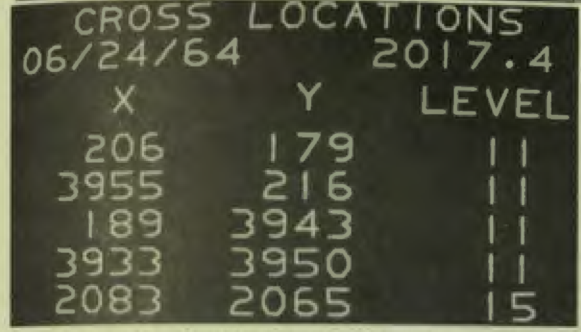
Because of its flexibility and ease of use, the DISPLAY FORMAT statement has been a valuable aid in the development of a cooperative man-machine problem-solving system.

Record Format Statement

The RECORD FORMAT statement provides a means of producing hard copy alphanumeric output through the use of the recording feature of the image processor. The form of the RECORD FORMAT statement is identical to that of DISPLAY FORMAT (i.e., RECORD FORMAT FMT, INOUT, L1, L2, L3, . . .). RECORD FORMAT recognizes all of the format specifica-

```
 VECTOR VALUES L1 = $CROSS LOCATIONS$

 VECTOR VALUES X = $X$

 VECTOR VALUES Y = $Y$

 VECTOR VALUES LEVEL = $LEVEL$

 VECTOR VALUES FMT = $(X5,H15,/1,X3,H8,X4,F6,/1,
1 X7,H1,X5,H1,X3,H5,R5,/1,X5,I4,X2,I4,X4,I2)$

 DISPLAY FORMAT FMT,INOUT,L1,DATE,TIME,X,Y,LEVEL,
1 DATA(1)...DATA(15)
```

Figure 17. Source Language Statements for DISPLAY FORMAT.



```
   CROSS LOCATIONS
06/24/64        2017.4
    X        Y     LEVEL
   206      179       11
  3955      216       11
   189     3943       11
  3933     3950       11
  2083     2065       15
```

Figure 18. Application of DISPLAY FORMAT.

tions and conventions used by DISPLAY FORMAT.

The unit record for RECORD FORMAT is 12,000 characters per film frame. These 12,000 available character positions are in the form of 100 lines of 120 character positions each.

With RECORD FORMAT, INOUT is a single cell and is used by the calling program to specify the desired film train operations associated with the recording. Appropriate bytes in INOUT define which film train is to be used, how many frames are to be advanced before and after the recording is performed, and when developing of the film frame is to occur (if at all).

RECORD FORMAT enables the programmer to use the image processor as he would use an off line printer with the added advantage that his turnaround time is measured in minutes rather than hours. The output can be developed immediately and moved to the project station of the image processor for on-the-spot review and then placed on an aperture card for hard copy reproduction. The hard copy output is very similar to that produced by an IBM 1403 printer.

Generate Format Statement

The DISPLAY FORMAT and RECORD FORMAT statements enable the programmer to easily produce alphanumeric output. However, it is frequently necessary to produce either displays or recordings which contain a combination of graphic data and alphanumeric characters of different sizes (see Figure 19). In addition, it is desirable to have the capability of altering a small part of a display without having to regenerate the whole display. The value of such a capability is best illustrated in the operation of the CHOICE subroutine (see Figures 15 and 16). The GENERATE FORMAT statement was specifically designed to meet these needs.

With respect to formats, input lists, and statement form (i.e., GENERATE FORMAT FMT, INOUT, L1, L2, L3, . . .), GENERATE FORMAT is identical to the previous two statements. However, instead of producing output directly (as is the case with DISPLAY FORMAT and RECORD FORMAT), GENERATE FORMAT provides the programmer with the array of CRT coordinates needed to produce a display of the alphanumeric information. In addition, GENERATE FORMAT enables the programmer to specify character size and line spacing (i.e., the user can define his own unit record).

By executing GENERATE FORMAT several times with different unit record specifications and combining the resultant arrays of CRT coordinates with an array of coordinates representing graphic data, the programmer can, through the use of a basic display or recording subroutine, produce output of the type shown in Figure 19.

The special features of GENERATE FORMAT were utilized in the I/O utility subroutines SETDA, SETPAR, and CHOICE to quickly and efficiently produce sequences of displays in which only portions of the basic display are altered. For instance, Figure 16 can

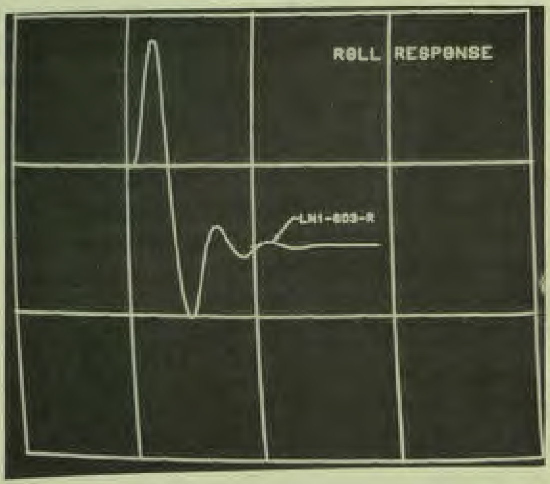


Figure 19. Application of GENERATE FORMAT.

be produced from Figure 15 simply by adding the three coordinate pairs necessary to generate the character 1 to the large coordinate array for the basic display.

The DISPLAY FORMAT and RECORD FORMAT statements provide the programmer with an easy and efficient method of utilizing the output capabilities of the 7960 system. The GENERATE FORMAT statement rounds out the output software package by providing the additional flexibility needed to handle the situations described above.

SUMMARY

Any attempt to implement a system which stresses cooperative man-machine interaction must concern itself primarily with the problem of man-machine communication. Furthermore, the man-machine communication will typically take place in a wide variety of situations and at many levels of problem discourse. Attempts to define all possible communication situations and develop specific programs for each will lead to a tremendous amount of redundancy and duplication of effort. It is possible to avoid this problem by realizing that all man-machine communication must pass through the I/O hardware interface. Thus, by providing an extensive, flexible, and powerful I/O software capability, the presence of the hardware interface need not concern the programmer and he can devote his whole attention to the more significant aspects of man-machine communication. The software described in the body of this paper has proven to be an immensely powerful tool which has enabled programmers to rapidly design and evaluate the wide variety of communication techniques necessary to any man-machine problem-solving system.

ACKNOWLEDGEMENTS

The material described in this paper is not solely the work of the authors. Significant contributions were made by other members of the programming staff. In particular, we would like to acknowledge the contributions of C. Richard Lewis of the Computer Technology Department, GM Research Laboratories, and Martin Weinrich. *

REFERENCES

1. JACKS, E. L., "A Laboratory for the Study of Graphical Man-Machine Communication," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
2. COLE, M. P., DORN, P. H., LEWIS, C. R., "Operational Software in a Disk Oriented System," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
3. HARGREAVES, B., JOYCE, J. D., COLE, G. L., FOSS, E. D., GRAY, R. G., SHARP, E. M., SIPPEL, R. J., SPELLMAN, T. M., THORPE, R. A., "Image Processing Hardware for a Man-Machine Graphic Communication System," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
4. KRULL, F. N., FOOTE, J. E., "A Line Scanning System Controlled from an On-Line Console," *Proceedings Fall Joint Computer Conference*, San Francisco, California, Oct. 1964, (this volume).
5. FREEMAN, H., "On the Encoding of Arbitrary Geometric Configurations," IRE Transactions on Electronic Computers, June, 1961.

---

* Now at the University of Michigan.

A LINE SCANNING SYSTEM CONTROLLED FROM AN ON-LINE CONSOLE

Fred N. Krull and James E. Foote
Research Laboratories, General Motors Corporation, Warren, Michigan

INTRODUCTION

Direct graphical input is one of the newest and most exciting sources of digital computer input. Programming techniques and hardware are beginning to appear which are designed to automatically process graphic information.[1, 2, 3, 4] This paper describes an experimental system which has been designed to facilitate the digitizing of line images. The equipment which is used for this purpose is an IBM 7960 Special Image Processing System, consisting of graphic console, image processor, and a modified data channel.[7] The image processor (Figure 1) contains a programmable Cathode Ray Tube (CRT) scanner, a CRT recorder, a 35mm camera, and film processing equipment.

The graphic console (Figure 2) contains devices to communicate with the operator and to control the image processor.

Both of these devices are tied directly to a computer via a modified data channel.

The principal objective of this portion of DAC–I (Design Augmented by Computers) project[5, 6, 7, 8] was to utilize the full capabilities of the image processor, graphic console, and digital computer to digitize a variety of line images to a high degree of accuracy. Program system objectives dictated that line widths .05% of image size (.01 inches on 20x20 inch document) must be detectable to an over-all system accuracy of ±.05% of image size. Thus, this system was designed to provide a rapid, versatile, and accurate method for converting graphical data to digital form. While it provides only for the digitizing of lines, a natural outgrowth of the work will be a library of pattern and character processors to be used by programmers in much the same manner as card and tape input/output (I/O) routines are used now.

The application requirements of the DAC–I project dictated that one of two approaches be used for the analysis and processing of a two dimensional graphical form:

a) Algorithms are specified which prescribe the rules for analyzing a graphical image. This approach assumes a structure for each characteristic image, and relies upon the availability of individual pattern processors to detect image components.

b) No fixed structure is assumed for the input document. Minimal restrictions are placed on image quality only. Emphasis is placed upon providing elementary functions which an operator may call upon and combine in order to process a complex form. All decision capability and selection of functions is left to the operator who is controlling the scanning device from an on-line console.

The second approach was selected since at this point in the project it was not possible to specify the structure of the large variety of input documents anticipated. We felt, for instance, that the system would be used to digitize

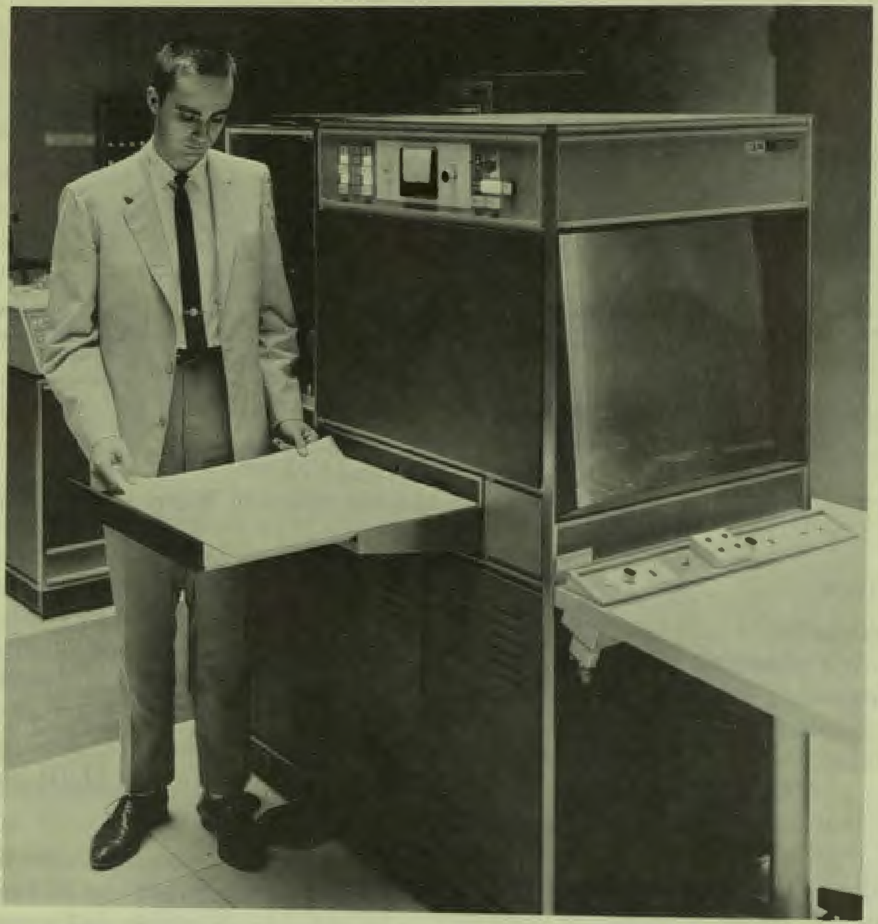


Figure 1. Image Processor.

documents ranging from mathematical graphs to engineering drawings. Therefore, it seemed advisable to concentrate our efforts on providing a reliable set of basic pattern analyzers. These elementary functions would then provide the basis for processing much more complex forms.

Since the console operator now becomes an essential part of the scanning sequence, it was necessary to devote considerable effort to the problem of man-machine communication. Development of the system has permitted us to carry out a series of experiments, designed to discover the best interface between man and computer. Scanning functions have been made automatic when practical, but the decision capability of the human operator is still used to best advantage. Thus, flow diagrams for scanning functions in many instances contain a block in which the operator is controlling the selection of a branch or setting the value of a parameter. Each situation has warranted an investigation to answer the question: "Who can perform the job better—man or machine?"

An essential part of this process was to economically match the speed of the computer with the speed of man. The solution was to multiprogram a 32K digital computer being used to process regular compilations and executions under batch monitor control.[5, 6] A 16K–16K logical core split was made available for use by the on-line operation and the batch monitor programs *. The on-line console operation has millisecond access to the CPU of the computer on a demand basis. When use of the CPU is no longer required, control is returned to the batch monitor program.

---

* This configuration has been updated to a 64K memory with a 32K–32K logical core split.

A dynamic storage allocation system and execution processor[6] occupy almost half of the 16K memory cells devoted to on-line console operation. The remainder of the core space (approximately 8K) is available for use by the scanning system programs and data tables. The scanning programs total more than 40K cells, exclusive of memory data tables. Hence, the dynamic storage allocation scheme is utilized to overlay subroutines as they are required. The entire scanning system was written in an algebraic language called NOMAD and a channel language called MAYBE.[6]

All functions are made available to the operator via program control buttons on the graphic console. There is almost a one-to-one correspondence between operational capability and program control buttons. The organization of this type of system is illustrated in schematic form in Figure 8.

This concept allows the operator to depress a button and execute a wide range of functions. Upon completion, the system may return to standby and wait for the next selection, or each function may automatically call on other functions. For example, an operator may depress the DISPLAY button to obtain a graphical image on the graphic console CRT. Similarly, a scanning function may display intermediate results by *logically* depressing the DISPLAY button. Thus, one function may logically depress other program control buttons. In addition, while one function is being performed, it may be advantageous to make other functions optionally available. Thus, function (B) can allow program control buttons (A) or (C) to be depressed, and subsequently signal the control program which option has been selected.

The functions themselves logically fall into four distinct areas: Film Operations, Scanning Operations, Display and Review Operations, and Modify and Store Operations. Each area presented the same problem; namely, how to best communicate with the man in order to perform a specific operation. The variety of solutions that have been employed are discussed in the remainder of this paper.

FILM OPERATIONS

A line scanning problem will, in general, require more than one film frame (i.e., more than one input document) and the user may wish to use either the exposure and rapid processing facilities of the image processor or off-line film

Figure 2. Graphic Console.

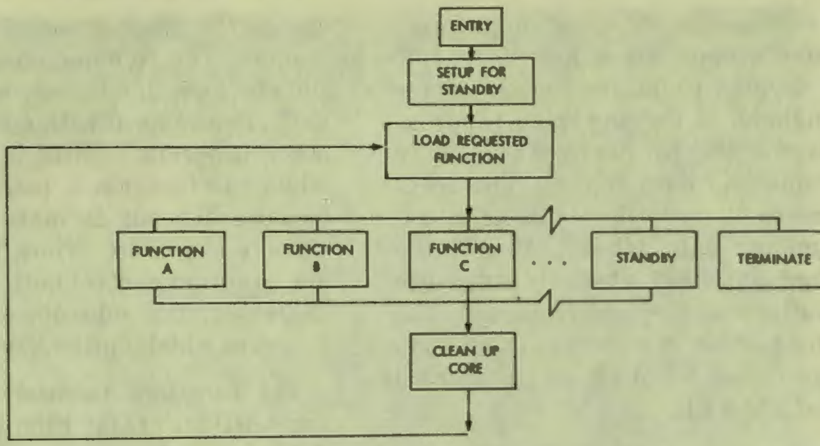


Figure 3. System Organization.

which has been pre-exposed and may or may not have been processed. During the course of the scanning operation, the user may at appropriate times wish to record current results on film either for purposes of verification or to form a permanent record of his work.

It was therefore necessary to include in this line scanning system a facility for the console control of both of the film trains [7] in the image processor which is shown schematically in Figure 4.

One film train (train B) may be used only for recording and reviewing output. The other film train (train A) may be used either for recording or for exposing input documents onto film and scanning the resulting film images. Both film trains can process an exposed film frame in approximately 30 seconds and project the image onto the 22x20 inch viewing screen.

The 22x22 inch paper input station can accept documents for exposure onto raw film in train A. Alternatively, pre-exposed microfilm may be inserted in the supply cassette. Film transport commands allow the film to be advanced into buffer 1, developed, and advanced or backspaced into buffer 2 under program control.

The console operator is provided with several film functions which make use of these facilities. An exposure processing function readies the paper input station to accept documents. The operator may then insert documents and make exposures by depressing an EXPOSE button on the side of the image processor. Each exposure is automatically advanced into buffer 1. A count of the number of exposures is displayed on the graphic console screen. When all exposures have been completed, the operator

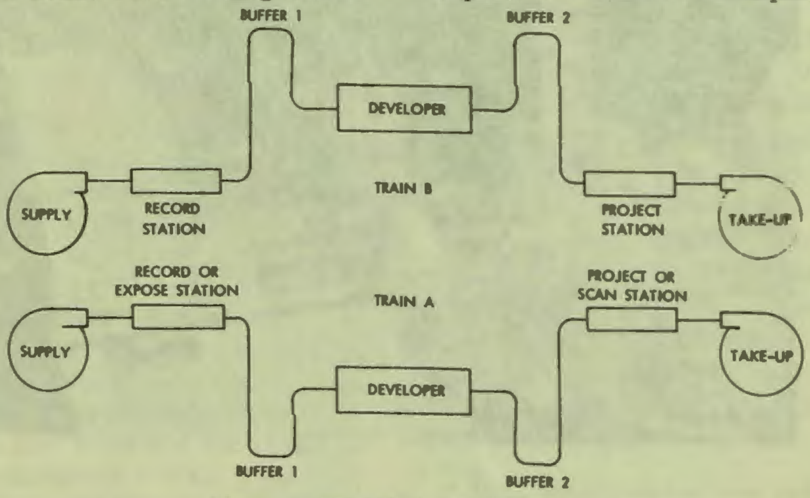


Figure 4. Film Train Configuration.

may initiate the processing cycle. The film transport is programmed to develop all of the exposed film in buffer 1 after adding a trailer to the exposures, and position the first exposure at the scan-project station. The operator may utilize manual controls on the image processor to advance or backspace frames that are in buffer 2 or on the take-up reel. Operational experience has shown the value of providing the console operator with a rapid means of preparing a film image suitable for scanning. A certain amount of flexibility and film quality is lost through the utilization of rapid processing, but this seems to be more than offset by the convenience of short turn-around time.

An auxiliary film processing function is available if off-line film is to be used. Utilization of pre-exposed film allows for a wider variety of input quality and document sizes. This film is inserted in the supply cassette and spliced to the film in train A. When the processing is initiated, both film buffers are emptied (i.e., the film is pulled tight). They remain empty as the film passes from the supply cassette, through the developer and then past the scan-project station. The graphic console operator can monitor the operation on the projection screen. Depressing the END key on the alphanumeric keyboard will halt the developing. In this manner rapid processing film which has been pre-exposed can be developed. Standard microfilm or rapid processing film that has been preprocessed is not significantly affected by passing through the developer during this operation. The use of preprocessed high contrast microfilm permits the scanning of material which is of much poorer quality than can be accepted at the paper input station.

During the course of various scan operations, it is frequently necessary to make a permanent record of the results. Selection of a recording function causes all current scan results to be recorded on film train B. The material recorded on film is the same as that which is shown on the graphic console by the DISPLAY function discussed later in the paper. After recording is completed, the frame will be automatically advanced into the first film buffer of train B. The developing process will be initiated automatically after six recordings but can be initiated at any time via a FILM CONTROL function. By means of manual controls on the image processor, a processed frame can be centered on the viewing screen. By appropriate manipulation of both film train projection lamp rheostats and positional controls (2 dimensional motion is possible on train B), both the scan results on train B and the original document on train A can be superimposed on the viewing screen for purposes of comparison.

The user also has at his disposal the ability to develop or clear film which has been moved into any of the film buffers. Clearing the film moves all exposures onto the take-up reel from which they can be removed and mounted in aperture cards. One of the accessory pieces of equipment available to the user is an aperture card printer. Thus, hard copy output is available to the user in a matter of minutes after he leaves the console. We find that this is particularly valuable in evaluating results and maintaining a record of work accomplished.

SCAN OPERATIONS

The unique feature of this line scanning system is that only those elements of an image which are selected by the console operator will be scanned and digitized. The basic element of an image is a line segment defined as a continuous curve terminated by two ends, two junctions or a combination of both. Since complicated images may contain many intersecting curves, a line will, in general, consist of several segments which must be added together logically. Experience has shown the junction to be a far more accurate delimiter than the end of a curve. The normal procedure of this system has been, therefore, to define the end points of lines precisely by means of perpendicular slash marks. Figure 5 shows a typical document from which line AD is to be digitized. Line AD consists of segments AB, BC, and CD. A line may, of course, consist of only one segment.

The requirement of digitizing selected lines which represent only a small fraction of the entire image suggested a line tracking technique. Analysis of a raster scan of the entire image area was deemed impractical because of the volume of data involved.

The automatic line tracking procedure which forms the heart of this line scanning system is described briefly below. Figure 6 shows a typi-

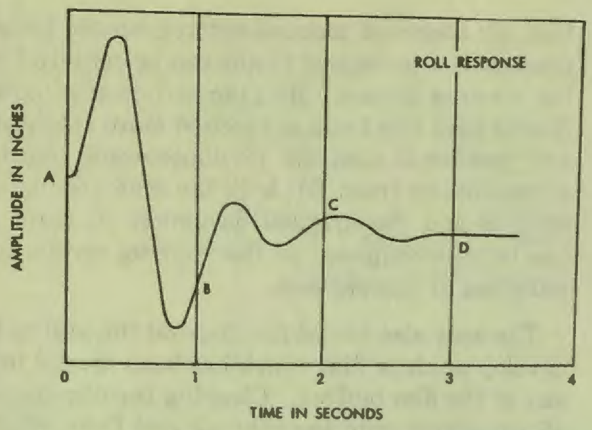


Figure 5. Sample Document.

cal line with two points (P_A and P_B) that have been sampled. The next point is estimated to be P'_C : a distance (ΔH) away from P_B on the line defined by points P_A and P_B. Two "scan feeler vectors" (P_1–P_2) and (P_3–P_4) are then plotted on the CRT parallel to and a distance (ΔV) away from the line (P_B–P'_C). ΔH and ΔV are specified by the console operator.

If nothing is encountered by either feeler vector, a scan vector (P_2–P_4) is plotted and the next point P_C is determined by the intersection of (P_2–P_4) with the line as shown in Figure 7.

The procedure is repeated until the line ends or one of the feeler scan vectors gets a "hit" indicating that either a junction has been encountered, or that the tracking step size (ΔH) and the feeler vector spacing (ΔV) are not compatible with the curvature of the line. When this occurs, a block of code will be called upon to analyze the situation, and action will be requested from the operator if necessary.

At each point, the threshold level (detection sensitivity of the scanner) is adjusted within rigid bounds, based on results at the last point, to the minimum necessary for line detection. When a new point is obtained, the threshold level bounds are modified if necessary. In this manner, large variations in density across a film frame can be accommodated while still preventing erroneous scanner responses due to improper threshold level. The threshold level bounds, along with tracking step size, feeler vector spacing and several other scanning parameters can be modified if necessary by the user at the console via a CHANGE PARAMETERS function.

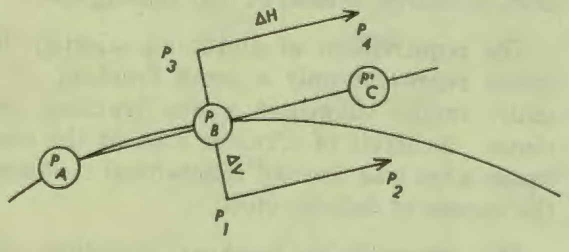


Figure 6. Line Tracking Procedure.

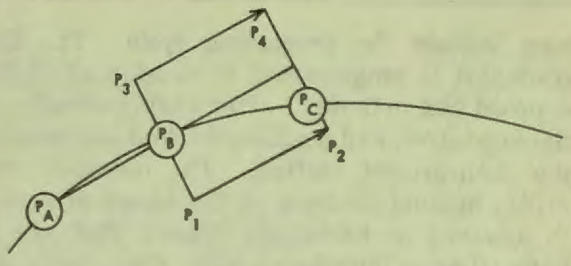


Figure 7. Line Tracking Procedure.

The automatic method of line tracking is utilized by the user whenever possible. If areas of difficulty occur, the user may request or the scan program will automatically generate a call for a raster sweep (TV scan) of the area of difficulty. The user may then utilize the sweep display on the graphic console screen for diagnostic purposes.

The graphic console operator is required to combine those functions which will enable him to process a complicated image. It is assumed that a film frame has been centered in the scan gate before any scan functions are selected. A RESTART function readies the system to accept results from a new image. All previous scan results are deleted.

A REGISTRATION function may be used to search for a border around the image corresponding to a 20x20 inch square border on the input document. The main function of this border is to provide an easy means of supplying the scanning system with coordinate and scaling data. The coordinates of the border are assumed to be (0,0), (0,20), (20,20), (20,0) but can be modified by the console operator by using a CHANGE PARAMETERS function. If registration is successful, the border, plus lines dividing each side of the border into quarters, will be displayed on the graphic console CRT. This grid aids the console operator in using the position indicating pencil to select lines for scanning. It is not necessary to register, however, in order to proceed with scanning since there are alternate methods for the system to obtain coordinate and scaling data.

A SCAN A LINE function readies the system to begin scanning the first segment of a new line. The user will immediately be requested (via appropriate comment on the graphic console screen) to select by means of a position-indicating pencil the approximate area for the scanning to begin. The user may then point directly to an area on the screen (using the display of previous results and the registration grid as a guide, if available). Alternatively, the console operator may request a gross raster scan and display of the entire image. He may then select a line with the pencil as shown in Figure 8.

Once a starting area has been supplied, a search will begin for two points on the first segment. These points will then be used to initiate the automatic line tracking procedure which will track to both ends of the segment in two steps. If any difficulty is encountered during this operation, the results up to that point along with an appropriate comment and a box indicating the region of trouble will be displayed. The operator must then take the proper action which usually begins with a TV sweep of the problem area. After a portion of a line has been scanned, an ADD A SEGMENT function may be used to add segments to the given line. Selecting this function (active only after the first segment of the current line has been scanned) readies the system to scan a segment *adjacent* to one end of the current line and add it to the current line.

The user is immediately requested to indicate with the position-indicating pencil the approximate location and initial slope of the next segment. A search will then be initiated for a point on the next segment which is separated from the endpoint of the current line by a distance equal to the tracking step size ($\triangle$H). This point and the endpoint will then be used to initiate line tracking which will proceed to the end of the segment. If any difficulty is encountered during this operation, all previous scan results, plus an appropriate comment and a box indicating the problem area, will be displayed. The user must then take appropriate action. If this entire operation is successful, the segment will be added logically to the current line to form the new current line.

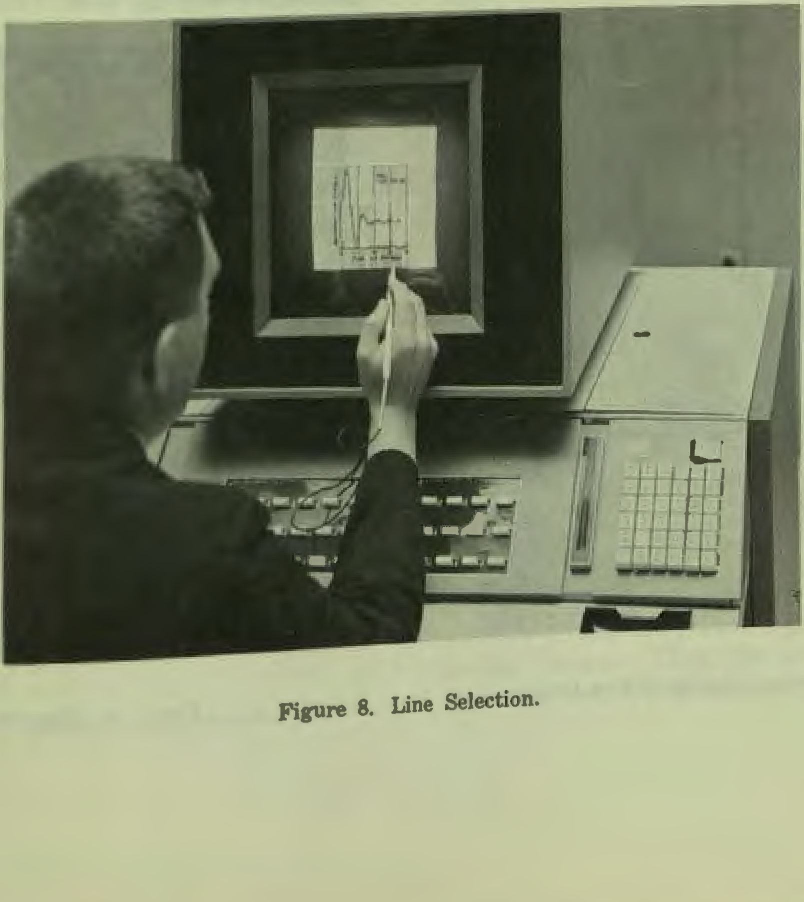


Figure 8. Line Selection.

A TV SWEEP function allows the user to examine a localized area for diagnostic or corrective purposes. This function will be initiated automatically whenever the tracking procedure encounters difficulty. The console operator may also use this function to investigate any data point along a line by selecting the point of interest with the position-indicating pencil. The scan responses are analyzed and a display is generated which nominally fills the entire graphic console screen. This enables the user to observe a magnified view of a localized area on the image (as it appears to the scanner). Figure 9 shows a TV sweep display of a typical junction. The size of the sweep area is approximately .5 x .5 inches on a 20x20 inch document.

The "X" indicates the center of the scan raster. The user may adjust the threshold level or change the location of the scan raster via appropriate alphanumeric keys. The raster may also be moved by indicating a desired location with the pencil. A new raster scan and display will then be generated. The system understands that the position-indicating pencil position superimposed over the TV sweep display refers to the corresponding position in the actual scan raster. Other appropriate system operations may be initiated from this mode at any time.

If a diagnostic study reveals that automatic line tracking is impractical for some reason, coordinate points may be stored in conjunction with the TV sweep facility. This function allows the user to add the current center of the scan raster as a data point and thereby move manually over an area of difficulty before resuming automatic tracking. Under extraordinary circumstances (e.g., very poor film, poor document quality, or an extremely congested image), an entire line can be digitized in this manner.

The CHANGE PARAMETERS function allows the user, through a combination of multiple choice and alphanumeric keyboard responses, to change many of the scan parameters. The value of each parameter is displayed on the graphic console CRT along with suitable descriptions (Figure 10). Through the use of a multiple choice sequence employing the position-indicating pencil, the user may select a parameter and type in a new value for that parameter. This function allows the user to materially change the performance of a subroutine from the console. These parameters are used within the programs both as numeric constants and as switches for branching operations.

DISPLAY AND REVIEW OPERATIONS

The graphic console 10x10 inch CRT is the principal medium by which the computer communicates with the user. Throughout the sys-

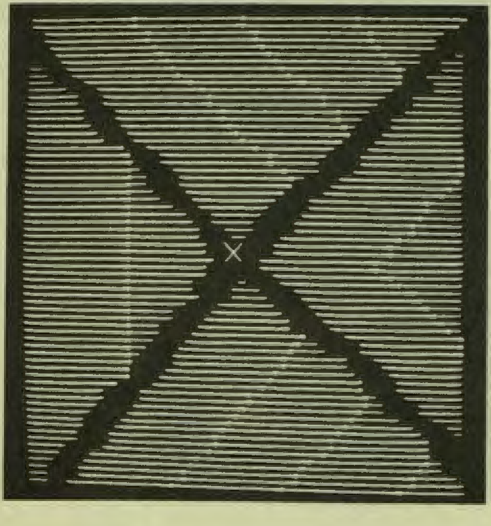
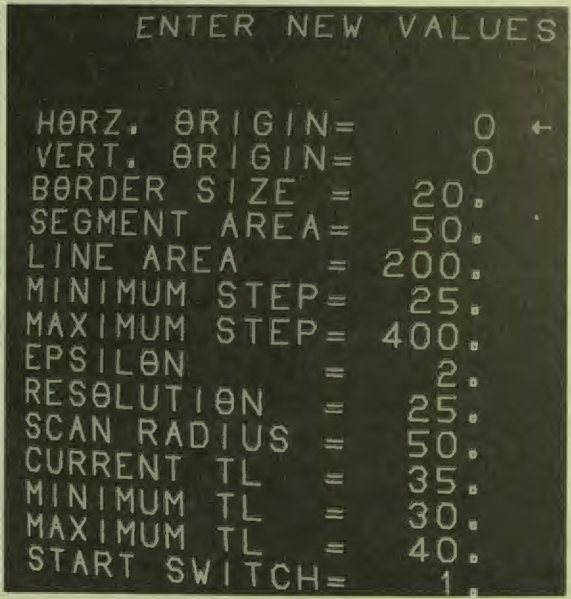


Figure 9. Junction of Two Lines.



```
      ENTER NEW VALUES

HØRZ. ØRIGIN=      0 ←
VERT. ØRIGIN=      0
BØRDER SIZE  =   20.
SEGMENT AREA=    50.
LINE AREA    =  200.
MINIMUM STEP=    25.
MAXIMUM STEP=   400.
EPSILØN      =    2.
RESØLUTIØN   =   25.
SCAN RADIUS  =   50.
CURRENT TL   =   35.
MINIMUM TL   =   30.
MAXIMUM TL   =   40.
START SWITCH=     1.
```

Figure 10. Parameter List.

tem, messages are continually being displayed on the CRT to which the user must respond. Although this display is low in accuracy with respect to the scanner or recorder, it is also utilized for the purpose of displaying scanned results. A display of the scan results may be used to check for completeness and to monitor the progress of the scanner. To some degree, the accuracy of the scanned results may also be checked at the graphic console.

As a film image is scanned, the digitized coordinate points are stored in an in-memory table. The structure of this table is quite simple. Each element of scanned data is stored as a linear array, with suitable identifiers and pointers to the next element of data. The scan operations may at any point in their logical sequence call upon the display operation to generate a pictorial representation of the in-memory scan data. A typical display is shown in Figure 11.

If an error condition has occurred during some previous function, the display operation may be so signalled. This will result in a small box being superimposed over the location of the error, plus addition of a suitable error comment (Figure 12).

After the scan results appear on the graphic console CRT, the following review operations become available.

a) Increase Scale
b) Change Mode
c) Identify Dimensions
d) Identify Coordinates

By selecting the INCREASE SCALE function, the size of the display will be doubled.

Figure 11. Display of Scan Results.

SEGMENT COULD NOT BE LOCATED

Figure 12. Error in Scanning.

Any scan data which falls outside of the field of the CRT is deleted by appropriate programming. Reselection of a display returns the scan results to their original scale. The CHANGE MODE function may be used to display all lines either as a continuous curve or as a series of crosses. The latter mode allows the user to ascertain the number and location of each single digitized point. The ability to view individual scanned points is essential when the scanner is being used to capture a minute feature of a line.

If some portion of the display outside the field of view is desired, the pencil may be used to point to the edge of the field. Removing the pencil from the screen will cause the scanned data to be redisplayed with the selected point relocated at the center of the field of view. Figure 13 illustrates a display in which the user has reviewed his scanned results by centering the field on the end of a line and magnified the scale four times to examine an end condition in detail.

Within the operation of the scaling and field centering mechanism can be found a lesson in man-machine interaction. A variety of functions can be selected subsequent to a display. Typical of these is the LINE SCAN function. After this function is selected, the user is requested (by comment on the graphic console screen) to use the pencil to point to a location on the screen. Thus, the sequence of operations was to select a function (by depressing a

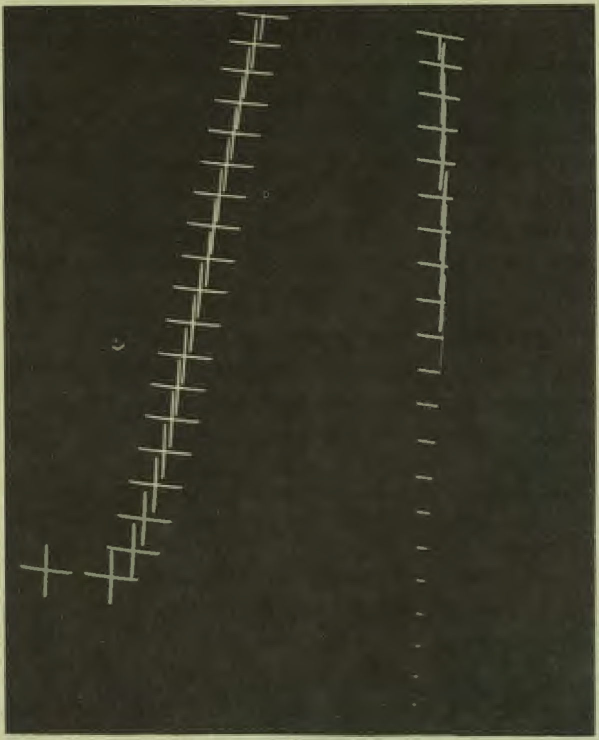


Figure 13. Magnified Display.

program control button) and *then* point. Our initial operational experience taught us that the more natural instinct was to point and then select an operation. Similar instances of man-machine interaction being uncomfortable were noticed throughout the development of the system. In most cases, the problem was rectified after user and programmer discussed alternative sequences of operation. Thus, many operations communicate with the man in more than one way, thereby anticipating the variety of ways in which a man may respond to a given situation.

While in review mode, the user is also frequently concerned with the accuracy of his results. The system provides two means for reviewing the accuracy of the scan data before storing the data on disk. If a square border has been registered from the film image (1.09x1.09 inches on film), all interior points are related to the coordinates of the corners of the border through the use of a four point perspective transformation. At the paper input station, the border would normally appear 20x20 inches square. The user through the use of the CHANGE PARAMETERS function may define the coordinates of the corners of the border. Thus, the user may select any point interior to the border and receive the inch coordinates of that point. The point may be selected by pointing with the position-indicating pencil or by using linear arrows on the alphanumeric keyboard to vernier a pointer.

If no border registration is available, the only accuracy test which can be made is to check the chordal distance between two points. For example, if the scanner were used to digitize a sine wave consisting of two cycles, the user could measure the chordal distance in inches between the ends of the cycles on the original document. An IDENTIFY DIMENSION function allows the user to use the pencil to point to *two* locations on a line and receive a display of the chordal distance between the two points. In this case, the program searches for the two points on the scanned line nearest to the two locations selected by the user. The program displays the distance assuming an 18.33 reduction ratio between document and film image.

MODIFY AND STORE OPERATIONS

The development of an elementary set of modify operations was a direct result of monitoring the use to which various users put the system. The most frequent request was for operations to delete erroneous data. Here again let us emphasize that the scanner and computer are utilized to perform the elementary operations while the user (through graphic console displays) retains judgment as to the correctness of the results.

A typical situation might be that the scanner began to digitize a scratch on the film rather than a line on the film image. The requirement for data deletion operations led to three methods for deleting data. The console operator may select a DELETE LAST SEGMENT function and cause the *last* element of scanned data to be deleted from in-memory storage. An entire line or a single point may be deleted by pointing to a location on the graphic console screen with the pencil and selecting the desired operation. In these two cases, the item of scanned data, be it a line or point, nearest to the selected location is deleted from in-memory storage.

Another frequently requested operation was the ability to add intermediate points to a display of scan results. Because of its finite sampling capability, the scanner may miss a particularly critical point on a line. In these cases, the user may point at the display with the pencil and cause a coordinate point to be inserted in the scanned results at that point. A high degree of accuracy may be obtained by adding coordinates in this manner, particularly if the scale of the image is enlarged before any operation is attempted.

After all scanning, review, and modify operations have been completed, the user can choose to save these results by storing them on a random access disk file. For our purposes, each digitized line is assigned a unique data name (e.g., LN5–1–537–R) which determines its storage location on disk.[7] Any other application-oriented program may have access to this data by referring to the same data name. Library I/O subroutines are available to the programmer for storing and retrieving data from the disk.

The function which stores data on disk provides the facility for transforming the digitized points to any desired coordinate system. The manner in which this is accomplished is for the user to provide the *desired* coordinates of two of the scanned points (usually the left and right end point). This data enables the STORE function to compute a linear transformation between the raster unit coordinate system of the scanner and the desired output coordinate system.

Since the STORE function requires alphanumeric input, we were interested in the variety of ways in which a man could communicate this data. The devices which could be made available for the transmission of alphanumeric data are as follows:

a) Alphanumeric keyboard used as a typewriter
b) On-line card reader
c) Scan and recognition of characters in the field of the image
d) Writing with the position-indicating pencil on the face of the graphic console CRT
e) Multiple choice operations by pointing with the position-indicating pencil on the face of the graphic console CRT

The options which are currently available are (a), (b), and (e). Our experience leads us to believe that (d) is not practical, simply because of the ease of using a typewriter and monitoring the message on the face of a CRT. Scanning and recognition of character sets on the film image is obviously very desirable. This work is under current development but will not be reported on in this paper.

When a line or series of lines have been scanned, reviewed, and modified, the console operator may elect to save the results by choosing the STORE function. At this point, all scanned data is corrected according to a table of calibration data stored on disk each morning by maintenance personnel. The calibration data gives a measure of the distortions in the scan raster. This data is obtained by scanning a high accuracy metal target. The calibration data may then be used to correct subsequent scan results.

After correction, all extraneous coordinate points (those lying within an epsilon of a straight line) are removed from the line. Resultant lines are then displayed singly on the graphic console CRT. A typical display is shown in Figure 14.

The density of points will be a function of line curvature. The user then has a multiple choice option in which he may choose to utilize card data, typed data, or border data for supplying coordinate information relative to the line. The pencil must be used to point to one of the fields in order to select the appropriate mode of input. We have found that when processing a volume of information, the users will prepare data cards ahead of time and utilize this mode of alphanumeric input. If only one or two lines are to be processed, the typed input will more likely be selected. In this case, the STORE function leads the user through a series of questions and responses, in which he is either required to make a choice or type in a reply at the alphanumeric keyboard. In all cases, a message on the graphic console CRT instructs the user as to the next operation or required response. As an added

Figure 14. Display of Sampled Line.

feature, the user may select the BORDER DATA option, in which case the coordinates of all digitized points are related to the coordinates of the corners of a 20x20 inch border. The coordinate values of the corners of the border are assumed to have been preset by the CHANGE PARAMETERS function.

After all alphanumeric information has been entered, a summarizing message is displayed on the graphic console screen. An example is shown in Figure 15. The console operator is requested to pass judgment on the quality of the results before they are stored on disk.

RESULTS

The actual performance of the equipment and system can best be measured by the accuracy and detection capability of the scanning operations. For these reasons, a series of experiments was conducted, aimed at determining the accuracy and detection capability of the device under various operating conditions. Figure 16 is a plot of the distribution of errors in scanning a vertical straight line located at the center of the image field.

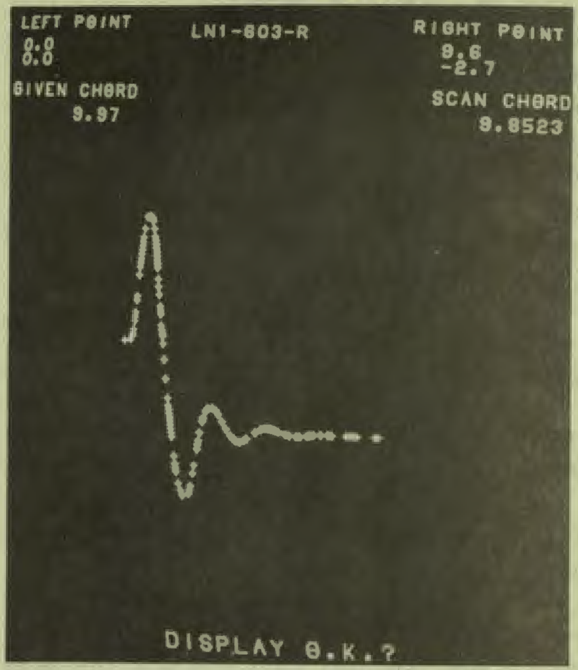


Figure 15. Summary Display.

It should be noted that this figure was generated via the CRT recorder which is a feature of the image processor. Also note that the unit squares along the chordal length of the line are measured in units of 5 inches, while the deviations are measured in units of .05 inches. As can be noted, the maximum deviation of the data from a straight line is ±.02 inches and there is a high frequency noise level of ±.005 inches. The original system objective was ±.01 inches average deviation on a 20x20 inch document. If this test is repeated with no utilization of calibration correction, the results are as shown in Figure 17.

Notice that while the high frequency noise level does not change, the maximum deviation or distortion is now .04 inches.

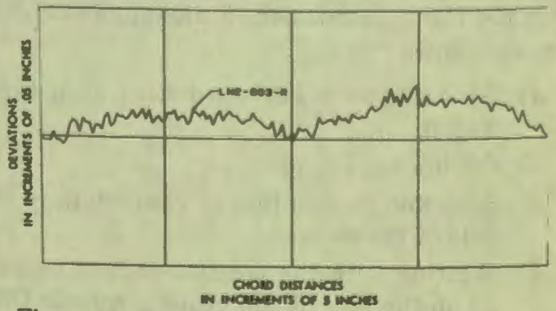


Figure 16. Error Distribution With Calibration.

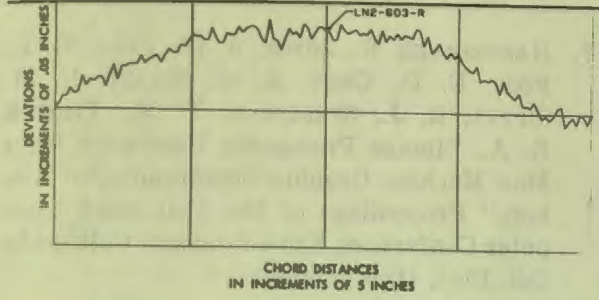


Figure 17. Error Distribution Without Calibration.

The second measure of system performance is the detection capability of the scanner. The range of threshold over which a line can be located is a measure of detection capability. Below a certain threshold level, the scanner can not detect any changes in the percentage of light transmission between background and lines. Above a certain threshold, the scanner will detect light over the entire image. This type of data may be plotted for a single film document for lines of various thickness. Figure 18 is a typical plot of minimum and maximum threshold level versus line width.

The maximum threshold level is the noise level, or the point at which the photomultiplier will always see light. This is a fairly constant value over the entire film image. The minimum threshold level is, of course, a function of line width, until such time as the line widths become appreciably greater than the effective CRT spot diameter. Various qualities of documents and various film exposures will move the wedge left or right and widen or close the wedge. For each particular digitizing application, wedge samples may be taken to determine the limits of detection. Optimum operating performance may be obtained from the combination of parameters which provide the widest wedge located furthest to the left with respect to line width. Extensive tests are now being conducted on a range of documents to determine these limits of detection. These test results will serve as a guide line for judging the quality of all documents.

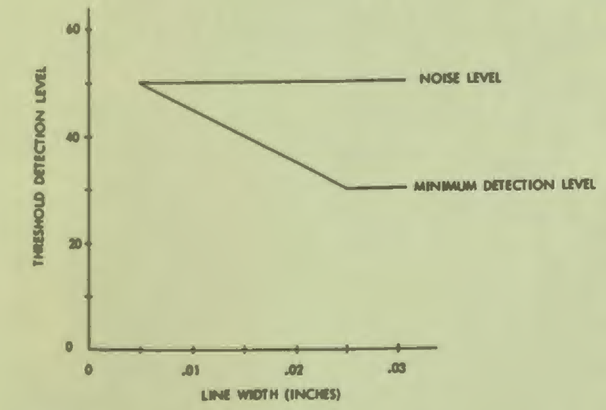


Figure 18. Detection Wedge.

CONCLUSIONS

This system can best be described as a line digitizer which is being used as an experiment in processing graphical data. It combines the speed and accuracy of automatic line tracking with the decision capability of a human operator. The system as conceived and implemented has proven the feasibility of close man-machine interaction. While it is not practical to operate the equipment and system with *no* training, users have become proficient in its use after only one or two hours of instruction. Through the use of graphic displays, it has been possible to program the computer to communicate with a console operator in a medium which is easily understandable. Thus, the utilization of a human operator as the key system component has been very successful. This has been particularly true when automatic line tracking is impractical and the user has had to intercede to assist the scanning.

ACKNOWLEDGEMENT

The line scanning system described in this paper was made possible through the efforts of many members of the Computer Technology Department programming staff. In particular, the authors would like to acknowledge the contributions of Gerald J. Devere and Dennis M. Walker.

REFERENCES

1. GRIMSDALE, R. L., "A System for the Automatic Recognition of Patterns," The Institution of Electrical Engineers, Dec. 1958.
2. GREANIAS, E. C., "The Recognition of Handwritten Numerals by Contour Analysis," IBM Journal, Jan. 1963.
3. FULTON, R. L., "Visual Input to Computers," Datamation, Aug. 1963.

4. McCormick, B. H., "The Illinois Pattern Recognition Computer-Illiac III," IEEE Transactions on Electronic Computers, Dec. 1963.
5. Jacks, E. L., "A Laboratory for the Study of Graphical Man-Machine Communication," Proceedings of the Fall Joint Computer Conference, San Francisco, California, Oct. 1964, (this volume).
6. Cole, M. P., Dorn, P. H., Lewis, C. R., "Operational Software in a Disc Oriented System," Proceedings of the Fall Joint Computer Conference, San Francisco, California, Oct. 1964, (this volume).
7. Hargreaves, B., Joyce, J. D., Cole, G. L., Foss, E. D., Gray, R. G., Sharp, E. M., Sippel, R. J., Spellman, T. M., Thorpe, R. A., "Image Processing Hardware for a Man-Machine Graphic Communication System," Proceedings of the Fall Joint Computer Conference, San Francisco, California, Oct. 1964, (this volume).
8. Allen, T. R., Foote, J. A., "Input/Output Software Capability for a Man-Machine Communication and Image Processing System," Proceedings of the Fall Joint Computer Conference, San Francisco, California, Oct. 1964, (this volume).

10264089 7