AD-A041 136 AIR FORCE HUMAN RESOURCES LAB BROOKS AFB TEX F/G 5/8
MAN-COMPUTER SYMBIOSIS THROUGH INTERACTIVE GRAPHICS: A SURVEY A--ETC(U)
APR 77 P A KNOOP
UNCLASSIFIED AFHRL-TR-77-10 NL

1 OF 1
AD
A041136

END
DATE
FILMED
7-77

AFHRL-TR-77-10

AIR FORCE HUMAN RESOURCES LABORATORY

MAN-COMPUTER SYMBIOSIS THROUGH INTERACTIVE GRAPHICS: A SURVEY AND IDENTIFICATION OF CRITICAL RESEARCH AREAS

By

Patricia A. Knoop

ADVANCED SYSTEMS DIVISION
Wright-Patterson Air Force Base, Ohio 45433

April 1977

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235

NOTICE

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This final report was submitted by Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base, Ohio 45433, under project 6114, with HQ Air Force Human Resources Laboratory (AFSC), Brooks Air Force Base, Texas 78235.

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DoDD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

GORDON A. ECKSTRAND, Director
Advanced Systems Division

DAN D. FULGHAM, Colonel, USAF
Commander

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

14 1. ~~REPORT NUMBER~~
AFHRL-TR-77-10

2. GOVT ACCESSION NO.

9 3. ~~RECIPIENT'S CATALOG NUMBER~~
Final

6 4. ~~TITLE (and Subtitle)~~
MAN-COMPUTER SYMBIOSIS THROUGH INTERACTIVE GRAPHICS: A SURVEY AND IDENTIFICATION OF CRITICAL RESEARCH AREAS.

5. ~~TYPE OF REPORT & PERIOD~~ COVERED
Professional Paper
Final

6. PERFORMING ORG. REPORT NUMBER

10 7. AUTHOR(s)
Patricia A. Knoop

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Advanced Systems Division
Air Force Human Resources Laboratory
Wright-Patterson Air Force Base, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62205F
61142002

11. CONTROLLING OFFICE NAME AND ADDRESS
HQ Air Force Human Resources Laboratory (AFSC)
Brooks Air Force Base, Texas 78235

11 12. ~~REPORT DATE~~
April 1977

13. ~~NUMBER OF~~ PAGES
40

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
12 41p.

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.
16 6114 17 20

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
Professional paper prepared in partial fulfillment of course requirements for the M.S. degree in Computer and Information Science, The Ohio State University, Columbus, Ohio. Copyright, Association for Computing Machinery, Inc. *Published in Journal of Communications of the Association for Computing Machinery*, March 1974, Volume 17, Number 3, pages 161 - 162 (TR-77-10, Figure 7, page 21). Reprinting privileges granted by permission of the Association for Computing Machinery, Inc.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

cathode ray tube (CRT) displays	man-computer interface
graphic languages	man-machine interface
interactive displays	symbiosis
interactive graphics	voice input techniques
instructor/operator stations	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this effort was to determine the research areas that appear most critical to achieving man-computer symbiosis. First, an operational definition of man-computer symbiosis was developed by: (a) reviewing and summarizing what others have said about it, and (b) attempting to distinguish it from other types of man-computer relationships. Most general views of it are typified by amplifications of the points originally raised by Licklider upon introduction of the concept in 1960. An operational definition was derived by casting the various types of man-computer relationships into a systems framework, and determining the major distinguishing features of each. The major distinguishing feature of man-computer symbiosis is the capability of both man and computer to transceive relevant unrequested information. ▷

(Cont on p 1473B)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404 415

1B

(cont fig 1473A)

Item 20 (Continued)

Using this

Using the derived definition, basic key requirements of a symbiotic system were projected. These included: (a) transceivers and memories, (b) graphics techniques, (c) a language, and (d) an attention-getting capability. These basic requirements were used to outline and provide structure to a literature survey to determine the state of the art and identify critical research areas.

The survey revealed that raster scan cathode ray tubes hold the greatest promise for use in symbiotic systems, but that a critical problem accompanies their use in that scan conversion methods are required. Existing methods are either too time consuming or too memory consuming, and attempts to improve on one of these aspects have all resulted in tradeoffs on the other. A promising technique was found for reducing the number of vectors comprising a scene. This technique would find its greatest utility in random scan systems, where its use may provide flicker-free displays whose densities approach those possible with raster scan systems. In the area of input device technology, it was found that few distinct improvements have been made in over a decade of research, and that no single device exists which is optimal for all basic types of input that the man would expectedly need to employ.

Despite considerable evidence as to its need, little work has been performed in extending man's memory precision and recall capability. The only notable research found was provision of a simulated short-term memory for application in automotive design work. Efforts to extend computer memory capacity center around the development of improved data structures, but there is some evidence that these efforts are misdirected.

There are no methods with immediate utility for symbiotic systems for efficiently removing hidden lines and surfaces to effect a 3D display. Most methods are too slow, with attempts at improvement only resulting in speed/memory tradeoffs. The only real-time solution is unsuitable for use with environments where many objects move independently, and is accomplished only at the expense of enormous processing power dedicated exclusively to hidden line/surface removal.

Finally, it was determined that the most promising solution to the language problem lies in the development of hybrid communication techniques using a combination of graphic languages and small scale voice input methods. This could offset some of the disadvantages of using graphic languages alone as well as provide a highly efficient attention-getting capability and enhance the rapport man feels with the computer.

The critical research areas that were identified were summarized in table form. In the author's opinion, advancements in these areas would effect the most rapid closure on achieving the type of system required for man-computer symbiosis.

SUMMARY

Problem

With the increasing complexity of modern weapon systems and simulators, new methods are required for providing efficient and effective interface between the instructor and the computer in flight training devices. Over the years, in efforts quite unrelated to training simulation, various studies have been conducted in attempts to create unique man-computer partnerships for solving problems. When, in such a partnership, both man and computer complement and benefit each other, the relationship is termed symbiotic. The problem was to find out what research areas appear most critical to achieving man-computer symbiosis for potential future applications in the design of simulator instructor/operator stations.

Approach

The approach was to determine a precise operational definition of man-computer symbiosis and use it to identify the requirements of a symbiotic system. The requirements were then used to outline and provide structure to a literature survey to determine the state-of-the-art and identify critical research areas.

Results

An operational definition of man-computer symbiosis was derived. The major feature that distinguishes it from other types of man-computer relationships is the capability of both partners to transceive relevant, unrequested information. Key requirements for a symbiotic system are transceivers and memories, graphics techniques, a language, and an attention-getting capability. A survey of literature in these four specific areas was conducted and critical research areas were identified.

Conclusions

Improved scan conversion methods for raster scan cathode ray tubes are required for widespread development and use of symbiotic systems. Also, some technique is needed for providing a simulated short-term memory for man's use in an interactive graphics environment. Hidden line and hidden surface display techniques are presently slow and inefficient. Advances in this area would constitute a breakthrough in display technology. Voice communication with the computer is probably the most influential change we will observe in the near future that will greatly enhance man-computer symbiosis. However improvements in word and phrase recognition reliability are needed. Finally, visual resolution processing techniques, which reduce display complexity by taking advantage of human perception characteristics, appears to be a promising compromise between limiting the information to be displayed and unrealistically increasing the computer resources required for the display portion of the system.

PREFACE

This report was written in partial fulfillment of course requirements for the M.S. degree in Computer and Information Science at the Ohio State University, Columbus, Ohio. The author sincerely thanks Professor R. L. Ernst for his helpful suggestions, as well as recommendation and personal loan of a number of papers used in the course of the survey. In addition, thanks go to Mr. J. D. Basinger, previously with the Air Force Human Resources Laboratory, Advanced Systems Division, Wright-Patterson Air Force Base, Ohio, for providing personal notes and papers on recently developed computer image generation techniques.

The work reported and results obtained are of direct use in helping to satisfy the objectives of project 6114, Simulation Techniques for Aerospace Crew Training, and task 611420, Advanced Training, with the author serving as both Project and Task Scientist. Therefore, although nearly two years have passed since the work was originally performed at the University, it was believed to be in the best interests of the Air Force to publish the results at this time for distribution to the scientific community.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

MAN-COMPUTER SYMBIOSIS THROUGH INTERACTIVE GRAPHICS: A SURVEY AND IDENTIFICATION OF CRITICAL RESEARCH AREAS

I. INTRODUCTION

The purpose of this paper is to determine and survey the research areas that appear most critical to achieving man-computer symbiosis. Since 1960, when Licklider published his now classic and often referenced paper introducing the term 'man-computer symbiosis' (Licklider, 1960), a tremendous volume of work has been performed in interactive computer graphics. The work is diversified, covering both hardware and software research and ranging from developments in graphic input devices and general display technology to graphic data structures, programming languages, and software/mathematical techniques for retrieving, sorting, and manipulating graphic data. Applications of interactive graphics are unbelievable in number and range from circuit and architectural design to complex problem solving.

Because man-computer symbiosis appears to be one of the major goals of and inspirations for much of this work, and because its goals are essentially to provide the ultimate in man-computer partnership, it was believed a worthwhile effort to find out where we stand in its regard. Six years after Licklider's paper, Sutherland (1966) identified several outstanding problems that remained unsolved at that time. Four years later, in 1970, DeGreene (1970) surmised that we were still many years away from operable man-computer symbiotic systems. And now what is the status of affairs? More important, what are the areas in which we can channel our best efforts to bridge the major gaps that remain? Finally, what is the state of the art in those areas of interactive graphics that, if mastered, can effect the most rapid closure on the goals of man-computer symbiosis? This paper is an attempt to answer these kinds of questions.

II. APPROACH AND OVERVIEW

A survey of work in critical problem areas, which is one of the purposes here, requires first the recognition of critical problems. This in turn requires some feeling for the key requirements of a man-computer symbiotic system to which any critical problems would expectedly be related. To ascertain system requirements necessitates, in turn, a definition of the system itself. The approach, then, was pretty much dictated by the purpose of the work and included the following steps:

1. *Determine an operational definition of man-computer symbiosis.* This proved to be no small task. Everyone intuitively "knows" what man-computer symbiosis is; therefore, no one has apparently felt a need to define it from an operational standpoint. The approach here was to review and absorb what others have said about it; attempt to contrast it with other types of man-computer relationships; and, finally, define it from a systems viewpoint.

2. *Project key system requirements.* Using the derived definition of man-computer symbiosis, three general requirements of a symbiotic system were projected. These requirements were stated in terms of "what basically do we need in such a system."

3. *Perform a general survey and identify critical problems.* A general survey of areas related to the established system requirements was conducted. This provided an excellent overview of the basic state of the art and revealed what are believed to be the major critical problem areas. Advancements in these areas would, in the author's opinion, most greatly benefit the progress toward man-computer symbiosis.

III. RESULTS

An Operational Definition of Man-Computer Symbiosis

General Viewpoints. Licklider (1960) originally discussed man-computer symbiosis in terms of combining the mutually beneficial talents of man and computer by means of: (a) minimizing their incompatibilities as potential partners, and (b) allocating tasks to them in accordance with their respective talents. He cites their major points of incompatibility as speed and language mismatches and discusses at some length their comparative capabilities. A few examples of the latter are: (a) men are noisy, narrow band, error prone devices with many parallel channels; computers are fast and accurate but do only one thing at a time; and (b) men are flexible and can program themselves contingently on the basis of new information; computers are single-minded and constrained by their preprogramming.

To varying extents, most references to man-computer symbiosis as well as interactive graphics per se essentially amplify on these basic concepts. David (1967) discusses at length the differences between man and computer and concludes generally that an effective interface must perform recoding and buffering. He also points out some of the considerations which limit the extent to which man can be made to adapt to the computer.

Miller (1965), emphasizing mainly the psychological and physiological capabilities and limitations of man, states that the man-computer relationship must be synergistic (mutually reinforcing) as well as symbiotic (mutually beneficial). He identifies three central human activities in problem solving (browsing, hypothesis formation, and hypothesis testing), and relates to each some implied requirements of a man-machine problem solving system. For example, browsing necessitates individualized selective display of timely information; and hypothesis formation requires a display which fosters the conceptualization of appropriate actions.

Smith (1970), in describing the design features of an interactive graphics system to solve numerical problems, cites the following basic system requirements: displays must be: (a) self-explanatory, and (b) self-helping (i.e., must check user inputs); user actions that are required should be simple and obvious; the system should possess optional verbosity so it can be used effectively and comfortably by a novice or experienced user; and feedback should be provided to the man when he acts. All of these requirements will be recognized as one form or another of accommodating the human's possession or lack of capability or his preference.

Both Matsa (1969) and Newman and Rogers (1966) highlight optimal task allocation as being the requisite for an interactive graphics system. Matsa states, for example, that regardless of the application, there is a certain amount of both man and computer required; and that interactive graphics involves combining the best of the capabilities of each by allocating tasks accordingly. Somewhat in contrast to this and some years earlier, Licklider (1962) pointed out, a few years subsequent to his introduction of man-computer symbiosis, that a strict task allocation is not really feasible because we cannot just divide a task into parts. His view is that partitioning a task into: (a) fine detail (computer's job), and (b) the overall (man's job) is not feasible because details, by themselves, are meaningless, and wholes do not exist without parts. While vague from the standpoint of suggesting a potential solution, Licklider's article conveys the message of 'whole > sum of parts' and provides a key concept in man-computer symbiosis.

In summary, both symbiosis and interactive graphics in general are typically viewed as problems involving: (a) minimizing differences between man and computer, mainly through appropriate interface; and (b) optimally allocating tasks to man and computer based on their respective talents. Distinctions between symbiotic systems and general interactive graphics systems are rarely, if ever, made in the literature. A key point made by Licklider subsequent to his introduction of man-computer symbiosis implies that straightforward task allocation does not necessarily result in a symbiotic system.

A Systems Viewpoint. The basic idea for the operational definition of man-computer symbiosis developed herein came from an article by Jacks (1968). In describing and rationalizing the design of an interactive system used at General Motors to solve automotive design problems, Jacks presents the human and computer as two systems, each possessing receivers, a processor and memory, and transmitters (see Figure 1). This viewpoint, of course, is the same as that adopted in control system design and analysis (Bekey, 1970). Its use by Jacks in the specific context of interactive graphics suggests a useful means of distinguishing symbiotic systems from those which are non-symbiotic.

Jacks discusses the chief characteristics of each component of the human and computer systems. He remarks that the central reason for using visual displays in interactive systems is man's ability to accept visual data (as opposed to other types of data) at a high rate. Considering the processor section of man, he remarks that little is known; however, man appears to operate in a burst mode—he accepts data rapidly, mulls over information to decide on an act, then releases a burst of information to the computer. On the computer side, Jacks discusses ways to accomplish matching I/O to the man (here again minimizing the differences between man and computer). His suggestions are both unique and valuable and seem to spring

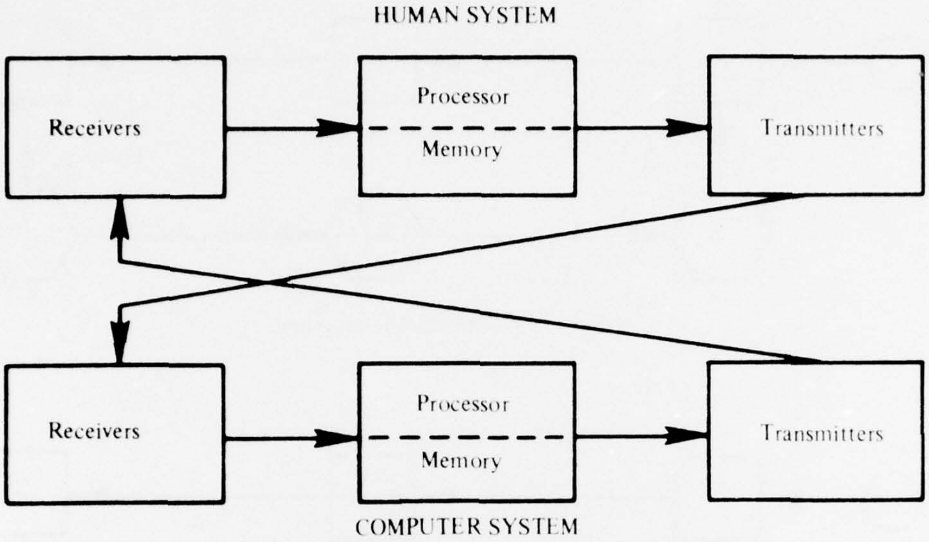


Figure 1. **Man-computer interactive system diagram.**

from his systems view of the situation. (One of his suggestions in particular, which will be discussed later in this paper, is to include on the computer side a simulated short-term memory for the man.)

By applying the systems viewpoint as Jacks did, various types of man-computer relationships can be categorized. It is proposed that there are three basic types of relationships: (a) the computer as an extension of man; (b) the computer as a partner of man, where the respective tasks of each are assigned via straightforward task allocation; and (c) man-computer symbiosis. Figures 2, 3, and 4 depict system block diagrams for each of these types of relationships.

In the extended man system (Figure 2), man receives external inputs about the task to be performed and his major interaction with the computer is to request a job to be done and receive a reply (data). This is a strict master/slave relationship wherein the computer receives no inputs except from the human and acts (responds) only when requested to do so. In the partner of man system (Figure 3) both computer and man receive certain external inputs relative to their respective tasks. Man acts as an overseer, in addition to performing his own tasks. His major interaction with the computer is to request and receive status information and, based on that information, override the computer's performance of its own tasks as required. In the symbiotic system (Figure 4), both man and computer can request information from each other as well as: (a) respond to a request for information, and (b) provide unrequested information.

It is proposed that the primary distinguishing features of the three systems are: (a) the major types of information traded between man and computer and, equally important, and (b) whether a transmittal to the man on the part of the computer is or is not always performed solely in response to a request from the man. Table 1 summarizes the three systems in terms of these distinguishing characteristics. Of particular note is that it is only in the symbiotic system that the computer independently generates information (or a request for information).

This proposed man-machine system classification scheme appears to be in accord with the views brought out originally by Licklider (1960) regarding man-computer symbiosis. Most important is that it incorporates his 'whole > sum of parts' concept (Licklider, 1962) by demanding that in a symbiotic system, the computer must be capable of independently acting, in a dynamic sense, on the basis of the overall system goals and the total system status at any given moment. This includes acting on the basis of what the man-component of the system is doing and has done. In contrast to its role in the other systems, the computer in a symbiotic system must be able to transmit *relevant unrequested information*—relevant to the goals of the system, and unrequested in the sense that the computer independently decides it is appropriate and of potential benefit for the man to receive it.

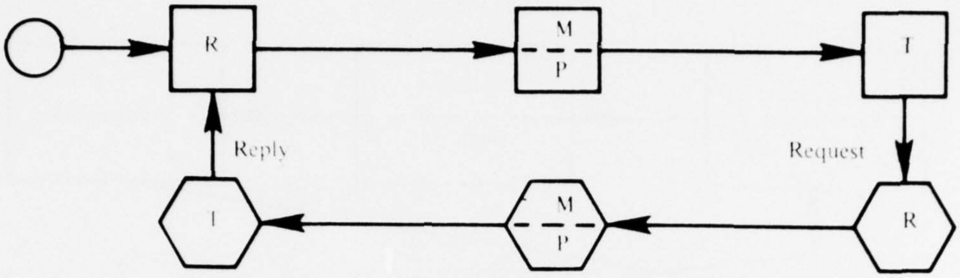


Figure 2. **Extension of man system.**

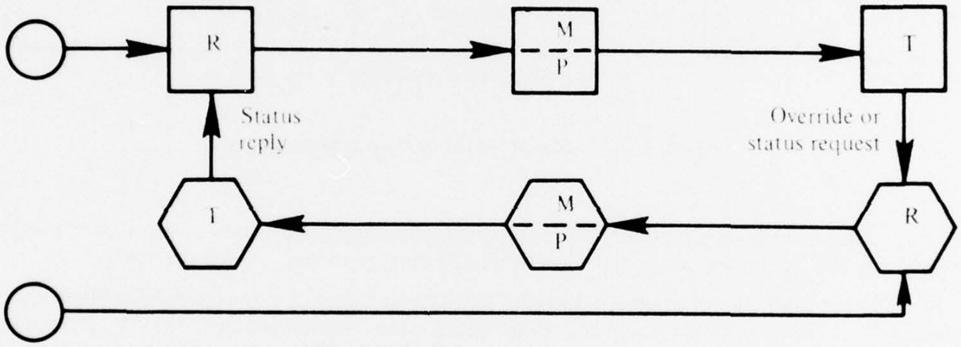


Figure 3. **Partner of man system.**

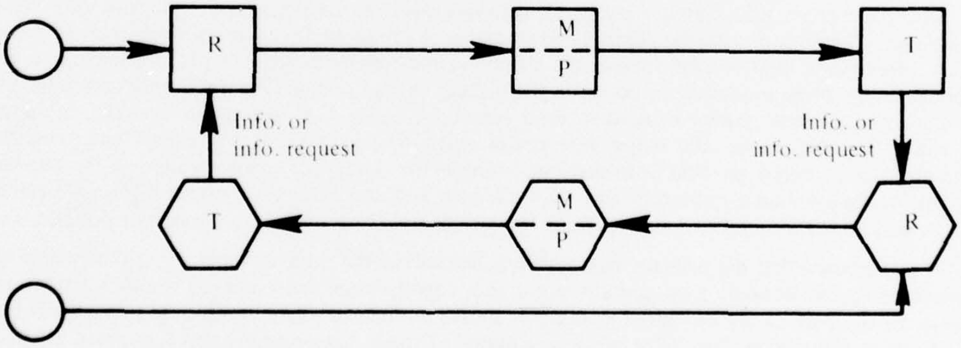


Figure 4. **Symbiotic system.**

○ – External inputs

□ – Man

⬡ – Computer

R – Receivers

M – Memory

P – Processor

T – Transmitters

Table 1. **Distinguishing Characteristics of Three Types of Man-Computer Systems**

	Major Type of Information Transmitted		Motivation for Transmission	
Type of System	Man to Computer	Computer to Man	Man to Computer	Computer to Man
Extension of Man	Request for Computation and Data	Results of Computation or Data Retrieval	Self-Motivated	Man's Request
Partner of Man	Override or Request for Status	Status Information	Self-Motivated	Man's Request
Symbiotic	Information or Request for Information	Information or Request for Information	1—Self Motivated 2—Computer's Request	1—Self-Motivated 2—Man's Request

An Operational Definition. Man-computer symbiosis is the joining of two subsystems (man and computer) into a single efficient system wherein both man and computer have the intercommunication capability of transceiving information that is relevant to the goals of the system. The information received by one or the other may or may not have been requested. The decision by one of the subsystems to transmit information is dynamically made as a function of: (a) the overall system goals, (b) the current and past system states, and (c) the current and past activities of the other subsystem.

Key System Requirements

From the definition in the previous paragraph and the major distinguishing features (Table 1) of various system types, it appears that the key requirement in a symbiotic system is the capability of man and computer to interchange relevant unrequested information. There are a number of system implications that can be derived from this general requirement.

The determination of whether or not information is relevant implies a need to know the status of the total system relative to its goals. Therefore, both man and computer require a *status transceiver and memory*. The fact that the information to be transmitted is both relevant and unrequested implies a required capability for arousing the attention of the receiving subsystem, *immediately if required*. Thus, an *attention-getting capability* is needed. Since the data to be transmitted is unrequested (unanticipated) and constitutes information, it is imperative that it be formatted for presentation in a way that will be clearly understood by the receiver. Therefore, *graphics techniques* are required for manipulating and formatting information to be displayed. Finally, the fact alone that the data to be transmitted constitutes information means that it inherently contains new knowledge for the receiver that is relevant to system performance. Therefore an unrestrictive, flexible, and easily applied means is required to "get the point across." Thus we need a *language*.

The above key system requirements are summarized in Figure 5 in terms of their emanation from the general requirement for transceiving relevant unrequested information. An attention-getting capability would expectedly be realized by means of the status transceiver and/or the language. Therefore it is considered a companion requirement to one or both of them rather than an independent requirement.

In summary, then, three key requirements of a symbiotic system have been identified: (a) status transceivers and memories, (b) graphics techniques, and (c) a language. These requirements emerge naturally from the necessity that each subsystem (man and computer) be able to transceive relevant unrequested information. Next, we turn to a general review of existing capabilities in areas related to satisfying each of these three key requirements.

General Survey and Identification of Critical Problems

In this section, a survey is presented of existing capabilities relevant to satisfying the key requirements for a symbiotic system. The purpose of this survey was to acquire a general understanding of

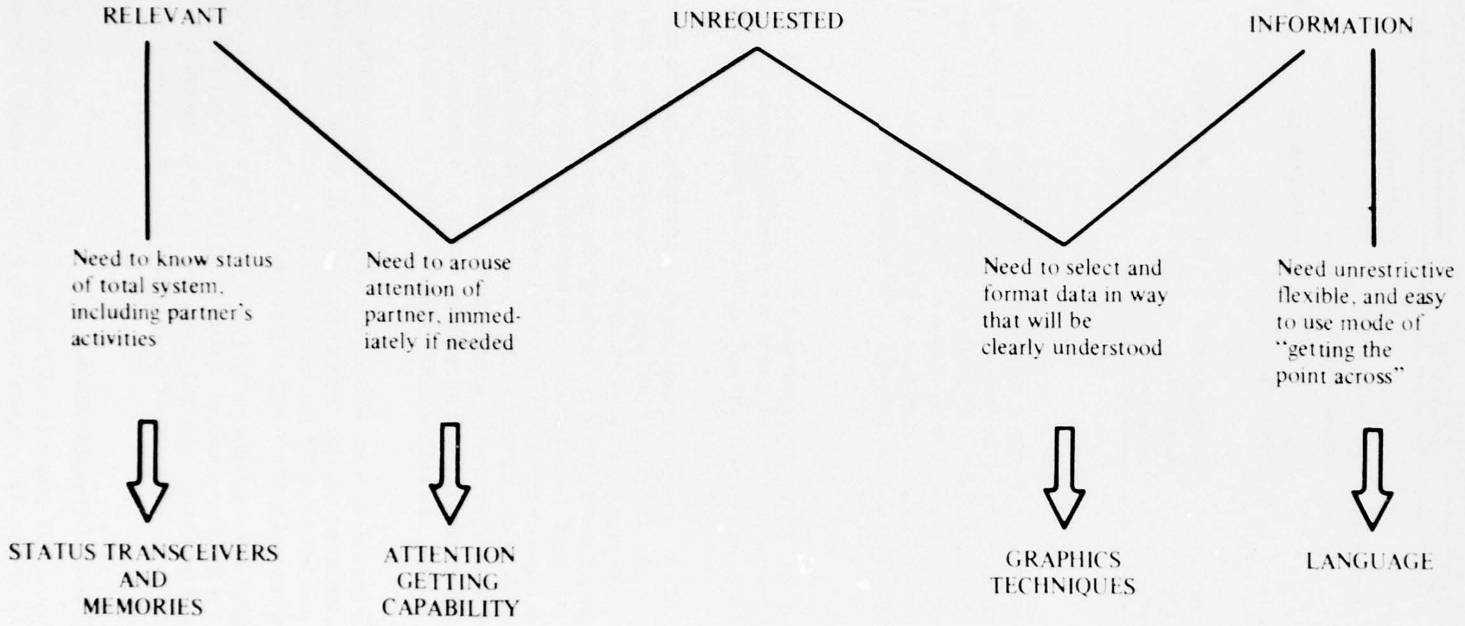

RELEVANT
UNREQUESTED
INFORMATION
Need to know status of total system, including partner's activities
Need to arouse attention of partner, immediately if needed
Need to select and format data in way that will be clearly understood
Need unrestrictive flexible, and easy to use mode of "getting the point across"
STATUS TRANSCEIVERS AND MEMORIES
ATTENTION GETTING CAPABILITY
GRAPHICS TECHNIQUES
LANGUAGE

Figure 5. **Key requirements for a symbiotic system.**

the state of the art and, equally important, to determine and explore those specific areas of research which seem most critical to the development of man-computer symbiotic systems.

Status Transceivers and Memories. Existing capabilities to satisfy requirements for status transceivers and memories can be partitioned into three groups: displays, input devices, and memory aids.

(1) *Displays*

Psychologists estimate that as much as 80% of our sensory data is received in the form of visual stimuli (Van Dam, 1966). This fact, plus the capability of man to accept visual information at high rates (Jacks, 1968), makes a computer driven display an effective and defendable selection as a means for computer-to-man communication. At the same time, however, effective visual communication using displays is not unbeset with problems. These problems arise largely out of the need to accommodate the perceptual characteristics of the human.

Considerable data are available on the visual perception characteristics of man and the resulting implications for display selection and design (see (Semple, Heapy, Conway, & Burnette, 1971) for an excellent review of related research and findings and extensive list of references). Generally the major factors which directly affect display selection and engineering are luminance, which man perceives as brightness according to a power law; luminance contrast, which is essentially the ratio of signal minus background luminance to total luminance, including that contributed by reflected ambient lighting; resolution; and display refresh rate (Gould, 1968). With the exception of luminance contrast, these requirements are generally *capable of being satisfied* using existing display technology. With the further exception of display refresh rate, the requirements are *in practice satisfied* in most existing applications. The key message which emerges from the literature (French & Teger, 1972; Noll, 1971; Sutherland, 1966) is that display refresh rate presents a problem not uncommonly encountered in interactive graphics work.

(a) Display Refresh Requirements

Display refresh rate refers to the frequency with which the visual scene on a display is refreshed or rewritten. Problems related to it arise when the refresh rate is close to the critical flicker fusion (cff) frequency of the individual. This is the frequency at and below which flicker in the visual scene is noticeable. The cff is dependent upon the individual as well as several other factors, including display brightness, contrast, and the ratio of viewing distance to display diameter (Gould, 1968; Semple et al., 1971). While flicker may be controlled and display refresh rates reduced by using long persistence phosphors, this limits the rate at which the display can be updated. Therefore, the refresh rate itself is the principal parameter in controlling flicker (Semple et al., 1971, p. 405).

A generally adopted rule of thumb is to refresh the display at least 35 to 40 times per second to avoid flicker (DeGreene, 1970). It may appear simple enough to merely adopt this rule of thumb and thus resolve the problem without further ado. However, minimum refresh rates are often required in high density displays because there is an inverse relationship between the refresh rate and the number of picture elements to be displayed (Semple et al., 1971, p. 411). This in turn is due to bandwidth limitations (in bits per second) on the computer display system itself. A typical manifestation of the problem is described by French and Teger (1972) who cite flicker as the major problem encountered in development of a graphical on-line design system. Their system capacity allowed the storage and display of 2,000 vectors; however, flicker occurred at 800. The principal solution recommended to help resolve the display refresh problem as well as several others is to use raster scan displays rather than the more conventional random scan types (Gould, 1968; Jordan & Barrett, 1974; Ophir, Rankowitz, Shepherd, & Spinrad, 1968).

(b) Raster Scan Cathode Ray Tube (CRT) Displays

Raster scan is one of five scanning methods used in computer data displays (Davis, 1969, p. 66). It is accomplished by deflecting the electron beam repetitively over a series of horizontal lines on the viewing screen and intensifying portions of the lines in response to input commands. Thus, information at position x,y on the screen is updated each time the electron beam scans line y. Commercial TV employs interlaced raster scanning, and it is with this technique that refresh rates may be reduced. Interlaced scanning refers to the technique of scanning (and updating information on) alternate lines on each of two successive passes down the screen (i.e., lines 1,3,5 etc. on pass 1 and lines 2,4,6 etc. on pass 2).

The reason that interlaced scan can reduce required refresh rates is due to man's persistence of vision (the tendency for a sensation resulting from a visual stimulus to remain after removal of the stimulus). This factor, plus the close proximity of adjacent scan lines, results in an effective refresh rate in an interlaced scan system that is higher than the actual refresh rate of each screen element. Thus, a TV system actually updates each displayed element at a rate of 30 per second, and though this is below "typical" cffs, flicker is not apparent.

Several other advantages to raster scan displays in interactive graphics work are discussed by Jordan and Barrett (1974). Chief among them, and of great relevance to widespread development and use of symbiotic systems, is the cost factor. For example, in one of the earliest uses of raster scan displays for interactive graphics (Ophir et al., 1968), a network of ten display consoles was developed for less than $80 thousand. A similar network using random scan character and vector generated displays would have cost from $300 thousand to $500 thousand.

Another significant advantage is that obtaining algorithms to remove hidden lines and surfaces from a picture can be accomplished with little extra effort using a raster display. This is because the more efficient algorithms are similar to those required to translate the standard x-y picture description into a raster format.

The major disadvantage of raster scan displays is that scan conversion techniques are required to perform the translation from x-y to raster format (Newman & Sproull, 1973; Noll, 1971). This can be accomplished using scan conversion hardware, but the equipment is fairly complex, adding to overall system cost and reliability problems. Therefore scan conversion is normally accomplished using software techniques. This, of course, adds to the computational load. Several scan conversion algorithms are available and are discussed in a subsequent section of this paper. However, it will be seen that scan conversion is a critical problem area.

(c) Random Scan CRT Displays

Despite their higher initial cost, higher refresh rate requirements, and the associated display density limitations discussed previously, random scan displays have been used far more widely in the past for computer graphics applications than have raster scan types. This is probably due to: (a) the lack of need for scan conversion, (b) excellent hardware character and vector generation capabilities existent in random scan displays, and (c) high commercial availability and advertisement.

However, in 1971, Noll predicted a distinct trend away from random scan displays for several noteworthy reasons. At Bell Telephone Laboratories, Noll and associates had, for some years, used a simple dot plotting CRT in man-machine communication research. The reason hardware vector and character generators were not used was because of their expense and associated reliability problems (they were a "continuing source of trouble" (Noll, 1971, p. 143)). The dot plotting approach was not satisfactory, however, due to flicker problems encountered with high data density (over 4,000 dots) and the considerable core storage required to hold each x-y coordinate pair. Therefore a scanned display system was implemented for use with a significant resulting improvement over their previous system.

Random scan CRT displays of a wide variety are commercially available in a highly competitive market. They may be basically classified according to the type of information presented: alphanumeric, graphic, or situation displays (Davis, 1969). In alphanumeric displays, alphanumerics and some symbols are presented in page format on the CRT. In graphic displays, line drawing capability is added to an alphanumeric capability. Situation displays may have characteristics of both alphanumeric and graphic displays in addition to an added capability of presenting some form of background data (e.g., radar or sonar).

The most common type used in interactive graphics work is the graphic type CRT. Unit cost is largely a function of the hardware options available. These include scaling, zooming, and 2D and 3D line rotation, plus variable intensity for achieving depth cueing. A review of some specific typical graphics displays is provided by David (1969). Due to the wide competitive market, there appears to be no inherent problem in obtaining (for a price) a random scan CRT display suitable for a specific application, given that the application can live with the basic data density limitations and associated refresh requirements mentioned earlier.

(d) Other Display Types

Other CRT devices are available (e.g., flat panel displays and direct view storage tubes); however, for various technical reasons, none have achieved the flexibility, relatively low cost, and wide availability of the conventional raster or random scan devices. Comprehensive reviews of alternative CRT devices are provided in Davis (1969); Kleiman (1969); and Newman and Sproull (1973). The direct view storage tube, for example, appears at first glance to provide a potential answer to the refresh problem, because it contains a built-in storage grid behind the front screen. However selective erasure of screen elements is difficult, if not impossible (Newman & Sproull, 1973), and picture quality is poor compared with that achieved by conventional random and raster scan CRTs.

Engineering development work in the display area is abundant, as might be expected due to the potential market for improved display devices (cf IEEE Conference, 1972). However it is probable that these other display devices will only achieve the versatility, low cost, and wide commerical availability of the more conventional CRT types a number of years from now. Thus, their widespread use in man-computer symbiotic systems, if it occurs at all, lies sometime in the future.

(2) *Input Devices*

If there is any area related to interactive graphics that has received as much attention as visual display technology, it is input device technology. Input devices other than the familiar keyboard can be classified according to whether they lend themselves chiefly to pointing, positioning, or drawing (Newman & Sproull, 1973). Most of them are better at one of these than the others. Therefore, while a collective capability for all three basic human actions exists (pointing, positioning, and drawing), use of any one input device usually results in a degraded capability for one or two of these actions.

(a) Light Pen

Possibly the most commonly used input device is the light pen. This is principally a pointing device consisting of a photocell and optical system which focuses into it any light in the pen's field of view. A switch on the pen is used to shutter light to the photocell when desired.

Output from the pen is amplified and fed to a flip-flop which can be read and reset by the computer. Thus, when the pen is used for pointing, the flip-flop must either be tested during the refresh cycle after displaying each point or sensed through an interrupt scheme. The time required for the pen to set the flip-flop becomes critical, because if this time exceeds the time it takes to execute a display instruction (typical situation), incorrect data will result regarding the display element at which the pen is pointed (Newman & Sproull, 1973). Therefore, most of the work in light pen technology concerns attempts to achieve faster response times from the pen. One technique is to use photomultiplier tubes (highly sensitive photocells); however, they are too bulky to be hand held and the light must be focused into them using a fiber optic pipe, adding to the complexity and cost of the pen. Another variation is to use transistor-type devices. These are small and inexpensive but have a slow response time (1 to 2 microseconds). Therefore they are suitable only for slower displays.

When used for positioning or drawing, the light pen requires a pen tracking program which continually senses the pen's position over the screen. Rapid movement of the pen can cause the tracking program to "get lost," so special techniques are needed to handle this, adding to computer loading and software complexity (Rundle, 1971).

The low cost of the light pen ($1,500. (Machover, 1972)) probably explains its popularity and widespread use despite a number of disadvantages associated with it. These include: (a) unnatural feeling when held in hand, (b) lack of precision due to pen's aperture and varying distance from the screen, (c) experience of fatigue when used with vertical CRTs, (d) slow response time, (e) necessity for frequent button-pushing, and (f) inconvenience incurred by the necessity to attach it by a cable to computer interface equipment (DeGreen, 1970).

(b) Tablet Based Devices

1. *Acoustic Pens and Tablets.* Input devices which are based on acoustic rather than optical principles have also been developed. Apparently, the initial motivation for such devices was the fact that light pens cannot be used with direct view storage tubes, and it was desirable in general to make the pen position sensing independent of display refresh (Machover, 1972).

The earliest acoustic device was the Lincoln Wand developed in 1966 (Newman & Sproull, 1973). This consisted of a hand-held ultrasonic transmitter emitting sound pulses picked up by four microphones at the four corners of the screen. This allowed input in 3 dimensions by calculating the pen's x,y, and z coordinates as a function of the delay between sound pulse transmission and reception. However, the calculations required were complex and consumed too much computer time to provide an efficient system.

An improved device is the Graff Pen Sonic Digitizer available commercially through Science Accessories Division (Newman & Sproull, 1973). This consists of strip microphones along two adjacent edges of a tablet. The stylus has a piece of ceramic mounted close to the tip, and at regular intervals emits a spark across the ceramic surface between two electrodes. This pulse of sound is picked up by the microphones, and two counters record the delay between generation of the spark and reception of the sound pulse. The device also works in 3D and is the most successful of such devices now available and the only 3D input device commercially available (Burton & Sutherland, 1974). Its weaknesses are that sampling is limited to 100 per second; response time is limited to the (relatively slow) speed of sound in air; and its accuracy is limited to 1 in 500 due to air movement even in a quiet room (Burton & Sutherland, 1974).

2. *Voltage Gradient Stylus/Tablets.* These devices are ones in which the stylus is used to sense a potential on the surface of a tablet consisting of semiconductor material. The x-y coordinates of the stylus are then determined by measuring this potential during the time periods in which it has been applied to successive horizontal and vertical lines across and down the semiconductor sheet (Newman & Sproull, 1973). The major problem with such devices is in finding a suitable tablet material with uniform sensitivity so that a reasonable potential can be developed.

The Sylvania Tablet (Teixeira & Sallen, 1968) works basically on the voltage gradient principle but detects stylus position in a slightly different manner. The tablet consists of a resistive sheet between two glass plates, and sheet signals are high frequency ac wherein the phase varies for each stylus position. Two different frequencies are used for horizontal and vertical signals, and the received signal is filtered into two frequency components whose phases then correspond to stylus coordinates.

3. *Rand Tablet.* Of all the devices developed either experimentally or commercially over the last decade or so, few distinctive improvements (particularly regarding accuracy and linearity) have been made over the Rand Tablet developed in 1964 (Davis & Ellis, 1964). This device was developed by Rand Corporation and further refined and marketed by Bolt, Beranek and Newman (Newman & Sproull, 1973). It consists of a 10 by 10 inch drawing area composed of two planes (x and y) of 1,024 lines separated by a thin mylar sheet. Each of the 1,024 copper lines in both planes carries a unique digitally coded signal that is detected by the stylus. Gray code is used so that the sequence of pulses in adjacent wires differs by only one pulse position. Pulses received through the stylus are amplified and sent to a decoding network where Gray to binary integer conversion is performed. The disadvantages of the Rand Tablet are its high cost ($10- to $15-thousand, (Machover, 1972)), high complexity, and the fact that the stylus itself is susceptible to wear and tear (Newman & Sproull, 1973).

(c) Stylus Independent Devices

In all the devices discussed so far, a stylus or pen is required for use in conjunction with the display surface or a tablet. The desirability of freeing the man from the need to hand-hold such a device has spawned efforts to develop techniques for detecting finger position, touch, or pressure.

1. *Pressure and Touch Sensitive Devices.* Hlady (1969) provides a review of techniques developed for touch or pressure sensitive devices. The earliest approach was to construct a grid of wires terminating at the front surface of the display tube and detect a point of touch by the influence of the body's capacitance on the underlying circuit. Hlady recommended an improved technique doing away with the grid of wires wherein surface waves are propagated on a transparent glass plate, and any object contacting the surface reflects energy to the source.

More recently, Pobgee and Parks (1972) described a pressure sensitive device based on the use of a membrane suspended over a solid surface. This technique and others in the class of pressure-and touch-sensitive devices suffer the disadvantage of incurring background noise from fingerprints and scratches on the sensitive surface.

2. *Position Sensitive Devices.* Salminger (1972) describes a technique ("light gate field") wherein tne display screen is divided into areas, each of which can be singled out by placing a finger or other object close to or upon it. The position of the finger is detected by its interruption of infrared signals passing from one side of the screen to a semiconductor light detector on the other. In the described application, the screen was partitioned into 30 areas. As might be obvious, resolution is a key limitation of the utility of this technique.

(d) Other Devices

Other input devices are primarily based on the positioning of a cursor on the screen by various techniques. The joystick and track ball (Davis, 1969) are two such devices, and they rely on visual feedback from the displayed cursor position to compensate for their lack of linearity. The Stanford Research Institute "mouse" (Engelbart & English, 1968) is similar to the track ball in principle, but uses wheels connected to potentiometers to provide a simpler and lower cost device.

(3) *Memory Aids*

In the earlier discussion of system requirements for man-computer symbiosis, the potential need for memory aids arose out of the more general requirement for status transceivers. On the computer side, memory is generally recognized as small (in comparison with man's) but precise; on the human side, man has a relatively large but imprecise memory. Therefore, improvements relative to symbiotic system development would expectedly be to increase the computer's memory capacity and increase the man's precision and recall capability.

(a) Computer Memory Capacity

Major efforts in this area seem to be concentrated on developing effective graphics data structures to permit mass storage to be used effectively. Essentially, then, the emphasis and the problem are not so much to provide greater memory capacity as it is to make efficient use of what we have, considering the serial nature of the computer as a processor.

The scope of this paper did not permit or warrant a survey of graphics data structures. One is provided fairly recently in Williams (1971), with a more general review in Newman and Sproull (1973). Much of the documented data structure research appears aimed at more and more effective means of building one type of display file. This is the structured file, wherein various picture entities are grouped in alternative ways so that they can be combined with other entities by the display processor in forming a complete picture. However, as Newman (1971) points out, there are two other types of files existing in graphics systems. One is the display file itself, which consists of lists of instructions for the display with all graphic data scaled and transformed as required. The other is a file which allows the user to build and modify the structured display file. Figure 6 illustrates these three files and their respective positions in a general graphics system.

The point of Newman's article is that three separate types of display files are generally implemented, adding to the memory and processor load that we are trying to reduce and make more efficient. He goes on to point out that some efforts have been made to reduce the display file triplication by combining the modifiable display file with the structured display file, but that most are imitations or variations of the same technique used in 1963 in developing Sketchpad (Sutherland, 1963). His own recommendation is to eliminate the structured data file altogether. He claims (p. 651) for one thing that its existence discourages the development of machine independent graphic languages because it is specifically structured for the display being serviced to minimize refresh time. As a result, it is generally difficult, if not impossible, to convert graphical programs from one machine to another. A distinct message acquired from Newman's article is that perhaps our efforts should be channeled toward more effective means of eliminating display files rather than toward building new data structures to handle the files as conventionally organized.

(b) Human Memory Precision and Recall Capability

Something accomplished in nearly every interactive graphics system is to permit man to access the computer's memory. This, in a sense, enhances his own memory precision and recall capability by allowing him to selectively sample data stored in the computer memory. But what about the problem of recalling: (a) the nature of the information that should be recalled, or (b) information which is specific, on a short-term basis, to the continually changing status of the system relative to its goals (which may be multiple)? There are some types of information which man requires perhaps only for a minute or two in a given unique application of the system and which, therefore, cannot be committed to man's long-term memory nor specifically anticipated and preprogrammed for storage in and retrieval from the computer's memory. Thus, except in highly rehearsed system applications, one would expect man's short-term memory (STM) to play a principal role.

The problem arises from the fact that man's STM is extremely capacity-limited and, in the absence of sustained attention, loses its store rapidly. Considerable research has been performed on the characteristics

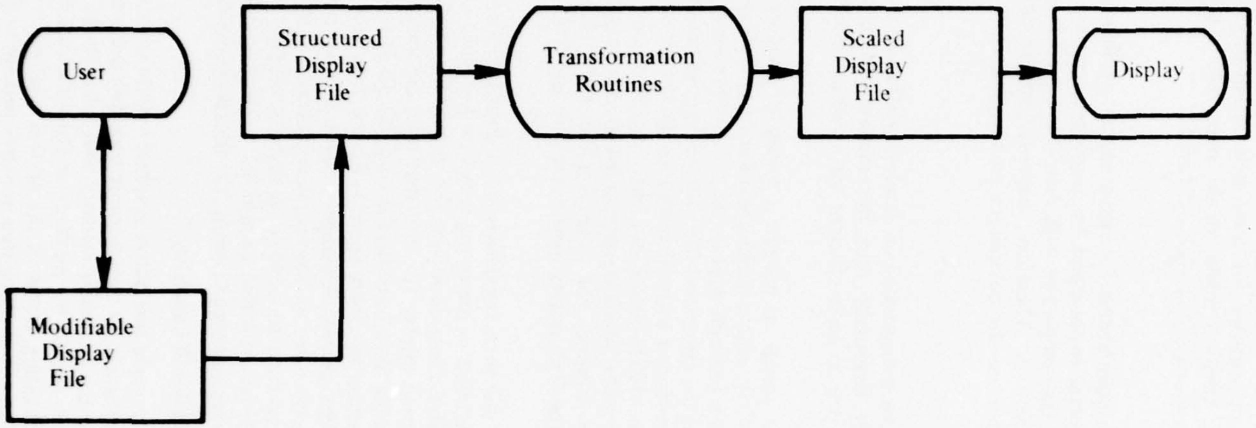


Figure 6. **Arrangement of three types of files in a graphics system.**

of STM (Fitts & Posner, 1969). It has been found that STM capacity is roughly 7 plus or minus 2 items, and that retention time ranges from 500 milliseconds to 15 minutes, depending largely on the nature of the items.

1. *Simulated STM.* Jacks (1968) describes a unique method of using the computer to provide a simulated STM for the man for short-term store of relevant data. His chief reason for a simulated STM requirement is due to man's processor characteristics when acting in the role of a problem solver (automotive design problems at General Motors in this case). These characteristics include the application of trial solutions and subsequent development of strategies. Jacks points out that even though man may develop a strategy, he cannot usually apply it a second time at a higher rate as a sequence of operations. In fact, he may have forgotten how to apply it at all and need to redevelop it. Thus, Jacks' simulated STM is primarily a short-term procedure memory. Using it, much of the design process becomes one of selective reapplication of recorded strategies.

Miller (1965) had earlier suggested the development of STM aids to compensate for man's poor ability to perform logical operations which, he said, was due in turn to his limited STM capacity. Despite this early suggestion (1965) and the very good evidence accrued by psychological research on the need for such aids, the literature reveals little if any active research on simulated STM for interactive graphics systems (with the sole exception of Jacks' 1968 article).

(4) *Review and Conclusions*

In way of review, the symbiotic system requirement leading to the above survey of capabilities in status transceivers and memories was that man and computer need to be able to determine whether or not information is relevant. The implied requirement was a need to know the status of the system relative to its goals. Accompanying these needs was a requirement for an attention-getting capability.

(a) Computer-to-Man Communication

Of all display types it appears that raster scan CRT systems hold the greatest promise for widespread use in symbiotic systems. This is principally due to the lower refresh rate requirement when interlaced scanning is employed and the significantly lower cost of raster scan displays. Little mention is made in the literature about attention-getting techniques beyond the traditional blinking of a display message. The success of this method is dependent on the assumption that the man is looking at the display at the time his attention is required.

(b) Man-To-Computer Communication

Most input devices are designed primarily for pointing or positioning and drawing, but none are optimal simultaneously for all three. Of all devices available, the Rand Tablet developed in 1964 is apparently the most accurate and linear; however, it is also one of the most expensive and the stylus is susceptible to wear and tear. Stylus-independent devices have primarily been geared to providing a pointing capability, but resolution is a chief limitation. One of the least expensive devices is the light pen, but its response times are relatively slow.

(c) Memory Aids

The only significant work in memory aids for the man appears to be that done by Jacks in 1968 in the development of a simulated STM. However his work was application oriented and included only procedure memory for use in automotive design work. Work on developing memory aids for the computer lies primarily in the development of improved data structures. However it has been suggested that efforts be channeled instead to reduction of the number of display files required in the system.

(d) Conclusions

Display technology is generally adequate, but few if any applications have incorporated reliable means of attention getting. Raster scan CRTs are preferable due to the lower refresh rates required using interlaced scan; however, these CRTs necessitate scan conversion and, so, are not without their own disadvantages. Despite an enormous amount of research, little definitive progress has been made since 1964 in providing improved, flexible, low-cost input devices. Devices which appear best suited for man's use in transmitting status information are poorly adapted to use as attention-getting devices. Little relevant work has been done to improve man's limited STM, despite considerable evidence of its need that has been accrued through psychological research. Techniques for improving the computer's memory capacity are largely geared to enhancing data retrieval speed through the development of better graphic data structures. Potential benefits may exist in efforts to reduce the number of display files.

Graphics Techniques. Graphics techniques may be classified according to whether they are geared to two- or three-dimensional data representation. These classes are not mutually exclusive because most 2D techniques are useful as well in 3D picture portrayal. However 3D techniques are in a class by themselves due to the special problems encountered in achieving perspective and depth cues.

(1) *Two-Dimensional Techniques*

Two-dimensional techniques include basic operations on data such as rotation and translation, pen-tracking techniques to permit drawing on the screen with a light pen, scan conversion methods for use with raster scan displays, and visual resolution processing, which is a data reduction technique recently applied in interactive graphics.

(a) Basic Operations

Translation, rotation, scaling, and zooming involve the application of simple transformations which can be concatenated to achieve desired goals. Scissoring, commonly done with hardware, involves selective intensification of points only when they lie within the screen boundary. Newman and Sproull (1973) point out, however, that scissoring still consumes time to trace invisible portions and can increase the probability of flicker. The alternative is to clip the data (restrict it to the screen boundaries) before display, and this can be done either with hardware or software. All of these basic operations are straightforward and well developed, posing no noteworthy technical problems.

(b) Pen Tracking

One of the tasks which cannot be performed using hardware is tracking the light pen when it is used to generate drawings (Rundle, 1971). This is not as simple a problem as it might seem, because the light pen is dependent on light from the CRT to permit it to deliver position data. Therefore, the display system must provide a source of light for the pen which follows the pen when it is moved across the screen. If moved too rapidly, the tracking program may lose the pen completely, and a search for it is necessitated.

The typical approach (incorporated in pen-tracking software provided by the manufacturer) is to generate a source of light in the form of sequentially drawn radial lines from a center position (Davis, 1969). Any movement of the pen away from the center causes it to intercept one of the radial lines, cueing the tracking program to the proper direction in which to move the light source. If the pen gets lost, the usual relocation method is to begin a spiral search from the last known position.

(c) Scan Conversion

To use a raster scan CRT system, some means is required to convert the x-y representation of data into scan line format. This operation is known as scan conversion, and it can be performed using analog hardware or software (Noll, 1971). Scan conversion hardware is expensive and complex. One technique, for example, is to use a conventional CRT driven by x-y coordinates from the computer as an intermediate output. The face of this CRT is then scanned electronically to accomplish the required conversion. Due to the reliability problems and expense of such equipment, scan conversion is usually performed by software (Noll, 1971).

The conventional method of software scan conversion is to produce a direct binary image of the entire display from the x-y coordinate description of the picture (Jordan & Barrett, 1973; Noll, 1971). This image can then be stored in core for line by line read-out to the display or on some peripheral storage device interfaced to the display (e.g., drum). The latter method was used in the first application of raster scan displays in interactive graphics (Ophir et al., 1968). The disadvantage of this technique is the sacrifice of efficiency resulting from the access latency of the drum used for storing the binary image. The former method (core storage) was used in what appears to be the second application of raster scan displays in interactive graphics (Noll, 1971). The disadvantage of this approach is that random access memory is relatively expensive and storage requirements are large. For a screen size of 512 by 512 dots, 262,144 bits are needed to store an entire binary image. Also, as Noll (1971) points out, this figure would be tripled if, say, two additional bits were used to determine the intensity of each point to provide four brightness levels. *Thus, a major problem in the conventional scan conversion method is the amount of storage required for the binary image.*

A second problem is the time required to perform the conversion. Although the calculation itself is straightforward, it must be performed for each x-y coordinate pair. In Noll's (1971) application, the calculation for each pair required 200 microseconds on a DDP-224 computer. To scan convert a picture filling the equivalent of one-third of a 512 by 512 dot screen, about 17½ seconds would be required. Fortunately, as Noll points out, it is not always necessary to update the entire picture and do a complete, new scan conversion. Often what is needed is just a modification of the scene already being displayed. This leads to a third problem associated with the conventional scan conversion technique.

The problem arises from the necessity to delete some portion of the currently displayed picture. Noll (1971) provides a classic, often referenced example of the problem that arises by presenting a hypothetical display situation in which a single line and a dot are displayed. The dot is moveable by the user via two knobs. Each time the dot changes position, its previous position on the screen must be blanked and the new one illuminated. If the background were initially made black by placing all zeros in the binary image, then a 0 must be placed in the previous dot position to erase it, and a 1 would have to be placed in the new position to illuminate it. The problem arises when the dot crosses the line, in which case this technique would result in removal of one dot from the line itself. For this reason, special precautions are required when deleting existing picture elements using a conventional scan conversion process wherein a direct binary image is maintained. *Thus, difficulty in picture modification may be considered a third disadvantage of this technique.*

A method has been proposed by Jordan and Barrett (1973) to reduce the memory requirement associated with the conventional scan conversion method. Their scan conversion technique to accomplish this is a modification of one commonly used in conjunction with hardcopy systems where high read-out rates are not required. In these systems, a small buffer area is used over and over to produce sequential groups of scan line images (e.g., 10 scan lines at a time). Each time, a complete image is generated from the x-y display file, but only those points corresponding to the part of the image to be plotted are stored in the buffer. The method developed by Jordan and Barrett follows a similar procedure but is faster because it only requires the binary image to be generated once rather than every time a new group of scan lines is to be processed.

The Jordan and Barrett procedure begins by sorting the x-y display file according to the scan line group in which each line will first appear in a top-to-bottom scan. The results of this sort are stored in a linked list data structure in which each list item (a single line) is represented by the beginning and ending coordinates of the line and an additional x coordinate used for maintaining the current x value as the line is traced through a given set of scan lines. In processing the linked list, a dot generating algorithm is applied to each line in the current scan line group, terminating if the line leaves that group by updating the current x coordinate value and attaching the list item to the next scan line group by changing the link. If the line is terminated within the present scan line group, the corresponding list item for that line is removed from the active list, and processing is begun on the next item.

Speed and storage requirements were analyzed by Jordan and Barrett (1973) using a PDP-8 computer. Using 16 scan lines per scan line group, processing time was only slightly greater than that required using a conventional scan conversion technique on the entire raster (e.g., about 2.5 seconds longer for a total line length of 100 raster units, and about 1.25 seconds longer for a line length of 40 raster units). However, memory requirements were only about 20% of that needed using the conventional technique. The authors conclude that a choice of 16 or 32 for the scan line group size will result in at least an 80% reduction in storage requirements, with less than 10% increase in execution time.

The problem of picture modification was also attacked by Jordan and Barrett in another effort (1974). They developed a design for a cell organized raster scan display in which lines (vectors) as well as characters can be processed. Conventional low-cost alphanumeric displays are based on a cellular partitioning of the screen, each cell being large enough to display a single character. This allows the character generator to be implemented with a read-only memory which stores the dot patterns for all alphanumerics. On a cell by cell basis, the generator fetches the display command from refresh memory and, based on the command itself and the current scan line being processed, retrieves one row of the characters from memory. The generator continues to process one row of each cell in the scan line before returning to the next row of the first cell. Thus, as Jordan and Barrett note, each cell is processed several times before it is completely displayed.

Their technique includes expanding the read-only memory to include line segments as well as alphanumeric characters, and placing the refresh memory in front of the character and vector generator to permit modification of picture elements in their original x-y configuration. (This differs from the implementations discussed previously (Noll, 1971; Ophir et al., 1968), where the character/vector generator preceded the refresh memory.) To implement the line segment capability in memory, a minimal set of basic patterns was derived which, when processed by the vector generator, can produce a line segment of any length and orientation. To make this possible, the generator itself included capabilities for reflection about the x axis, translation, shifting, masking, and union (binary ORing) of patterns.

Two alternative scan conversion techniques for this proposed display system were developed (Barrett & Jordan, 1974). One technique is an extension of the method described previously (Jordan & Barrett, 1973), which implements processing of all lines in one cell before proceeding to the next. Thus, in addition to transferring the current x coordinate value for a line to the next scan line group when required, the technique operates on a cell by cell basis and transfers the current x coordinate value to the next cell as required. This is termed a presorting algorithm since, as described earlier, the lines of a drawing are presorted according to the row in which each line will first appear in a top-to-bottom scan (i.e., sort lines *before* generating display commands). The authors point out (Barrett & Jordan, 1974) that this method is not well suited to interactive graphics, where a small number of screen elements are modified frequently. The reason is that the entire x-y representation of the picture must be available before the scan conversion process begins, due to the technique of tracing each line to its termination point to record all endpoints prior to starting the scan conversion. Therefore, to modify a picture the x-y representation must first be updated, followed by a complete scan conversion.

The alternative method is termed a postsorting algorithm wherein display commands for an entire line (vector) of the picture are generated first, then placed into their proper positions in a cell display file (i.e., sort display commands *after* they are generated for a given line). This allows commands representing any given line to be added or deleted without affecting the commands for other elements of the picture. Thus, it is well suited to an interactive graphics situation but, as the authors note, requires a more complex data structure.

An analysis of execution times and memory requirements was performed for the two cell oriented scan conversion techniques and compared with those of the conventional technique (Barrett & Jordan, 1974). A summary of key results is presented in Figure 7. The presorting technique took slightly longer than the conventional technique but required less memory—roughly the same comparative results as those obtained in the earlier study (Jordan & Barrett, 1973). The time required to add or delete a line, however, was significantly greater than for either the conventional or postsorting techniques. The conclusion was that the presorting algorithm is suitable for static display updating but, as anticipated, not well suited for interactive graphics applications.

The postsorting technique required considerably more time than either of the other two. For example, on a CDC 6400 computer scan conversion of a total line length of 120 units of raster width (equivalent to about 23% of a 512 by 512 dot screen) required about 24 seconds using the postsorting method, and between 3 and 5 seconds for the other two methods. Memory requirements for the postsorting method exceeded those using presorting. They were less than those using the conventional technique only for total line lengths not exceeding about 125 units of raster width, at which point and beyond they equalled or exceeded the memory requirements using the conventional method. The time to modify a line using postsorting was slightly greater than that using the conventional method for line additions, but significantly less for line deletions. This, of course, is where the conventional method encounters one of its major problems. For example, times to add and delete a line in a picture containing a total line length of 45 units of raster width were about 1.35 seconds and .03 seconds, respectively, for the conventional method. But the time required using the postsorting method was only about .16 seconds for both addition and deletion. Thus, the authors conclude that the postsorting scheme represents a significant improvement in interactive capability over the conventional raster scan conversion technique.

(d) Visual Resolution Processing

All of the 2D techniques discussed up until now have represented operations on graphic data, largely to transform it into a desired representation for display. Bussell and Taxin (1972) have described an additional technique which is aimed at reducing the quantity of data to be displayed without noticeably

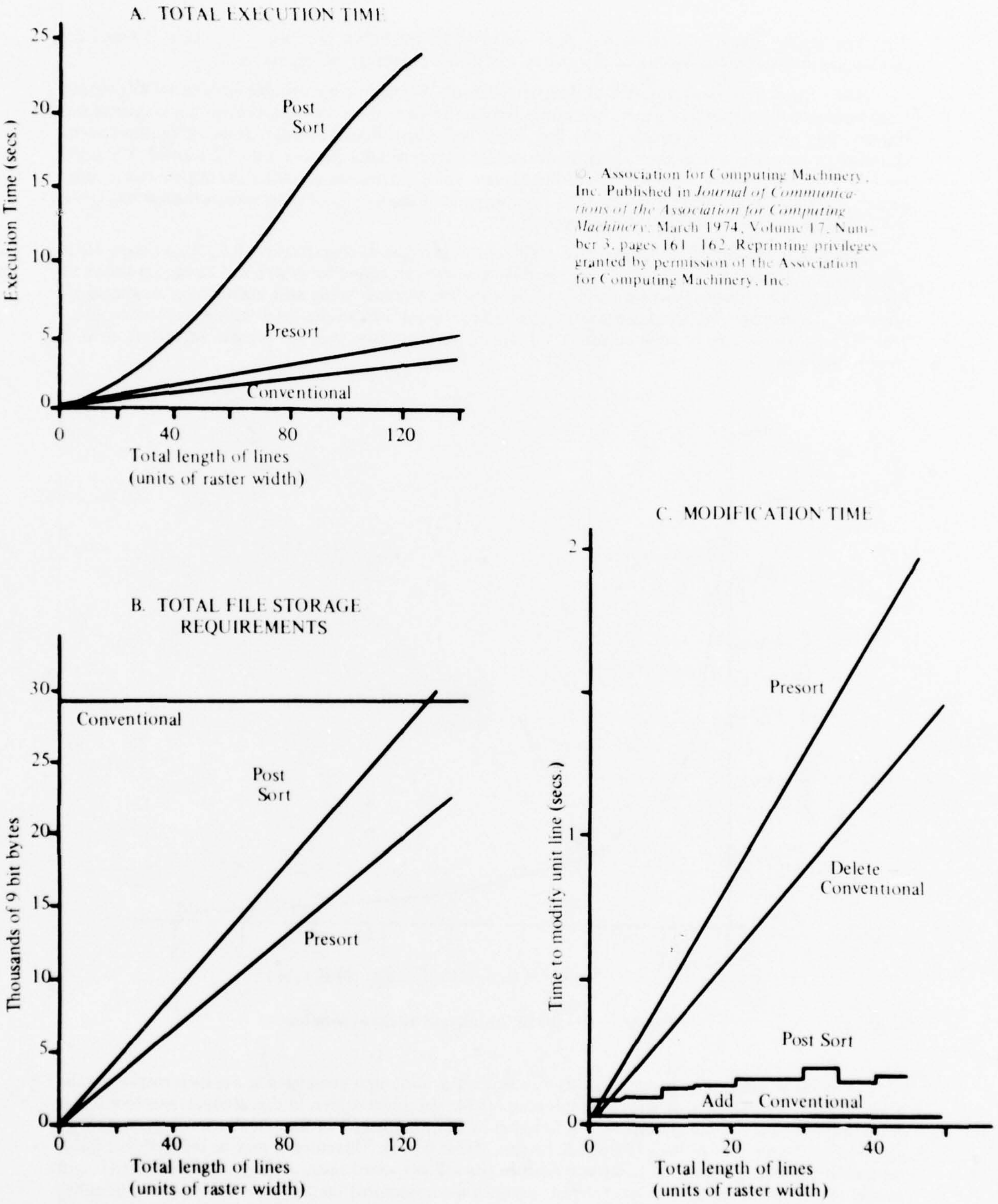


Figure 7. **Summary of key results of Barrett and Jordan experiment.***

*Copied from (Barrett & Jordan, 1974, pp. 161, 162) degrading display quality. Their technique is called visual resolution processing because it is based on eliminating from a scene to be displayed those vectors which cannot be visually resolved.

Use of their technique begins with the establishment of a minimum resolvable element (MRE), which is the length of the diagonal of a rectangle whose sides are equal to the x and y axis resolution factors of the display. Any vector whose length is less than MRE and which does not differ from its predecessor in direction by an angle greater than alpha (alpha usually experimentally chosen) is then discarded. The angle test is required to preserve connectivity of the display. Thus, the process simplifies the display and reduces data by stepping along a given connected line drawing and replacing series of short vectors (having negligible directional differences) with single longer vectors.

Tests were performed on a display of 6 alphabetic and special characters (A,B,C,D,&,@) using MRE values ranging from 0 to .8. Results, which were subjectively evaluated by Bussell and Taxin, are shown in Figure 8. A 78% reduction in the number of vectors was possible while still maintaining an excellent, distortion-free display. The preprocessing required added about 20% to the total vector preparation time. Thus, the technique can be used to effect a tradeoff between time and the number of vectors or lines comprising a scene.

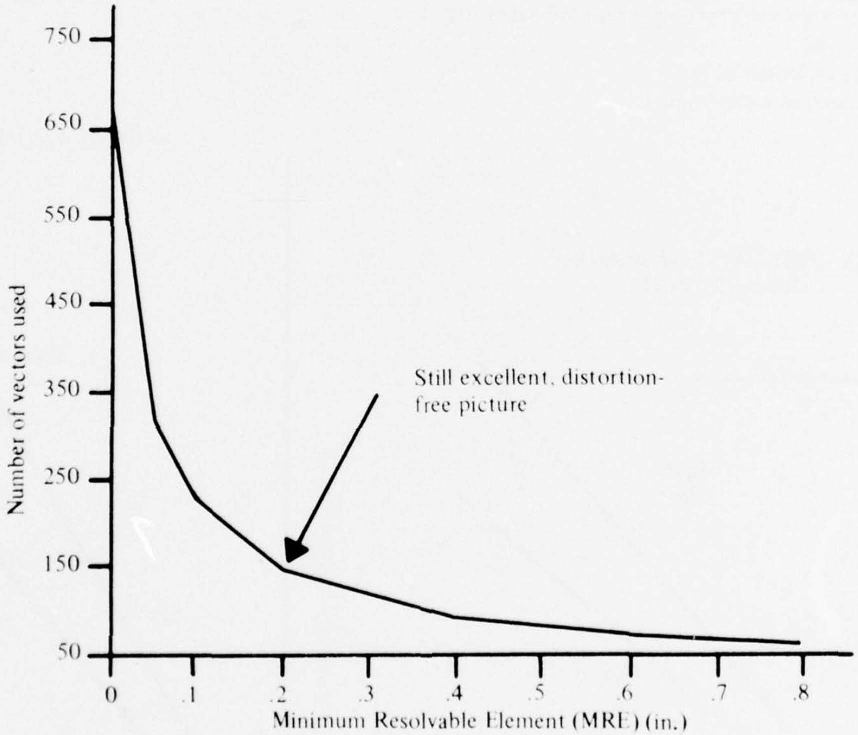


Figure 8. **Results of Bussell and Taxin experiment.**

A point worth noting is the potential effect of visual resolution processing in a system employing the scan conversion algorithms of Barrett and Jordan (1974) described earlier. In that study, it was found that the execution time for the algorithms is a function of the total length of lines in the drawing, but is little affected by the number of lines (Barrett & Jordan, 1974, p.161). Therefore, it may be inferred that visual resolution processing would be of little benefit in the cell organized raster display system of Barrett and Jordan. It would probably have its greatest utility in a conventional random scan CRT system. In either case, however, it could be used to appreciably reduce memory requirements for storing the x-y display image. The real value of visual resolution processing can only by hypothesized on the basis of the relatively simple tests performed (Bussell & Taxin, 1972), and the above remarks should be interpreted accordingly.

(2) *Three-Dimensional Techniques*

Psychologists estimate that on a single dimension men are capable of discriminating 7 plus or minus 2 objects (David, 1967). This is relevant to symbiotic system design because large quantities of data representing system status may need to be transmitted to the man. At the same time, display clutter must be avoided, and display quantity limited to what the man can expectedly discriminate and assimilate. David (1967) points out that multidimensional displays can help considerably in presenting quantities of data in a form interpretable and usable by man. Therefore, 3D techniques are of particular interest.

Newman and Sproull (1973) outline three basic techniques for achieving 3D effects: (a) display of several orthogonal views of the object, (b) perspective displays, or wire frame drawings, and (c) 3D displays involving hidden line and surface removal. These are listed in increasing order of their general desirability. The first two techniques present no real technical problem, and their use is commonly encountered in applications articles in the literature. Ophir, Shepherd, and Spinrad (1969) describe an applied enhancement of perspective displays by making them stereographic as well.

The third technique, for production of 3D displays with hidden line/surface removal, is not at all a simple task and has received considerable attention in the past 10 to 12 years. Research efforts have explored various means of achieving depth cues, including hidden line and surface removal, shading of objects, and variation of texture.[1] The fundamental problem, however, appears to be the efficient removal of hidden parts, because the success of the other techniques is largely dependent on this capability. In 1966 Sutherland identified the hidden line problem as one of ten that were then unsolved in computer graphics. Seven years later, Newman and Sproull (1973) stated that contemporary hidden line and surface algorithms were not optimal, each having its own advantages, disadvantages, and inefficiencies. Therefore, this problem appears to be the most immediately critical to the effective application of 3D techniques in symbiotic systems.

(a) Hidden Line/Surface Techniques

The first practical solution of the hidden line/surface problem was developed in 1963 by L. G. Roberts at MIT Lincoln Laboratory. The Roberts algorithm tests each line in the picture against every opaque surface to discover what parts of it will be visible (Newman & Sproull, 1973; Sutherland, Sproull, & Schumacker, 1973). With minor distinctions in technique, several other algorithms based on the same principle sprang up in the late sixties and early seventies (Appel, 1967; Galimberti & Montanari, 1969; Loutrel, 1970). According to Sutherland et al. (1973), all such algorithms, sharing the requirement that every object be compared with every other object, suffer from long computational time requirements that preclude their use in complex graphics applications. Generally, the computation cost grows as the square of the number of objects in the environment.

The mid-sixties produced a surge of "brute force" techniques characterized by their test, on a line by line basis, of every point comprising every line (Appel, 1967). Their main disadvantage is, again, long computation times even for moderately complex objects (Appel, 1967; Loutrel, 1970). Appel (1967) notes, for instance, that on an IBM 7094, the computation time for a scene consisting of 40 surfaces and 150 lines can exceed 15 minutes! This would clearly be unsatisfactory in any symbiotic system.

According to Wright (1972, 1973), the algorithm most widely used was developed in 1968 by B. Kubert. His method is in many ways a brute force approach but takes advantage of the observation that any line segment from the point of regard to a visible point on the surface contains only the visible point (Kubert, Szabo, & Giulieri, 1968). Line segments directed to invisible points contain more than one point on the surface. By capitalizing on this observation, Kubert's algorithm tests each point for visibility in a relatively efficient way. However it is only applicable to single valued functions of two variables; i.e., $z = f(x,y)$ (Wright, 1972). Speedwise, the computing time on a CDC 6600 grows proportionally to N-cubed, where the surface is represented by an N by N height array (Wright, 1973).

Sutherland et al. (1973) notes that hidden line/surface algorithms may be categorized into two classes, depending on whether they involve a search of the object space (one class) or the image space (the

[1] Newman and Sproull (1973) believe, however, that the depth cueing effects used in computer graphics have been traditionally chosen more for their expediency than for the accuracy of their effect.

second class). Object space algorithms examine each object to discever which parts will be hidden by other parts or other objects. Only then is an image space created from the visible object parts. The image space algorithms, however, concern themselves only with what the image will be at each resolvable dot on the display screen.

The techniques of Roberts and others discussed above (Appel, 1967; Galimberti & Montanari, 1969; Loutrel, 1970) are examples of object space algorithms and suffer from this disadvantage of long computation times. Generally, the image space techniques appear to hold the greater promise.

1. *Image Space Algorithms.* Image space techniques may be classified according to: (a) when, in the order of the various computations, the visibility test is performed; and (b) the order of sorting along the x,y, and z axes (Sutherland et al., 1973). Depth priority techniques postpone the visibility test until last and base it on a computation of the depth of various surfaces as observed from the point of regard. List priority algorithms essentially perform the visibility test first by ordering all surfaces into the form of a list according to their respective depths. Within each type (depth and list priority), data may be sorted in various orders with respect to the three axes, but certain orderings appear to predominate depending on the computational position (order) of the visibility test.

Depth Priority Techniques

Depth priority techniques can be further subdivided into those which sample areas of the screen (area sampling methods) and those which sample points of the screen (scan line methods).

Area Sampling Algorithms. An algorithm developed by J. E. Warnock at University of Utah in 1969 is cited as a representative example of area sampling and is described by Sutherland et al. (1973). The Warnock algorithm is based on a search for homogeneous windows on the image surface, each of which, in their entirety, can be displayed or considered hidden. A window (rectangle in the xy plane) is considered homogeneous if: (a) no faces fall within it at all, or (b) one face completely covers it. In the first instance, there would simply be nothing to display in the window, or perhaps it would be shaded to represent background. In the second instance, the window is displayed if the covering face is nearer the viewpoint than every other face that falls in the window. In the event a window is intersected by a face, it is subdivided into four smaller windows and each is retested. This subdivision continues, as necessary, down to the size of a raster element, terminating earlier only when a window is determined to be homegeneous. Sutherland et al. (1973) points out that a major problem with this method is that it is not easily used with raster scan CRTs, since decisions about various windows are not reached in a left to right, top to bottom order.

Wright (1973) describes a technique in which areas of the screen representing lower boundaries on y are successively created along the z axis. As a search along the z axis proceeds, any point lying below the greatest lower boundary thus far established is declared invisible. Thus, beginning at the point closest to the observer and moving away, no line is drawn which is below a line that has already been drawn. Wright's (1973) method was compared with Kubert's. It was shown that on a CDC 6600, computing time for Wright's technique grows as N-squared (for an N by N height array), whereas Kubert's method incurs time growths proportional to N-cubed. Both of these techniques may be classified as (xy)z sorts, since the xy plane is first considered, followed by a depth search along the z axis.

Scan Line Algorithms. The general philosophy underlying scan line algorithms is the same, regardless of whose particular algorithm is involved (Teixeira & Sallen, 1968). First, edges are sorted by y so that only those edges intersecting the current scan line are examined. Next, a sort along the x axis is performed to determine sections of each scan line within each of which the same face may be visible. Finally, a z sort is performed among all segments falling within each section of a scan line to determine which is visible. Thus, the techniques are characterized as yxz sorts.

Bouknight (1970) describes a line scan algorithm particularly well suited to generation of shaded images. Mechanization of the x sort is performed by first associating a discrete variable (flag) with each polygon comprising the scene. As the scan moves from left to right, the flag for a polygon is flipped each time the scan crosses one of its edges. Only those polygons whose flags are flipped to the "in" position (i.e., scan is within those polygons) are used in distance computations. As each edge is crossed, a decision is made on which of the "in" polygons is closest to the observer, and information is accordingly generated about the intensity of the scan from that point up to the next change point. (Intensity is computed as a function of the angle between the surface of the polygon and a line drawn from the light source to the surface.)

Bouknight ran tests of his algorithm compared with that of Warnock (homogeneous windows method) on a CDC 1604 computer. Results are summarized in Table 2. Bouknight adds that the implementation of his algorithm occupies 30 thousand words of memory, including data storage, and is capable of processing about 1,000 polygons, each with a maximum of four vertices.

Table 2. **Results of Comparison of Warnock and Bouknight Methods**

Figure	Polygons	Edges	Computing Times[a]	
			Warnock	Bouknight
A-Frame Cottage	17	60	2 min.	13.5 sec.
Torus	225	900	8 min.	1 min. 40 sec.
Array of Cubes	160	640[b]	20 min.	2 min.

[a] CDC 1604 computer.

[b] Array was windowed so that only 444 edges were processed.

List Priority Techniques

The first real-time solution to the hidden surface problem was developed at General Electric by R. A. Schumacker (Rougelot, 1969; Sutherland et al., 1973). The technique used a list priority approach to develop a high quality image presentation system for use in visual flight simulation which was implemented at the NASA Manned Spacecraft Center in 1968. Basinger (1973) and Bouknight (1970) report that the NASA system (625 scan lines) was developed for space rendezvous and docking simulation studies; and that the basic system capabilities included up to 240 edges (maximum 40 planes with 6 edges per plane).

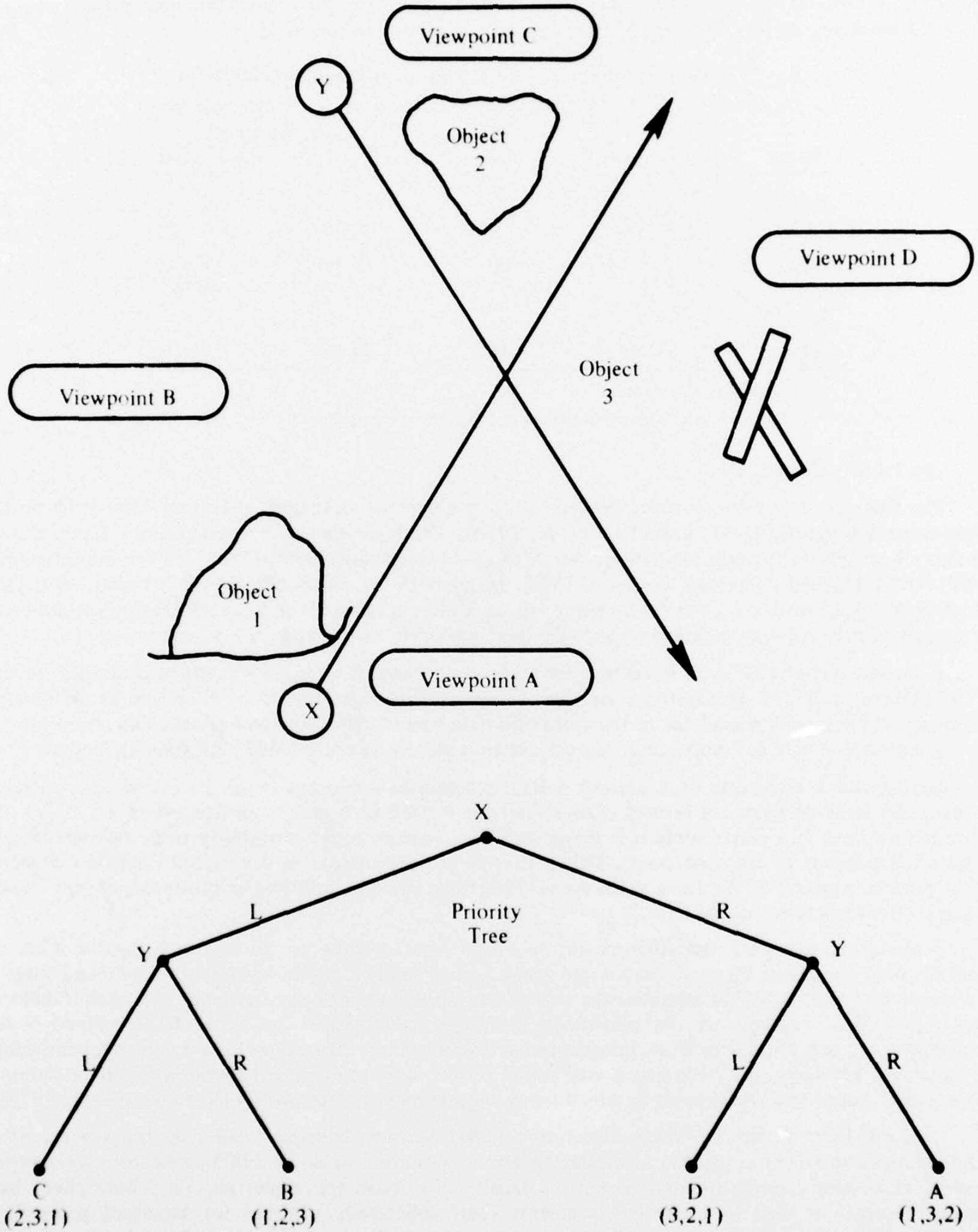
A second system (525 scan lines) was delivered to the Naval Air Training facility at Kingsville, Texas in 1972 (Basinger, 1973). This system is capable of displaying scenes composed of 512 edges. In addition, a 2,000-edge environment is available in the form of a data base implemented on magnetic disk, from which any 512-edge scene may be composed during the system's use in an experimental pilot training program.

Schumacker's technique (zyx sorting order) is based on a concept which he introduced wherein certain collections of faces are termed clusters and are treated as a group (Sutherland et al., 1973). A collection of faces is a cluster only if it permits a preset priority order of visibility to be established for clusters independent of the view point. This permitted an examination of the relations between clusters rather than between all of the faces in the scene. This, then, could be followed by processing of only those cluster sections which are visible.

Schumacker restricted the environment to ensure that clusters are all linearly separable. Cluster priorities were computed by developing a tree whose nodes represent planes separating the various clusters, and whose two node branches represent the side of the plane on which the viewpoint is located. Figure 9 gives a simplified example of the prioritizing technique. Essentially, the Schumacker method is to precompute as much priority ordering information as possible about the environment before ever producing any pictures. Therefore, his technique is well suited to the static environment found in flight simulation, but not well suited to environments in which many objects move independently (Sutherland et al., 1973).

2,000-Edge Computer Image Generation (CIG) System. Basinger (1973) describes a recently developed system which is capable of displaying scenes comprised of up to 2,000 edges for a wide angle field of view pilot training simulation research facility. The basic techniques include Schumacker's list priority method as well as many enhancements that, collectively, provide for increased processing capability. With this system, it is possible to simultaneously display scenes for two independently operated cockpits, each operating anywhere within a 1,250 by 1,250 nautical mile area.

The environment is modeled in the form of a disk-resident data base which incorporates Schumacker's prioritizing information for the various objects (clusters). The data base contains 100,000 edges, expandable to 600,000. Two general purpose digital computers (SEL 86, 32 thousand word core commonly addressable) and one special purpose computer are used to process data from the environment



(Note: L,R in tree denotes whether viewpoint is to left, right of line identified at preceding node point.)

Figure 9. **Simplified illustration of Schumacker's cluster prioritizing technique.**

and produce the required scenes for either or both of two cockpits. One SEL 86 CPU interfaces with memory in the simulation computer to determine *environment features that are in the pilot's field of view*, and perform basic coordinate transformations. The second SEL 86 CPU interfaces with the first via commonly addressable memory and fetches required edge data from the 4.2 M Byte disk containing the environment. This edge data is stored in core as the active data base. The special purpose computer (16 cabinets of hardware housing additional core memory and digital processors) is interfaced with both general purpose computers via their common memory, and accomplishes the actual generation of video signals.

Hidden faces are determined by *scanning the environment* with a vector extending from the viewpoint to the various faces of objects, and comparing its angular orientation with that of a unit vector normal to the surface of the face. Edges common to adjacent faces that are both hidden are also hidden. This effects a rapid elimination of hidden edges and is a key feature in accomplishing the processing in real time (Basinger, 1974). Another key feature is included in the scan conversion process itself. Inputs to the converter include the raster line at which each edge first and last appears, and the slope of the edge in the display plane. This permits edges to be efficiently scan converted by building their binary image recursively. When conversions are accomplished for scan line n, elements of edges which have not yet terminated are computed for scan line n+1 using the slope information (Beck, 1974).

The CIG system drives fourteen 36 inch CRT displays, seven arranged around each cockpit to effect a ± 120° horizontal by +120°, –40° vertical field of view for the pilot. It is part of a simulation complex developed under the direction of the Air Force Human Resources Laboratory's Advanced Systems Division, Wright-Patterson Air Force Base, Ohio, for use in training simulation research. It is obvious that the impressive CIG processing capability is largely attributable to an appreciable amount of computing power devoted exclusively to that task. Thus, while some of the various techniques it employs may merit extension and further investigation, the cost of a large-scale implementation such as this would preclude its *widespread use in man-computer symbiotic systems*.

2. *Additional Observations.* The review presented of hidden line/surface techniques does not nearly exhaust the vast bibliography on the subject nor cover the multitude of methods that have been tried to develop improvements here and there on the alternative types of algorithms. However it is believed to have fulfilled its major objective of outlining the basic approaches that are available and some of the principal problems encountered with them.

Sutherland et al. (1973) concluded their survey with the observation that sorting is, in essence, the major problem of hidden surface detection. The sorting order, in particular, may represent a yet-untapped framework from within which new methods may be developed and explored. (That new methods are required is fairly obvious in view of the reported computation times, representatives of which were discussed earlier.) The sorting orders used in the image space algorithms were:

Area Sampling – (xy)z
Scan Line – yxz
List Priority – zyx

Sutherland suggests that one avenue for new research may be to try other sorting orders.

It seems, too, that perhaps an optimal sorting order does not exist per se, but depends on the nature of the environment. The major result of sorting is to eliminate information, and the density of information along the various axes could be used to identify that axis along which the sorting process will eliminate the greatest amount of information. Therefore, it may be worthwhile to investigate the use of variable sorting orders which depend on the display environment of the moment.

(3) *Review and Conclusions*

The symbiotic system requirement leading to the above survey of graphics techniques was that both man and computer require techniques for formatting and presenting information (which may be unanticipated by the receiver) in a way that will be clearly understood. Both 2D and 3D graphics techniques were reviewed in light of this requirement.

(a) 2D Techniques

Basic 2D operations such as rotation, scaling, zooming, etc, pose no distinct technical problems, as methods for accomplishing these operations are relatively straightforward and have been proven. Pen-tracking techniques are readily available; however, the ever present possibility of losing the pen's position if it is moved too rapidly remains a problem that has been handled so far only by means of a systematic search of the screen.

Scan conversion is normally performed using software due to the complexity of analog equipment for implementing a hardware capability. The conventional method of scan conversion (producing a direct binary image of the display) has three associated problems: (a) large storage requirement for the binary image, (b) large computational times, and (c) difficulty in picture modification. Techniques to reduce storage requirements are accompanied by an increase in computation time; however, the potential percent reduction in storage is about 8 times the corresponding percent increase in time. Efforts to enhance picture modification capability are based around cell organization of the display and placement of the refresh memory in front of the character/vector generator. Associated scan conversion techniques have been developed for the cell organized display. The technique seemingly best suited for interactive graphics applications achieved a significant reduction in the time required to delete lines during picture modification. However, its general processing time was on the order of 4 to 8 times greater than that of previously developed techniques.

A unique method of significantly reducing the number of vectors comprising a scene has been developed (visual resolution processing). However the method would have its greatest utility in random scan CRT systems. In raster scan systems, it could be used to reduce the size of the x-y data base but would not favorably affect computational time.

(b) 3D Techniques

The most significant problem in the use of 3D techniques appears to be the removal of hidden lines and surfaces. The earliest techniques developed as well as the techniques most widely used today employ object space algorithms which examine each object to discover which parts will be hidden. Only then is a picture created by deriving an image space or display plane image. These techniques all suffer the disadvantage of long computation times.

Techniques employing image space algorithms are characterized by their attention to the nature of the display image at each resolvable dot on the screen, rather than to the total object underlying the image. Depth priority image space methods postpone the visibility test until last, and perform it on the basis of surface depth as observed from the viewpoint. These methods perform their tasks by operating on successive screen areas or on successive scan lines. The Warnock area sampling method is not easily used with raster scan CRTs because of the unsystematic processing order of screen areas. The Wright area sampling method has a significant speed advantage over earlier techniques, but at that, the computational time grows as N-squared, where an N by N height array is involved. Scan line techniques are characterized by a systematic yxz order data sort to: (a) limit attention to the current scan line, (b) determine sections of the scan line within which each face may be visible, and (c) determine which faces are visible based on their proximity to the viewpoint. These techniques appear to be generally faster than area sampling methods, but they are memory consuming and still require minutes of computer time for single, moderately complex objects.

List priority image space techniques perform a portion of the visibility test first by generating a list of surfaces ordered according to their respective depths. Schumacker's method is the best known and has been applied in several large systems to provide visual simulation for crew training. While achieving the real time solutions required in such applications, the method is not well suited to environments in which many objects move independently. Large-scale applications, such as that described by Basinger (1973), require extensive computing power devoted exclusively to display generation to accomplish the job in real time.

(c) Conclusions

Scan conversion methods are generally too time consuming or too memory consuming to be efficiently used in complex display generation for man-computer symbiotic systems. Efforts to reduce time or memory have only resulted in a tradeoff between the two. Hardware scan conversion appears to provide the only readily available solution. However, the additional cost and potential maintenance problems make this solution far less than optimal.

The major problem in achieving 3D effects appears to be the elimination of hidden lines and surfaces. A tremendous amount of work has been performed in attempts to efficiently resolve this problem. The image space techniques appear to hold the greatest promise from the standpoint of minimizing computational time. However, they seem to accomplish this time reduction by sacrificing flexibility (e.g., they lack ability to handle environments in which many objects move independently). Present methods of 3D display, with hidden parts removed, are generally time consuming and applicable only to the display of simple scenes in a man-computer symbiotic system. Promising areas of research may lie in further variation of sorting orders (with respect to the x,y, and z axes) and in making the sorting order dependent on the image space environment rather than fixing it at the onset.

Language. Work on developing an efficient language for graphics applications has included the development of programming languages and small-scale voice input techniques, with the former being several orders of magnitude more prevalent.

(1) *Graphic Programming Languages*

There is nearly a one to one mapping from the set of graphic programming languages onto the (sizeable) set of interactive graphics applications. A survey of the entire area would have easily resulted in an additional paper nearly equal in length to this one. Therefore, what follows is more a general impression than a survey, the latter being necessarily deferred as a possible sequel to the effort being reported.

Graphic programming languages seem to be classifiable on the .basis of the strength of their association with FORTRAN or some other high level language. The number of languages per class is greatest for those classes having strongest FORTRAN associations, dwindling rapidly as that strength lessens and finally disappears. This is undoubtedly due to the (commendable) goal of producing a universal graphic language, which is partly achievable by basing it upon a universal high level language already in existence.

FORTRAN subroutine packages are probably the most widely used of graphic languages (though strictly speaking a subroutine package cannot properly be considered a language). Typical of these are GSP (Rully, 1968), DISPLAYTRAN (Gagliano, Thombs, & Cornish, 1968), and ARTA (Mezei & Zivian, 1972). Use of such packages simply involves preparation of a FORTRAN main program which calls various graphics subroutines as required. The subroutines are, necessarily, graphic device dependent. In addition, long argument lists make use of the subroutines cumbersome.

Attempts to minimize requirements for frequent subroutine calls with long calling vectors lie primarily in the development of FORTRAN extensions and additions (Bracchi & Somalvico, 1970; Bracchi & Ferrari, 1971; Dell' Aquila, Murro, Prezioso, & Refice, 1972; Frank, 1968; Hurwitz, Citron, & Yeaton, 1967; Shearing, 1971) This includes the development of such FORTRAN enhancements as the definition of graphic variables and extension of the language itself to include such commands as DRAW, ERASE, and EXPAND. Considerable dependence on subroutine calls still exists but is lessened by such language enhancements.

Lecarme (1972) points out several weaknesses of graphic languages with strong FORTRAN associations. He says that calling vectors for subroutine calls often contain 10 and sometimes up to 39–53 parameters. Also, the position of the parameters in the calling vector is significant, and awkward techniques are required to omit parameters when desired. He also claims that many subroutines themselves are anonymous under their "foreshortened names" and difficult to use if only for that reason.

These observations have led to the development of a number of special purpose machine-dependent graphics languages having little or no association with FORTRAN and little or no commonalities among themselves (Hormann et al., 1971; Lecarme, 1972; Newman, 1973; Sibley, Taylor, & Gordon, 1968). One such language, for example, is LOGO (Newman, 1973), which consists of a set of macros typified by

LINETO	x,y	
DISPLAY	s	s is predefined subroutine
DRAW BOX IN	x_1,y_1,x_2,y_2	

Another is UAL (Hormann, Leal, & Grandell, 1971), which is list oriented, permitting assignments resulting in pointer changing and value changing, as well as having its own unique set of macros.

The overriding impression gained from a cursory review of existing graphic programming languages is that those attempting to be general purpose are far less than optimal, while those which have more extensive flexibility are application dependent and would need to be reprogrammed for use in a given man-computer symbiotic system. Also, as with most programming languages, considerably more attention has been paid to the efficacy of the language in the eyes of the computer than in the eyes of the man. Thus, most appear potentially cumbersome, slow, and unnatural to use in a system purporting to produce a symbiotic relationship between man and machine. Perhaps equally important, graphic programming languages per se seem highly unsuited to symbiotic system use where attention getting and rapid interaction may occasionally be required.

(2) *Small-Scale Voice Input Techniques*

Since most computers are deaf, the simplest and most common practice is to treat man as a mute, as in the case of graphic programming languages. Large-scale (i.e., unlimited vocabulary) voice input capabilities are certainly a potential tool in man-computer symbiotic systems, but at least for the very near future, the state of the art does not suggest their widespread, low-cost availability. However, several fairly successful attempts have been made to provide a limited voice input capability that is well within the state of the art and ideally suited to man-computer symbiotic systems.

Small-scale voice input techniques perform four basic functions. First, a speech processor (usually analog) extracts signals from the input waveform that will be used to characterize the utterance. Bagley (1972) for example extracted three measures: (a) a measure of signal level, proportional to short-term average power; (b) a measure of zero crossing rate of the signal; and (c) a measure of zero crossing rate of the first time derivative of the signal. Elder (1970) used a group of bandpass filters to extract time-sampled measures proportional to the power output of each filter (similar to Fourier analysis).

The second task is to recognize when a word or utterance has been spoken. This requires imposition of a restriction that a moment of silence occur between each word so that word boundaries may be detected (Elder, 1970). Then the recognition algorithm, testing speech samples continuously, scans the amplitude signal and waits for it to rise above, then fall below, some fixed threshold to detect the utterance boundaries.

The third task is to model an utterance when its occurrence has been recognized. Elder's technique is to store an A_{ij} array where each entry is proportional to the power output of filter i at time instant j. Bagley forms a 22-dimensional characteristic vector by breaking each utterance into 7 equal length blocks and averaging each of the 3 measures within each block. A 22nd element is computed which is proportional to the duration of the utterance.

The fourth task is to decide which, of a limited number of utterances, has been spoken. This decision task would be formidable were it not for the imposition of restrictions on vocabulary size and training the system to recognize only known speakers. These are the two major restrictions which distinguish small-scale voice input capabilities from their general purpose, large-scale counterparts.

(a) Training the System

The basic task in training the system is to first compile reference models for each utterance. In application, each reference model is compared with the current utterance model and the decision task becomes one of determining the 'closest fit.'

System training is the compilation of the reference models. One technique is to compute a separate set of reference models for each speaker and store them in some form of mass storage. This is the method used by Elder (1970). Then, when a previous user approaches the system, his reference arrays are fetched for use in the decision task. Training is performed by asking each new user to recite the vocabulary set several times, and computing a single model of each utterance by using an averaging method.

Another method, used by Bagley (1972), is to compute a single set of reference models for all speakers. This is accomplished by recursively building a prototype vector dictionary consisting of characteristic vectors (utterance models) for each utterance obtained from various speakers. During training, this prototype vector dictionary is used in a closest-match scheme to guess which utterance has been spoken. If the guess is correct, no updating is performed on the dictionary. If incorrect, the characteristic vector for the utterance is added to the dictionary and testing/training continues. This is continued up to some preset limit on the dictionary size.

(b) The Decision Process

Deciding which utterance has been spoken requires the computation of a measure of closeness between the model or characteristic vector for the input and the reference models or vectors. The typical approach is to compute a summed-square difference error over the respective components of the two vectors being compared (Bagley, 1972; Elder, 1970). Other possibilities are to form a scalar or dot product or an absolute difference error. A disadvantage of this technique is that it requires computation of the selected measure for every reference model vector.

Elder (1970) proposes (but did not implement) a technique to reduce this computational load by limiting the number of reference models that are compared with the model for the unknown utterance. He suggests that a prominent-feature comparison be performed first to eliminate reference vectors that differ markedly from the vector under test. Such features might include the real-time length of the utterance or the number of peaks occurring in the power-versus-time curve for an utterance.

(c) Experimental Results

Elder's technique (unique reference models for each speaker) resulted in a 3.6% error in attempting to correctly identify words from a vocabulary size of 37. The test included 3 utterances of each word in the vocabulary. In an attempt to reduce error, a modified decision scheme was contrived wherein the two reference arrays closest to the unknown's array were determined. If these two arrays were very similar (as determined by their summed square difference error), then the utterance was "rejected," and no attempt was made to identify it. The theory was that in this situation, a slight perturbation in the manner of speaking could have resulted in a different identification. With this scheme, errors were nonexistent but 36% of the utterances were rejected. It was concluded that the modified scheme has little practical utility.

Experiments were also run by Elder to determine the effect on error frequency of reducing the number of filters used to obtain basic model variables for the utterances. In the original configuration, 16 filters were used, with crossover frequencies ranging from 100 to 4,100 Hz. Reducing the number of filters to 8, while retaining the same bandwidth, resulted in an increase of errors from 3.6% to 6.3%.

Bagley's technique (single set of reference models for all speakers) resulted in error frequencies which depended on the number of utterances made by each speaker during training. When applied to a speaker who had been involved in the training process, errors ranged from about 10% for 3 repetitions of each word in training, to 2% for 30 repetitions. The vocabulary size was 33. When applied to a speaker who had not been involved in the training process, the error frequency was 14%. This figure was reduced to 2% by selectively dropping from the vocabulary 8 words which had the highest associated error rates.

Both methods were implemented on an IBM 360/65 computer. Response time for the Elder method was about 185 Ms, and that for the Bagley method was 215 to 240 Ms. For core storage of the reference models, Elder's method required 9,472 bytes for the 37 word vocabulary, and Bagley's method required 4,400 bytes for the 33 word vocabulary.

(d) Observations

It appears that a distinct time/memory tradeoff exists between the techniques of modeling each speaker independently (Elder, 1970) and compiling an overall model for all speakers (Bagley, 1972). The former is faster but consumes about twice the core for storing the reference model. With adequate training, Bagley's method achieves a slight increase in accuracy over Elder's, but this observation must be weighed in light of the differing vocabulary sizes used in the two experiments. A distinct advantage of Bagley's method is its usability with unknown speakers. This can be performed with an accuracy equal to that achieved for known speakers if the vocabulary set can be judiciously reduced to eliminate the most troublesome words.

(3) *Review and Conclusions*

A review of the language aspect of computer graphics was inspired by the symbiotic system requirement for an unrestrictive and easily applied means of "getting the point across" in transmitting information that is relevant to system performance. Along with this was the companion requirement for some attention-getting capability.

Graphic programming languages are abundant and varied, but the most prevalent are those which have strong associations with FORTRAN or some other high level language. These include FORTRAN subroutine packages as well as various additions to and extensions of FORTRAN. Partly in an attempt to avoid the awkwardness of using many subroutine calls with lengthy calling vectors, efforts have also been devoted to developing special purpose languages that do not depend on an association with a higher level language. These types of languages are largely application specific and highly machine dependent.

Graphics programming languages are geared primarily to adaptation of the man to the computer and, in their use, man is denied the advantage of his most natural communication mode—speech. However, a few promising techniques exist which may be immediately applied in man-computer symbiotic systems for voice input capabilities. On a small scale, limited vocabulary voice input techniques have been experimentally developed with as little as 2% error in recognition accuracy. The principal variants of these techniques are the method of modeling the input waveform and the method of developing a reference set of models (training the system). Response time and memory tradeoffs exist largely as a function of whether each known speaker has his own reference set of models, or a single model exists for all known speakers. Accuracy is not significantly affected by these alternatives if sufficient time and trials are devoted to training the system.

In conclusion, graphic programming languages are probably adequate for communications that: (a) are anticipated by the transmitter, and (b) do not require speed. However, as with nearly any programming language, they are at best a communication aid that must be learned and which necessitates planning and time to apply. For this reason, they do not, by themselves, fulfill all language requirements for a symbiotic system, where both attention getting and extensive flexibility may be required to transmit information critical to on-going system performance.

An extremely promising adjunct to a symbiotic system is a small-scale voice input capability. This would provide a highly efficient attention-getting capability for one thing, and possibly could be used in conjunction with a graphic programming language to provide an effective overall means of communication. Hill (1971) cites several other advantages of such a capability, including additional reliability and its naturalness to the man. Equally important is the potentially favorable effect that it could have on the man's attitude about the system and his rapport with the computer. A small vocabulary size does not appear to be a serious limitation if the vocabulary were judiciously selected. However error rate may be critical, because use of voice input would have its greatest utility in situations where speed and accuracy are both relevant. (For instance, mistaking "No" for "GO" at a critical point in system operation could be disastrous.) The most fruitful areas of research in the overall language area seem to be: (a) reducing error rate in small-scale voice input techniques, and (b) combining graphic programming languages and voice input techniques into a single efficient communication capability.

IV. OVERVIEW AND CONCLUSIONS

The purpose of this paper was to determine and survey the research areas that appear most critical to achieving man-computer symbiosis. First, an operational definition of man-computer symbiosis was derived which distinguishes it from other types of man-computer relationships. This was accomplished by viewing such relationships from a systems standpoint and characterizing them by the types of information or data that are transceived by man and computer. The primary distinguishing feature of a symbiotic system is the capability of both parties to transceive relevant unrequested information.

Using the derived definition, key system requirements were projected. These included status transceivers and memories, graphic techniques, a language, and an attention-getting capability. A survey was then performed of research and the state of the art of technology addressing each of these requirements. Critical problems were identified during the course of the survey, and emphasis was placed on a review of research areas related thereto.

Status Transceivers and Memories

It was concluded that raster scan CRTs hold the greatest promise for symbiotic system applications due mainly to the reduced refresh rates required when using an interlaced scan. This provides a capability for transmitting larger quantities of data without incurring flicker in the display, which is a major source of trouble in many graphics applications. While raster scan CRT technology itself poses no specific research problems, requirements for efficient scan conversion methods do.

Input devices are many and varied, but there is no single device that is low cost and optimal for pointing, drawing, and positioning data on the CRT. These are relatively significant capabilities to develop for man-computer symbiosis because pointing, drawing, and positioning are the major actions by which man can communicate with the computer using the display. The light pen is a relatively flexible input device, but faster response times are needed. It is also desirable, from the standpoint of convenience to the human, to provide a stylus-independent input device. However, the poor quality of achievable resolution is a major problem here.

Despite a lot of evidence as to its need, little work has been accomplished in efficiently extending man's memory precision and recall capability. The most promising approach seems to be to provide him a simulated short-term memory. Only one study was found which attempted to do this in an interactive graphics application. It was application specific, however, and not directly generalizable to man-computer symbiotic systems. Efforts to extend computer memory capacity have mainly consisted of improved data structures to enhance retrieval speed. There is some evidence that perhaps a better direction of effort would be to reduce the number of display files typically required.

Graphics Techniques

The major research problems existing in the area of graphics techniques are the development of efficient scan conversion methods and techniques for solving the hidden line/surface problem. Existing scan conversion methods are either too time consuming or too memory consuming. Attempts to improve on one of these aspects have resulted in tradeoffs on the other. Hardware scan conversion is possible, but only at a high cost and with highly complex electronic equipment which detracts from overall system reliability. The only real-time solution to the hidden line/surface problem requires enormous processing power for a large-scale implementation, and is not suitable for use with environments where everything moves independently. Other solutions are extremely slow in general, and, again, speed/memory tradeoffs have occurred in every attempt at improvement.

Tracking the light pen continues to be a problem due to the possibility of the tracking program getting lost if the pen is moved too rapidly. Since the light pen is one of the lowest cost input devices available and is fairly flexible, its use in near-future symbiotic systems is anticipated. Therefore, pen tracking may be considered one research problem relevant to symbiotic system development.

Even though raster scan CRTs hold greater promise than do random scan types due to the flicker problem, they do have the companion requirement for scan conversion. One way of reducing the flicker probability in random scan vector drawing systems is to reduce the number of vectors needed to compose a scene. Visual resolution processing seems to be one technique for accomplishing this. It certainly merits further investigation to determine whether or not its full-scale use in conjunction with random scan systems can produce flicker-free displays whose densities approach those possible with raster scan systems.

Language

The most promising research area here is the further development and refinement of small-scale voice input techniques. Improved accuracy seems to be the most significant requirement. This would not only provide a highly efficient attention-getting capability, but would enhance the rapport which man feels with the computer by allowing him to use his most natural communication mode. Another promising area is to investigate hybrid communication techniques using a combination of graphic languages and small-scale voice input methods. This could offset some of the disadvantages of graphic languages (cumbersome, slow, and require training the person) by complementing their use with the speed and naturalness of speech.

Summary of Critical Research Areas

Table 3 provides a summary of the research areas determined to be most critical. Advancements in these areas would effect the most rapid closure on achieving the type of system required for man-computer symbiosis. The interesting and, in a way, discouraging observation that evolves from Table 3 is that many of these research areas are not new and, to one extent or another, have already been investigated for about a decade. In view of this, one wonders how much can be expected in way of further improvements in the next decade.

On the more optimistic side, many of the areas have enjoyed some very recent advancements (past 2 to 4 years) which, if not optimal themselves, have suggested new avenues of investigation. This is especially true of the areas of scan conversion, hidden line/surface, and voice input. Still other areas (simulated STM, visual resolution processing, and the idea of hybrid communication techniques) are, as yet, relatively unexplored and therefore wide open as far as potentially bridging the gap to man-computer symbiosis.

This paper has determined and reviewed what appear to be the most fruitful areas in which we can channel our next efforts. Hopefully, therefore, it has contributed to the goals of man-computer symbiosis by serving as a point of departure and possible inspiration to existing or potential aspirants to those goals.

Table 3. **Summary of Critical Research Areas**[a]

Status Transceivers and Memories	Graphic Techniques	Language	Attention-Getting Capability
1. Raster scan CRTs & associated software	1. Scan conversion	1. Small-scale voice input techniques (improved accuracy)	1. Small-scale voice input techniques
2. Simulated STM	2. Hidden line/ Surface techniques	2. Hybrid communication methods (voice input + graphic language)	
3. Low cost, general purpose input devices	a. Sorting Order		
	b. Variable sorting order		
a. Light pen response time	c. Image space techniques		
b. Stylus-independent devices (resolution)	3. Visual resolution processing with random scan CRTs		
4. Reduction of number of display files	4. Pen tracking		

[a] In each category, research areas are listed in descending order (author's opinion) of criticality.

REFERENCES

Appel, A. The notion of quantitative invisibility and the machine rendering of solids. *Proceedings of the 22nd National Conference of the Association for Computing Machinery,* Washington, D.C.: Thompson Books, 1967, 387–393.

Bagley, J.D. A voice input facility for digital computers. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society,* London: North Holland Publishing Company, 1972, 680–685.

Barrett, R.C., & Jordan, B.W. Scan conversion algorithms for a cell organized raster display. *Journal of Communications of the Association for Computing Machinery,* 1974, **17**(3), 157–163.

Basinger, J.D. An approach to computer image generator for visual simulation. AIAA Paper No. 73-928. Paper presented at the meeting of the American Institute of Aeronautics and Astronautics Visual and Motion Simulation, Palo Alto, California, 1973.

Basinger, J.D. Personal communication, 1974.

Beck, J.W. Personal communication, 1974.

Bekey, G.A. The human operator in control systems. In K.B. DeGreene (Ed.), *Systems psychology.* New York: McGraw Hill, 1970, 248–276.

Bouknight, W.J. A procedure for generation of three dimensional half toned computer graphics presentations. *Journal of Communications of the Association for Computing Machinery,* 1970, **13**(9), 527–536.

Bracchi, G., & Somalvico, M. An interactive software system for computer aided design: an application to circuit project. *Journal of Communications of the Association for Computing Machinery,* 1970, **13**(9), 537–545.

Bracchi, G., & Ferrari, D. A language for treating geometric patterns in a two dimensional space. *Journal of Communications of the Association for Computing Machinery,* 1971, **14**(1), 26–32.

Burton, R.P., & Sutherland, I.E. Twinkle box – a three dimensional computer input device. *Proceedings of the American Federation for Information Processing Society* (Vol. 43). Montvale, NJ: AFIPS Press, 1974, 513–520.

Bussell, B., & Taxin, H. Visual resolution processing for interactive graphics. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society* (Vol. 1). London: North Holland Publishing Company, 1972, 748–751.

David, E.E. Physiological and psychological considerations. In W.J. Karplus (Ed.), *On-line computing.* New York: McGraw Hill, 1967, 107–128.

Davis, S. *Computer data displays.* Englewood Cliffs, NJ: Prentice Hall, 1969.

Davis, M.R., & Ellis, T.O. The rand tablet: a man-machine graphical communication device. *Proceedings of the Fall Joint Computer Conference.* Baltimore, MD: Spartan Books, 1964, 325.

DeGreene, K.B. Man-computer interrelationships. In K.G. DeGreene (Ed.), *Systems psychology.* New York: McGraw Hill, 1970, 281–336.

Dell'Aquila, C., Murro, O., Prezioso, G., & Refice, M. A new interactive graphic language. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society* (Vol. 1). London: North Holland Publishing Company, 1972, 445–449.

Elder, H.A. On the feasibility of voice input to an on-line computer processing system. *Journal of Communications of the Association for Computing Machinery,* 1970, **13**(6), 339–346.

Engelbart, D.C., & English, W.K. A research center for augmenting human intellect. *Proceedings of the American Federation for Information Processing Society, Fall Joint Computer Conference* (Part 1), 1968, 395–410.

Fitts, P., & Posner, M. *Human performance.* Belmont, CA: Brooks/Cole Publishing Company, 1969.

Frank, A.J. B-Line, bell line drawing language. *Proceedings of the American Federation for Information Processing Society, Fall Joint Computer Conference* (Part 1), 1968, 179–191.

French, L.J., & Teger, A.H. GOLD – a graphic on-line design system. *Proceedings of the American Federation for Information Processing Society, Spring Joint Computer Conference* (Vol. 40). Montvale, NJ: AFIPS Press, 1972, 461–470.

Gagliano, F.W., Thombs, H.W., & Cornish, R.E. A conversational display capability. *IBM Systems Journal on Interactive Graphics in Data Processing,* 1968, 7(3,4), 281–291.

Galimberti, R., & Montanari, U. An algorithm for hidden line elimination. *Journal of Communications of the Association for Computing Machinery,* 1969, **12**(4), 206–211.

Gould, J.D. Visual factors in the design of computer controlled CRT displays. *Human Factors,* 1968, **10**(4), 359–375.

Hill, D.R. Man machine interaction using speech. In F. Alt & M. Rubinoff (Eds.), *Advances in computers* (Vol. 11). New York: Academic Press, 1971, 165–230.

Hlady, A.M. A touch sensitive X-Y position encoder for computer input. *Proceedings of the American Federation for Information Processing Society, Fall Joint Computer Conference* (Vol. 35). Montvale, NJ: AFIPS Press, 1969, 545–551.

Hormann, A., Leal, A., & Grandell, D. *User adaptive language (UAL): a step toward man-machine synergism.* TM 45 39/000/01. Santa Monica, CA: System Development Corporation, 1971.

Hurwitz, A., Citron, J., & Yeaton, J. GRAF: graphic additions to FORTRAN. *Proceedings of the American Federation for Information Processing Society, Spring Joint Computer Conference* (Vol. 30). Montvale, NJ: AFIPS Press, 1967, 553–557.

Record of the 1972 conference on display devices. Institute of Electrical and Electronics Engineers Conference, October 11–12, 1972.

Jacks, E.L. Observations on a graphic console system. In D. Secrest & J. Nievergelt (Eds.), *Emerging concepts in computer graphics.* New York: W.A. Benjamin, Incorporated, 1968, 131–170.

Jordan, B.W., & Barrett, R.C. A scan conversion algorithm with reduced storage requirements. *Journal of Communications of the Association for Computing Machinery,* 1973, **16**(11), 676–682.

Jordan, B.W., & Barrett, R.C. A cell organized raster display for line drawings. *Journal of Communications of the Association for Computing Machinery,* 1974, **17**(2), 70–77.

Kleiman, E.B. A survey of various information display devices. In *Data displays for information systems.* Detroit, MI: American Data Processing, Incorporated, 1969, 17–37.

Kubert, B., Szabo, J., & Giulieri, S. The perspective representation of functions of two variables. *Journal of the Association of Computing Machinery,* 1968, **15**(2), 193–204.

Lecarme, O. A system for interactive graphic programming. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society.* London: North Holland Publishing Company, 1972, 440–444.

Licklider, J.C.R. Man-computer symbiosis. *Proceedings of the Institute of Radio Engineers Transactions on Human Factors in Electronics* (HFE-1), 1960, 4–11.

Licklider, J.C.R. *The system-system and bridges over the gulf between man-machine system research and man-machine system development.* AFOSR 1673 and AFOSR 1127 (published as a single report). Cambridge, MA: Bolt Beranek & Newman, Incorporated, 1962. (NTIS No. AD-424 284).

Loutrel, P. A solution to the hidden line problem for computer drawn polyhedra. *Proceedings of the Institute of Electrical and Electronics Engineers Transactions on Electronic Computers,* 1970, EC-19(3), 205–213.

Machover, C. Computer graphics terminals – a backward look. *Proceedings of the American Federation for Information Processing Society, Spring Joint Computer Conference* (Vol. 40). Montvale, NJ: AFIPS Press, 1972, 439–446.

Matsa, S.M. What has computer graphics to offer? In R.D. Parslow, R.W. Prowse, & R.E. Green (Eds.), *Computer graphics.* New York: Plenum Press, 1969, 1–6.

Mezei, L., & Zivian, A. ARTA, an interactive animation system. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society* (Vol. 1). London: North Holland Publishing Company, 1972, 429–434.

Miller, R.B. *Psychology for a man-machine problem solving system.* TR 00.1246. Poughkeepsie, NY: International Business Machines, 1965. (NTIS No. AD-640 283).

Newman, W. Display procedures. *Journal of Communications of the Association for Computing Machinery,* 1971, **14**(10), 651–660.

Newman, W. An informal graphics system based on the LOGO language. *Proceedings of the American Federation for Information Processing Society* (Vol. 42). Montvale, NJ: AFIPS Press, 1973, 651–655.

Newman, J.R., & Rogers, M.S. *Experiments in computer aided inductive reasoning.* TM 3227. Santa Monica, CA: Systems Development Corporation, 1966.

Newman, W., & Sproull, R. *Principles of interactive computer graphics.* New York: McGraw Hill, 1973.

Noll, A.M. Scanned display computer graphics. *Journal of Communications of the Association for Computing Machinery,* 1971, **14**(3), 143–150.

Ophir, D., Rankowitz, S., Shepherd, B.J., & Spinrad, R.J. BRAD: the brookhaven raster display. *Journal of Communications of the Association for Computing Machinery,* 1968, **11**(6), 415–416.

Ophir, D., Shepherd, B., & Spinrad, R. Three dimensional computer display. *Journal of Communications of the Association for Computing Machinery,* 1969, **12**(6), 309–310.

Pobgee, P.J., & Parks, J.R. Applications of a low cost graphical input tablet. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society* (Vol. 1). London: North Holland Publishing Company, 1972, 737–741.

Rougelot, R. The general electric computed color TV display. In M. Faiman & J. Nievergelt (Eds.), *Pertinent concepts in computer graphics.* Urbana, IL: University of Illinois Press, 1969, 261–281.

Rully, A.D. A subroutine package for FORTRAN. *IBM Systems Journal on Interactive Graphics in Data Processing,* 1968, **7**(3,4).

Rundle, A. Interactive software techniques. In R.D. Parslow & R.E. Green (Eds.), *Advanced computer graphics.* New York: Plenum Press, 1971, 29–39.

Salminger, P. Experimental display with input by light gate field selection. *Proceedings of the 1971 Conference of the International Federation for Information Processing Society* (Vol. 1). London: North Holland Publishing Company, 1972, 742–747.

Semple, C., Heapy, R., Conway, E., & Burnette, K. *Analysis of human factors data for electronic flight display systems.* AFFDL-TR-70-174. Wright-Patterson AFB, OH: Air Force Flight Dynamics Laboratory, 1971.

Shearing, G. VISPLAY, the C.S.I.R.O. graphical FORTRAN system. In R.D. Parslow & R.E. Green (Eds.), *Advanced computer graphics.* New York: Plenum Press, 1971, 967–993.

Sibley, E., Taylor, R., & Gordon, D. Graphical systems communication: an associative memory approach. *American Federation for Information Processing Society, Fall Joint Computer Conference* (Part 1), 1968, 545–555.

Smith, L.B. The use of interactive graphics to solve numerical problems. *Journal of Communications of the Association for Computing Machinery,* 1970, **13**(10), 625–634.

Sutherland, I.E. Sketchpad: a man-machine graphical communication system. *American Federation for Information Processing Society, Spring Joint Computer Conference* (Vol. 23). Baltimore, MD: Spartan Books, 1963, 329–346.

Sutherland, I.E. Computer graphics: ten unsolved problems. *Datamation,* May 1966, 22–27.

Sutherland, I.E., Sproull, R.F., & Schumacker, R.A. Sorting and the hidden surface problem. *American Federation for Information Processing Society* (Vol. 42). Montvale, NJ: AFIPS Press, 1973, 685–693.

Teixeira, J.F., & Sallen, R.P. The sylvania tablet: a new approach to graphic data input. *American Federation for Information Processing Society, Spring Joint Computer Conference.* Washington, DC: Thompson Books, 1968, 315.

Van Dam, A. Computer driven displays and their use in man-machine interaction. In F. Alt & M. Rubinoff (Eds.), *Advances in computers* (Vol 7). New York: Academic Press, 1966, 239–290.

Williams, R. A survey of data structures for computer graphic systems. *Proceedings of the Computing Surveys, Association for Computing Machinery* (Vol. 3), 1971, 1–21.

Wright, T. A one-pass hidden line remover for computer drawn three space objects. *Proceedings of the 1972 Summer Computer Simulation Conference,* LaJolla, CA: Simulation Councils, Incorporated, 1972, 261–267.

Wright, T. A two space solution to the hidden line problem for plotting functions of two variables. *Proceedings of the Institute of Electrical and Electronics Engineers Transactions on Computers,* 1973, C-22(1), 28–33.

☆U.S. GOVERNMENT PRINTING OFFICE: 1977 — 771-057/25