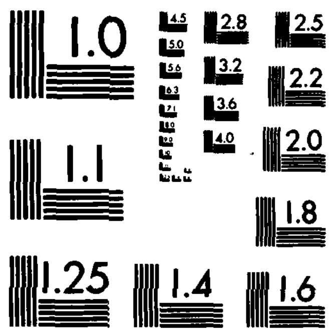
AD-A158 903 AUTOMATED CAMOUFLAGE PATTERN GENERATION TECHNOLOGY SURVEY(U) BDM CORP MCLEAN VA G N ROMSTEDT ET AL. 07 AUG 85 BDM/W-85-0135-TR DAAK70-83-D-0019

1/2

UNCLASSIFIED F/G 15/7 NL




MICROCOPY RESOLUTION TEST CHART

DARDS-1963-A

②

# THE BDM CORPORATION

7915 Jones Branch Drive, McLean, Virginia 22102-3396 • (703) 821-5000 • Telex: 901103 BDM MCLN

AD-A158 903

# Automated Camouflage Pattern Generation Technology Survey

PREPARED FOR THE US ARMY BELVOIR RESEARCH AND DEVELOPMENT CENTER, FORT BELVOIR, VIRGINIA 22060-5166



AUGUST 7, 1985 BDM/W-85-0135-TR

85 9 05 059

BDM

THE BDM CORPORATION

7915 Jones Branch Drive, McLean, Virginia 22102-3396 • (703) 821-5000 • Telex: 901103 BDM MCLN

# AUTOMATED CAMOUFLAGE PATTERN GENERATION TECHNOLOGY SURVEY

August 7, 1985

BDM/W-85-0135-TR



Prepared for the US Army Belvoir Research and Development Center, Fort Belvoir, Virginia 22060-5166.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A158 903

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY<br>2b. DECLASSIFICATION / DOWNGRADING SCHEDULE | 3. DISTRIBUTION / AVAILABILITY OF REPORT<br>Approved for release to U.S. Government agencies only. |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>BDM/W-85-0135-TR | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>The BDM Corporation<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>U.S. Army Belvoir Research and Development Center |
| 6c. ADDRESS (City, State, and ZIP Code)<br>7915 Jones Branch Drive<br>McLean, Virginia 22102 | 7b. ADDRESS (City, State, and ZIP Code)<br>Fort Belvoir, Virginia 22060-5166 |
| 8a. NAME OF FUNDING / SPONSORING ORGANIZATION<br>U.S Army Belvoir Research and Development Center<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>DAAK70-83-D-0019 T.0 0019 |
| 8c. ADDRESS (City, State, and ZIP Code)<br>Fort Belvoir, Virginia 22060-5166 | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. \| PROJECT NO. \| TASK NO. 0019 \| WORK UNIT ACCESSION NO |

11. TITLE (Include Security Classification)
Automated Camouflage Pattern Generation Technology Survey (UNCLASSIFIED)

12. PERSONAL AUTHOR(S)
G. N. Romstedt, D. Satko, D. Thoma, G. M. Sanz

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Final Technical | FROM 9/84 TO 8/85 | 1985 August 7 | 120 |

16. SUPPLEMENTARY NOTATION

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | optical digitization<br>Image capture system<br>robotic controlled painter<br>automated camouflage pattern generation |

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


This document summarizes the findings of technology surveys which were conducted to identify available hardware capabilities in three separate areas. First, a survey of computing systems was undertaken to determine their suitability for automated camouflage pattern generation. Second, a review of optical digitization methods and commercial hardware was conducted to evaluate presently available systems capable of being incorporated into the pattern generation system. Finally, commercially available robotic controlled painter systems were examined to determine their interface requirements to the camouflage pattern development software.


| | |
|---|---|
| 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT<br>☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Kenneth J. Dean | 22b. TELEPHONE (Include Area Code)<br>(703) 664-4458<br>22c. OFFICE SYMBOL<br>STRBE-HA |

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

# FOREWORD

The BDM Corporation, 7915 Jones Branch Drive, McLean, Virginia, 22102, is pleased to present this final report to the U.S. Army Belvoir Research and Development Center, Fort Belvoir, Virginia, 22060-5166. This document, entitled "Automated Camouflage Pattern Generation Technology Survey," summarized the results of analyses conducted to identify available hardware technology for application in the areas of automated camouflage pattern generation, optical digitization, and robotic painting.

# TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

## LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

# LIST OF TABLES

# CHAPTER I
# INTRODUCTION

## A. BACKGROUND

The United States and the Federal Republic of Germany have conducted a joint investigation and evaluation of camouflage patterns for Central Europe. The investigation compared the current U.S. four-color pattern, the German single-color green, and several alternatives in both number and types of colors as well as patterning style. The objective was to find a superior pattern which could be adopted as a standard by both nations' forces and thus eliminate one further identifying discriminator for the threat to use. In these documented field tests, a new three-color pattern was found to be superior. This pattern, using low gloss brown, green, and black paints, relies on areas of opposing contrast to disrupt the obvious shape and form of the vehicle. Precise control over the color ratios is maintained to match the predominate central European terrain and to make the horizontal surfaces relatively darker than the vertical. → 5a # 473

In February 1983, AMC directed that the US/GE three-color pattern be adopted by U.S. forces and applied to all Army vehicles. Since the joint investigation involved only a few basic combat and support vehicles, specific camouflage patterns were required for the remaining several hundred types of Army equipment. This task would be expensive and prohibitively time consuming to accomplish using traditional manual techniques. Two particularly difficult tasks are the control of color ratios on curved surfaces and the matching of patterns from view to view. These tasks require numerous projected intermediate views just to approximate the correct result. An automated system was considered necessary to precisely manage those tasks and significantly reduce the labor required to generate and edit the camouflage pattern.

In April 1984, The BDM Corporation presented the U.S. Army Belvoir Research and Development Center (BRDC) with an automated camouflage pattern generator in support of the ongoing camouflage modernization/standardization program. This pattern generator provided BRDC with the capability to produce a camouflage pattern in precise conformity to stated US/GE design criteria. Significant cost and time savings will be realized by using the automated system instead of manual techniques, particularly in the future when new camouflage patterning requirements are stated.

The software and techniques produced in the above effort are readily applicable to the full spectrum of vehicles and other items targeted by the camouflage modernization/standardization program. In addition, the automated pattern generation system is transferable to a wide range of computer-aided design and drafting (CADD) systems. The software is coded in standard Fortran 77 with Tektronix graphics drivers for the maximum portability. The commercial software requirements are a three-dimensional mechanical drafting package and a surface finite element mesh generation package.

## B. OBJECTIVES

The objective of this automated camouflage pattern generation technology survey is to support the actual implementation of the automated pattern generator developed for the Belvoir R&D Center by BDM. The aim is to convert the methodology into a suitable and cost-effective production system for the production of 1,500 separate camouflage patterns at an initial rate of 20 patterns per month.

To accomplish this goal, BDM performed the following investigations:

(1) An investigation of the cost and feasibility of operating the pattern generator methodology/software on in-house computing equipment available at the Belvoir R&D Center. The cost effectiveness of the in-house systems was compared with those of commercially available systems, and recommendations on the most preferred system were made.

(2) A study of the potential for system growth to include the capability of optical digitization of engineering drawings.

This capability would provide a high speed, high volume, and low cost means of generating accurate inputs to the designated automated camouflage pattern generator system.

(3) An examination of the technical/logistical requirements which would be introduced by adding robotic controlled painting equipment to the system. With such equipment, painting camouflage patterns would be fast, economical and consistent. The use of numerically controlled painters would allow direct control by the pattern generation software, thereby eliminating the slow and error-prone process of manually teaching a robot.

## C. TECHNICAL APPROACH

BDM divided the automated camouflage pattern generation technical survey into the following five sections:

1. Evaluation of BRDC In-House Computing Equipment

An objective evaluation of the CDC Cyber 170/835, Prime 750, IBM 4341, and Sage systems was performed. Analysis parameters included capacity, availability, space requirements, and operating costs.

2. Identification of Applicable Commercially Available Systems and Comparison with In-House Equipment

The commercially available systems were judged in the same objective manner as the in-house systems. Procurement costs were included in this analysis. The most cost efficient of those in-house and commercially available systems examined was recommended for use by the Belvoir R&D Center.

3. Summary of Optical Digitization Hardware

A comprehensive survey of all applicable hardware in an optical digitization system was conducted. The primary hardware components included were video cameras, video acquisition systems, image processing systems, display systems, and hard copy devices.

4. Technical Survey of Digitization Software

This survey defined the major functional elements necessary in an optical digitization program. The list of desired functions included edge detection, filtering, rotation, zooming, panning, and image combination. Present image resolution difficulties were discussed, and a possible vectorization procedure was presented as a means of improving poor resolution.

5. Review of Commercially Available Robotic Controlled Painter Technology

A survey of present robotic technologies was conducted in order to determine commercial capabilities for painting applications. Available designs, robot control methods, and programming characteristics were discussed with respect to desirability in an optimum robotic painter system. A review of several commercial systems was included.

## D. SUMMARY OF RESULTS

The conclusions resulting from these technical surveys are listed below:

(1) While the Belvoir R&D Center currently owns the IBM 4341, a computer which could satisfy requirements for the automated camouflage pattern generation system, the Auto-trol AGW is the recommended computer system due to much lower operating costs and technical superiority.

(2) Present optical digitization technology has advanced sufficiently to allow the optical scanning of plane view mechanical drawings with good resolution. Unfortunately, incorporation of this technology into the desired pattern generation system is not possible at this point due to as yet unresolved problems in pattern recognition by the computer and in the extension of several two-dimensional images into a three-dimensional composite representation.

(3) The necessary robotic mechanical technology presently exists for the camouflage pattern painting application. Due to the complexity involved in painting a large vehicle in three separate colors, the use of several robots per vehicle is suggested. While some robots are equipped with a digital control interface, the programming will be intricate. An additional restraint is presented by the fact that the digital interface will have to be tailored to each of the robots selected due to a lack of standardization in robot communications "languages".

The following chapters support the findings above. Chapter II presents the results of the computer systems inquiry. The technical survey of optical digitization systems is contained in Chapter III. The investigation of robotic controlled painter applicability to the camouflage pattern generation system is provided in Chapter IV. The final chapter, Chapter V, details the important conclusions which can be drawn from this technical survey.

# CHAPTER II
# AUTOMATED CAMOUFLAGE PATTERN GENERATION
# METHODOLOGY TRANSITION
# COMPUTER SYSTEMS INQUIRY

## A. BACKGROUND AND OBJECTIVE

The automated camouflage pattern generation system developed by BDM for BR&DC was implemented on a timeshared VAX 11/780 minicomputer using leased Tektronix graphics hardware and licensed computer aided design (CAD) software. The mechanical drafting chores were implemented on an automated drafting system, with the developed patterns manually superimposed on the finished master drawings.

The objective of this computer systems inquiry is to convert the methodology into a production system that is suitable and cost effective for the production of 1500 patterns at an initial rate of 20 patterns per month. This transition will include the integration of pattern design and drafting functions into a single system, and will facilitate the future growth of the system into optical digitization and numerical control functions. Because of the long term nature of this program and the labor intensive task of developing vehicle descriptions and master mechanical drawings, it is essential that the Government control the production equipment directly. This will insure that the substantial labor investment is not lost for future efforts and that the product is obtained at a minimum cost.

For this system the Belvoir R&D Center is making feasibility and cost comparisons among its existing in-house computing equipment and other commercially available equipment. Task Order 0019 specifically cites the following in-house systems be considered:

(1) CDC Cyber 170/835,

(2) Prime 750,

(3) IBM 4341,
(4) Sage, and
(5) VAX 11/70.

These systems with the exception of the VAX are owned by BRADC and operated on a time share basis by the Computer Center. The VAX 11/70 is an apparent typographical error, referring either to the PDP 11/70 or to one of the VAX 11/700 series computers (all made by the Digital Equipment Corporation). Since BRADC does not have a VAX, the BDM VAX 11/780 is considered as an alternative "in-house" system. The commercially available equipment to which the in-house systems are compared are the three integrated CAD systems marketed by the Auto-trol Technology, Computervision, and Intergraph Corporations. These systems are described in detail in Chapter IV of the Technical Report prepared for the previous Task Order.

## B. METHODOLOGY

Of primary concern in this inquiry is the objective evaluation of the five specified in-house computer systems. This evaluation will first consider feasibility, and then cost, in determining whether any of the in-house systems represent viable central processors for the automated pattern generation system. A questionnaire has been developed to address critical issues and is included as Figure II-1.

### 1. Feasibility

The demands on the central processor are fairly stringent. It is expected that three interactive workstations as a minimum will be necessary to support the planned initial production rate of twenty finished patterns per month. Because this rate may be insufficient, the system selected should not preclude the addition of a fourth workstation to augment the pattern production effort. Thus, the ability to support four workstations represents a conservative basis on which to judge the computer systems' feasibility.

CAMOUFLAGE MODERNIZATION PROGRAM
COMPUTER SYSTEMS INQUIRY

Objective: Support four interactive 3D color graphics workstations simultaneously in the automated development of camouflage patterns.

Capacity: These workstations will fully occupy a VAX 11/780. Does this system have sufficient unused capacity?

- What fraction of the prime shift is the computer idle?
- How does this system compare to the VAX?
- What upgrades would give this needed capacity and at what cost?

Availability: These workstations will be used 16 hours/day, 5 days/week, with optional 3rd shift and weekend schedules.

- What are your normal hours of operation?
- What additional costs do you incur to accommodate this schedule?
- Do you have fixed processing requirements which will preclude some portion of this schedule?

Hardware Issues: The following are required hardware; E size pen plotter, 19200 baud communication for terminals, tape drives, 4 MB Memory (actual or virtual), and 256 MB Disk Space.

- Are they available?
- Can they be obtained and at what cost?

Software Issues: The generation of patterns requires the following software: PATRAN-G, unspecified Drafting Package, FORTRAN 77.

- Does the system currently have these packages?
- Can they be obtained and at what cost?

Facility: The terminals and work area will required about 700 ft$^2$ of dedicated office space.

- Is this space available?
- Can it be supported by high speed data communications?

Costs:

- What are your rates?
  $/CPU hour:
  $/MB disk storage/day:
  $/connect hour:
  other charges:
- What are your shift cost differentials:
  2nd shift?
  3rd shift?
  weekends?

Figure II-1. Computer Systems Inquiry Questionnaire

a. Capacity

Supporting three or four workstations is not a trivial matter. The techniques and algorithms of three-dimensional visualization are particularly computation intensive. To be responsive to many users, the computer must be fast. The interactive nature of the design and drafting tasks implies that the computer's responsiveness is critical to the production rate. Four workstations, operating simultaneously, can easily overwhelm and drastically slow down a computer, particularly if other users or programs are also competing for system processing time. Based on BDM's experience during the development effort, it is estimated that the four workstations will fully occupy a VAX 11/780. Therefore, the existing in-house systems must have unused capacity equivalent to a VAX 11/780 to be viable.

b. Availability

The systems' availability is the second factor to be considered. To support the planned production rate, it is expected that two shifts of CAD operators will be employed. The third shift and weekends are reserved for abnormally heavy production periods. This schedule is well within the normal operating hours of most systems, and accommodating to a full 24 hour operation is rarely a problem. Modern computers can be left unattended for most normal operations and they would be simply left on overnight if needed. The second aspect of availability is the requirement for the system to periodically process certain jobs of extremely demanding nature and high priority. An example of these jobs might be the biweekly installation payroll. It must be done and it must take precedence over other work. During the period it is running, all other tasks, including the camouflage pattern generation system, will wait.

c. Hardware and Software Issues

Certain items of system hardware are required to make the pattern generation system viable. If they aren't currently available, they might be purchased and integrated into the system. If they can not be acquired, then these items will prevent the effective application of the computer to camouflage pattern generation. An E-size pen plotter will be needed to make hardcopy plots of patterns, 19200 baud communications are needed as a minimum to transfer graphics data to the workstation, tape drives are needed for backing up and archiving completed patterns, 256 megabytes of on-line hard disk space as a minimum is needed to support multiple processes and data files, and 4 megabytes of actual or virtual memory is needed to process the largest expected single task of pattern generation.

Similarly, commercial software must be available to accomplish certain tasks. A 3-D finite element creation package such as PATRAN-G is needed to develop the input representation of the vehicle, a drafting package is needed to perform the mechanical drafting chores in producing the finished patterns, and a FORTRAN 77 compiler is needed to translate the developed patterning software into the machine instructions.

d. Facility

Finally, dedicated facility space must be available in the close vicinity of the computer system. High speed data communications are supported only over short distances (maximum 150 feet). It is expected that at least 700 $ft^2$ of space will be needed for the workstations, plotter, worktables, and tape storage. Additional space will be needed for management and inspection staff offices (est. 650 $ft^2$) and drawing storage (est. 1300 $ft^2$). This facility will require normal utility service, including temperature and humidity control, to maintain the equipment, tapes, and drawings. The facility must to be dedicated for the duration of the project.

2. Costs

The evaluation of costs is quite straightforward. Three general categories of costs are considered. The first category are the initial equipment procurement costs for the initial three terminals and the plotter and software costs for the CAD software. The second category are the timesharing costs charged to the computer users. These costs are based on CPU and disk usage, terminal connect time, and assorted other measures of computer usage. The final category are those incurring operations and maintenance costs such as licensing fees and maintenance contracts. These costs are each estimated and then summed to yield the total estimated cost for 1500 patterns, and the per pattern cost.

## C. FINDINGS

For the comparison of systems, the existing in-house computers were considered directly and objectively alongside the BDM VAX and the three integrated CAD systems by Auto-trol, Computervision, and Intergraph. Those that passed feasibility tests were evaluated for cost. The feasibility results summary is presented in Table II-1. They are described, system by system, in the paragraphs below. Data on BRADC computers was obtained in interviews with Barbara Lowder and Dick Gardner of the Management Information Systems Directorate and is frequently subjective.

### 1. CDC Cyber 170/835

This computer is located in the BRADC computer center, which is already crowded with equipment. The maximum communications rate is 9600 baud, half the required speed. Although the CDC is an extremely capable processor, it is heavily used for current BRADC scientific and business applications and is rarely idle. It is also limited by its relatively small fixed memory size of about 98,300 words (equivalent to at most 737,250 bytes but almost impossible to realize). Extended core storage is available but is still insufficient for a single user.

### 2. Prime 750

The Prime is also located in the BRADC computer center and thus lacks the requisite facility space. It is also heavily used and limited by a maximum 9600 baud communications rate. It is a virtual memory machine and is currently employed in 2-D graphics applications.

### 3. IBM 4341

The IBM is located in Building 354 which has recently been converted for use as a computer center. Building 354 has sufficient facility space for the workstations, but not for the entire office and storage space requirements. The IBM ca. communicate at 19200 baud. Since

TABLE II-1. FEASIBILITY COMPARISONS OF SYSTEMS

| SYSTEMS CONSIDERED | GOVT. OWNED | CAPA-CITY | AVAIL-ABILITY | HARD-WARE | SOFT-WARE | FACI-LITY | REMARKS |
|---|---|---|---|---|---|---|---|
| CDC CYBER 170/835 | YES | NO | YES | NO | YES* | NO | System heavily used, 9600 baud max, no space in Computer Center. |
| PRIME 750 | YES | NO | YES | NO | YES | NO | System heavily used, 9600 baud max, no space in Computer Center. |
| IBM 4341 | YES | YES | YES | YES | YES* | YES | Facility available in Bldg 354. |
| SAGE | YES | NO | YES | NO | NO | NO | System not competent to task. |
| VAX 11/780 | NO | YES | YES | YES | YES | YES | |
| AUTO-TROL AGW | NO | YES | YES | YES | YES | YES | |
| COMPUTERVISION | NO | YES | YES | YES | NO | YES | System cannot be user programmed. |
| INTERGRAPH | NO | YES | YES | YES | YES | YES | |

* Requires the substitution of ANVIL4000 for PATRAN-G for finite element generation requirement, and upgrade of operating systems.

it is a recent addition to the BRADC computing center it is currently underutilized. IBM computers have typically been applied to business management problems and it is expected that its major use will be in that arena, as well as a few specifically requested scientific packages designed for IBM systems (e.g., SAS Graph). The camouflage pattern generator would use up most of the remaining processing capacity of the IBM. Being one of the first programs on the system, it would be "vested" in a sense that it would discourage other applications, including the intended business management ones.

Transferring to the IBM would require the substitution of the MCS product ANVIL 4000 for the PATRAN-G software package, which is not supported on the IBM. ANVIL 4000 is a competent substitute for PATRAN but is rather expensive, priced at $120,000 for the entire system. It also includes drafting software as well as 3-D model building and finite element mesh generation. Also required will be the planned (FY85) upgrade of the IBM to the VM SP operating system and increase in memory to 16MB. The IBM is comparable to the VAX 11/780 in overall speed, but is perhaps 20 percent faster. It is estimated that 4 CPU hours per pattern will be required. Specific timeshare rates for the IBM have not been established, but are expected to be comparable to the CDC. At $360/CPU hour and $10/connect hour, the estimated annual timeshare cost is $650,000. In addition, the terminals and plotter will require an initial investment of $130,000.

4. Sage

The Sage computers are maintained at the BRADC computer center and form a network throughout BRADC of word processing and spreadsheet applications. They are not considered available or competent to this task.

5. VAX 11/780

The BDM Corporation used one of its two VAX 11/780's to develop this pattern generation system. They are located at the corporate offices in McLean, Virginia. Other potential production contractors may also have VAX computers that can be used. The BDM VAX is about 60 percent available so some current users would be displaced to the other system. It is estimated that 5 CPU hours per pattern are required on the VAX to complete a pattern. At $108/CPU hour and $8/connect hour, the estimated annual timeshare cost is $380,000. In addition the terminals, plotter, and drafting software license will require an initial investment of about $250,000.

6. Auto-trol AGW

The Auto-trol Advanced Graphics Workstation is the first of the three integrated CAD systems that would need to be purchased. It can be completely dedicated to camouflage pattern generation and is flexible in facility assignment. An AGW system with three highly competent workstations, plotter, printer, and tape drive has been priced at approximately $500,000. Annual maintenance and software license fees are estimated at $60,000. These workstations run all necessary software and are specifically configured for design and drafting tasks. Each workstation contains a dedicated micro-computer that rivals a VAX 11/730 in processing power, a dedicated disk drive, and up to 4MB of memory. The workstations are networked together so they can share data files and processing power. Additional workstations can be added with no degradation to system responsiveness.

7. Computervision

The Computervision CGP-200X/APU system was considered but excluded because it is not able to be programmed in FORTRAN. The Computervision package is extremely competent and very expensive, but they have chosen to deliver only turnkey systems with a closed and tightly controlled architecture.

8. Intergraph

The Intergraph system is a package of intelligent terminals connected to a supplied VAX 11/750 computer. It is competently equipped with design and drafting software and can run all software used or developed in the camouflage pattern production scheme. It would be purchased at a total estimated cost of $580,000. Annual maintenance and software licensing fees are estimated at $80,000. Because it is configured around a centralized computer, the expandability of the Intergraph system is somewhat limited by the capability of the VAX 11/750.

## D. CONCLUSIONS

The IBM 4341 represents the only in-house computer that could technically satisfy the computing requirements of the automated camouflage pattern generation system. Its use would require the substantial commitment of its processing availability and nearby facilities to meet those requirements. In so doing, it would effectively preempt the other intended uses for the computer and space.

The VAX 11/780, although technically competent, is not owned by BRADC and thus cannot represent a permanent capability for pattern generation. The integrated CAD systems by Intergraph and Auto-trol are the best alternatives for in-house capabilities because of their dedicated capacity and lower costs. Of these, the Auto-trol is less expensive and more flexible.

The relative cost summary is presented in Table II-2. The expected cost to use the IBM is a factor of five greater than to purchase a dedicated computer aided design and drafting system such as the Auto-trol AGW. The benefits of the Auto-trol package, consisting of three workstations, plotter, printer, and tape drive, include flexibility in the facility location, dedicated processors for each workstation, infinite expandability, integrated design and drafting software, and single point of contact for maintenance. These advantages are even more significant than the cost differential in ensuring a permanent CAD capability at BRADC.

TABLE II-2. ESTIMATED COSTS FOR FEASIBLE SYSTEMS[1]
(ALL COSTS IN THOUSANDS OF DOLLARS)

| FEASIBLE SYSTEMS | CURR. GOVT. OWNED | INVESTMENT COSTS (SYSTEMS, TERMINALS PERIPHERALS) | TIMESHARE COSTS/YEAR | OPERATIONS & MAINTENANCE COSTS/YEAR | TOTAL COST | COST PER PATTERN |
|---|---|---|---|---|---|---|
| IBM 4341 | YES | 250[4] | 650 | 0[2] | 4,150 | 2.77 |
| VAX 11/780 | NO | 250 | 380 | 0[2] | 2,530 | 1.69 |
| AUTO-TROL AGW | NO | 500 | 0[3] | 60 | 860 | 0.57 |
| INTERGRAPH | NO | 580 | 0[3] | 80 | 1,060 | 0.71 |

1. Estimate is based on requiring three workstations, running two shifts, producing an average of 20 patterns per month over a production run of 1500 patterns.
2. Assumes O&M cost is covered by timeshare user fees.
3. Systems are dedicated.
4. Does not include required system upgrades which are currently planned.

# CHAPTER III
# SURVEY OF IMAGE PROCESSING TECHNOLOGIES

## A. GENERAL

### 1. Introduction

In its early stages of development, image processing was limited by the high cost of hardware and applications software to use in just a few specialized areas - such as land survey, military, research, and medical applications. Today, however, new technology is allowing real-time image processing to become widely available for a broad range of uses in many industrial and commercial applications including the growing fields of computer-aided design and computer-aided manufacturing (CAD/CAM). The specific application dealt with herein will be the automated camouflage pattern generation for military vehicles. It is believed that through the use of commercial technologies, great savings in labor might be achieved in the input and construction of 2D plane view drawings, and in the creation of a 3D surface model of the vehicle to be camouflaged. Currently these tasks are tedious, even using state-of-the-art CAD systems, because the operator is forced to manually enter and edit every detail. Because the operator is often merely tracing an existing drawing, it is conceivable that the drawing could be optically scanned and automatically put into format for further CAD system manipulation. This survey will examine the image processing technologies, both hardware and software, that are currently available and how they are being used to determine the system requirements and feasibility in the automated production of camouflage patterns.

### 2. Present Applications of Image Processing

Because of the low cost and high performance/high reliability characteristics of the many image processing systems on the market today, they are currently being used in industrial, medical, and research establishments throughout the world. Typical applications include:

a. Medical Imaging: Digital radiography, x-ray averaging and recording, ultrasonic analysis, NMR, CAT, etc.

b. Microscopy: Semiconductor and silicon wafer inspection; medical and biological research; electron microscope image averaging.

c. Infrared Thermography: Pseudocolor enhancement of infrared photographs to facilitate analysis.

d. Astronomy: Image averaging of telescopic images for tracking and identifying stars.

e. Satellite and Aerial Imagery: Land use; forest management; cartography; weather forecasting; military reconnaissance.

f. Petroleum and Mineral Exploration: 3-D computer displays of earth's interior aid in isolating petroleum reservoirs; images of geological formations used to locate mineral deposits and in recording seismic data.

g. Publishing and Archival Storage: Storage and restoration/reconstruction of valuable documents (Smithsonian).

h. Teleconferencing: Digitizing images for transmission over standard telephone lines and microwave transmitters.

i. Security: Video image matching and subtracting for inventory and sensitive area security.

j. Factory Inspection Systems: Robotic vision and quality control on production lines; pattern matching and digital subtraction for high-speed circuit board inspection; flaw analysis; non-destructive industrial x-ray inspection; area measurement; distance measurement; uniformity, etc.

## B. HARDWARE TECHNOLOGY

### 1. Introduction to Hardware Technology for Image Processing

This section will describe the architecture and design philosophy of a typical image processing system. Probably the most significant single aspect of an image processing system is its architecture/design. The multiplexing of data channels, the pipelining of image processors, the availability of high-speed random access refresh memory chips, and the perfection of very high resolution color and monochrome monitors all have combined to allow the development of many powerful and highly interactive image processing systems.

With a RAM refresh memory storing the picture image, it becomes possible to "roam" or "zoom" around a large data base in a simultaneous interactive environment. "Roaming" (or "panning") slides an image across the viewing display at a particular pixel increment providing fast access to the undisplayed regions of the image. "Zooming" refers to the enlargement or reduction of the entire screen image allowing the user to readily view small details of an image or their position in the larger view. Both roam and zoom are non-destructive, real-time functions. They can be used in tandem to instantly examine any part of the image without having to recreate the larger view.

Furthermore, with a "table look-up" approach to image representation, the number stored in RAM for each pixel is merely a pointer to a table of colors or grey levels. Colors may be instantly varied by changing the value in the table rather than changing all the pointers. The number of simultaneous colors available on the monitor is limited by the number of bits allocated to each pixel and hence to the number of pointers available. To accommodate special requirements, the system refresh memory can often be allocated to trade off bits per pixel against the total number of pixels. The impact is that the screen resolution and number of colors simultaneously displayable can be varied to produce an optimum result for any application. By placing the allocation of refresh memory and color table values under software control, one obtains a truly interactive "digital light table" processing system.

In addition to breakthroughs provided by larger, cheaper random access memories, pipeline processing technology has also contributed to modern day image processing systems. For instance, the typical 3 x 3 convolution filters followed by both linear and non-linear function memory combinations can now process a 512 x 512 x 8 monochrome image in 1/30 of a second (equivalent to the frame time of a television). Convolution is the process by which the pixel values of an output image are calculated by way of various operations performed on the input pixel values. Convolution will be discussed in greater detail in the survey of software technology, but for now, it simply illustrates the capabilities of a pipeline processor.

The typical components of an image processing system might include the following, each of which is described in subsequent paragraphs:

- Video Camera (or digitizing input device)
- System Bus
  - Processor Module (Microcomputer)
  - Host Interface/DMA
- Image Bus
  - Video Acquisition System
    - Video Controller
    - Video Acquisition Processor
  - Image/Graphics Refresh Memory
    - Image Refresh I/O Control Card
    - Image Refresh Data Base Control Card
    - Frame Write Card
  - Pipeline Image Processor
  - Display Systems
    - Display Controller
    - Display Processor
- Display Devices

## 2. Image Processing Hardware Modules

### a. Video Cameras/Lenses

Cameras intended for home video recording are generally color cameras of medium to low resolution. These will provide acceptable performance for such applications as surveillance, picture/portrait capture, and process monitoring. Their primary limitations are the video bandwidth and scan resolution which limit spatial resolution to about 300 x 300 pixels even though the image may be digitized at a much higher resolution.

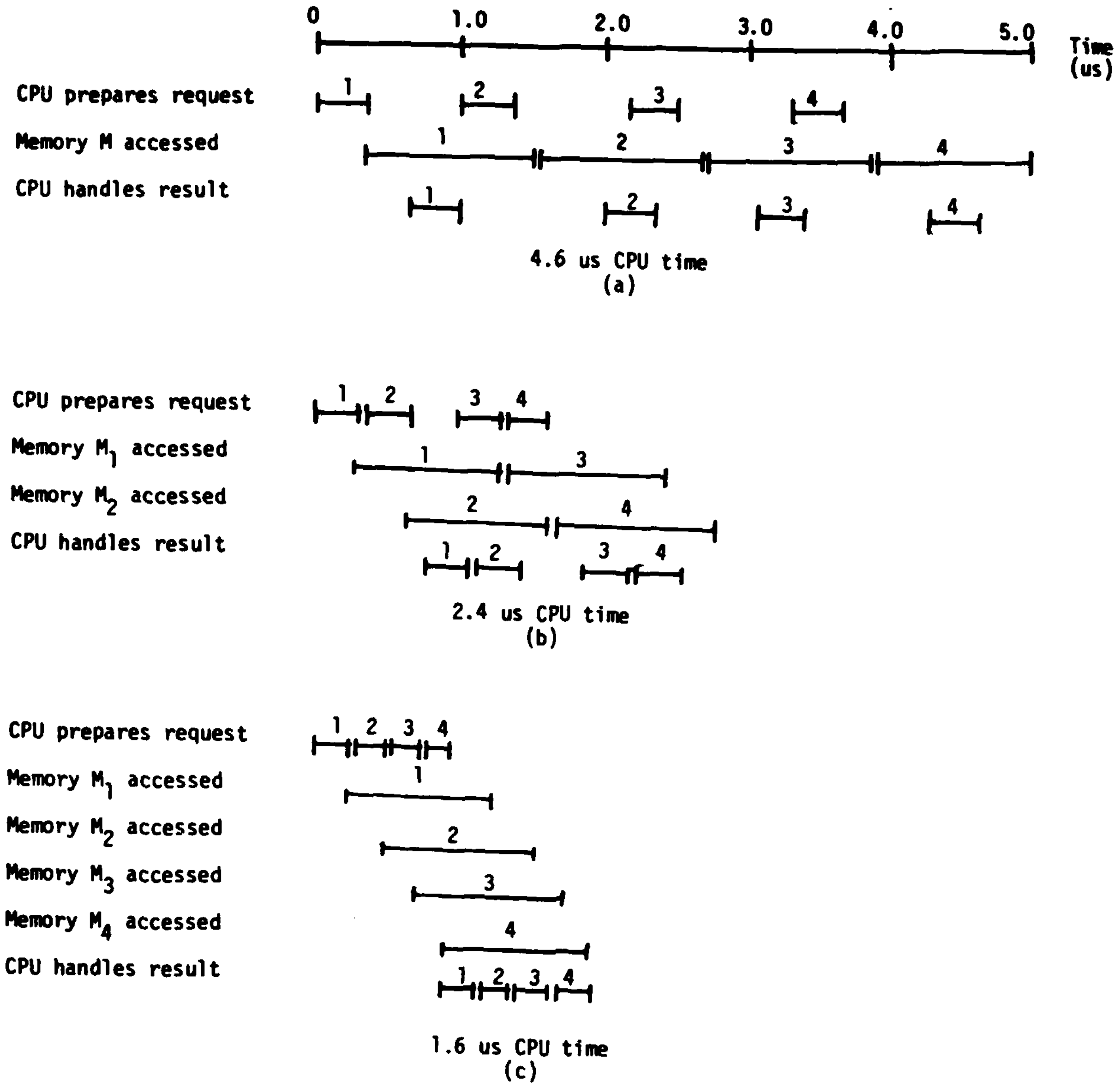
Another limitation is the scan speed. Scanning is inherently limited by the speed at which the Central Processing Unit (CPU) can access and allocate sequential memory addresses to store the captured image. One method of speeding up this process is called interlacing or interleaving. Interlace involves the allocation and arrangement of sequential memory addresses into different memory modules or different segments of memory. By this means, the total time taken to access a sequence of memory locations can be reduced since several memory accesses may be overlapped by a high speed CPU. This process can be thought of in increments or ratios (i.e., 2:1, 4:1, N:1) of interleaving.

For example, consider a memory with 0.6uS access time (i.e, the time to get a word from memory to the processor) and a 1.2uS cycle (i.e., the time after the initiation of an access before the memory can be processed again), and a processor requiring 0.2uS to prepare a memory request and a further 0.2uS to handle the result. Under these conditions, a sequence of four memory accesses would take 4.6uS with no interlace, 2.4uS with 2:1 interlace and 1.6uS with 4:1 interlace. This process is illustrated in Figure III-1.

Notice in this example that 4:1 interlace provides a smaller incremental advantage than does 2:1. This is a result of the particular choice made of CPU and memory timing, which happens to be fairly well suited for 2:1 interlace. Notice further that the 4:1 interlace leaves the CPU fully occupied (at least as far as the example goes). The result is that interlacing greater than 4:1 will provide no increase in system speed. The speed for 4:1 (or more) has become CPU-limited rather than memory limited.

The example chosen closely parallels CPU and memory characteristics found in image processing systems. For this reason, it appears that interlaced 2:1 video cameras would offer the best performance. Furthermore, 2:1 interlace cameras are less expensive than the higher ratio models.

Four commonly used types of cameras are Vidicon, Newvicon, Ultricon and CCD (Charged Couple Device). The choice of camera type is




Timing diagram showing sequence of four memory accesses (1,2,3,4) with (a) no interlace, (b) 2:1 interlace and (c) 4:1 interlace. (Time scale = 0.2us/division)

Figure III-1. Interleaved Memory Access

determined by its performance specifications such as dynamic range (light ratio), resolution (# of lines), susceptibility to "burning", and response time (speed). It should be noted that Vidicon tubes are subject to "burning", i.e., if left exposed to continuous high levels of light, the scene will become permanently "burned" into the tube. Newvicons, Ultricons, and CCD arrays do not exhibit this sensitivity, and they also offer much higher sensitivity to low-light levels and much broader spectral response. Tube size is also an important factor. For instance, a one-inch tube will offer higher light sensitivity and greater resolution than a two-thirds inch tube; however, the tube itself and the appropriate lenses are much more expensive. (See Table III-1 for a summary of video tubes.)

As with photography, lens choice is a critical part of obtaining focus and linearity over the image field. The "F"-stop relates to the light-gathering properties of the lens. As the "F" rating is lowered, more light can be passed. The size (mm) of the lens is inversely proportional to the field of view at a given distance. Hence, a 50 mm lens has a broader view than a 100 mm lens; however, a wide lens is more susceptible to non-linear distortions along the edges than a narrow lens. A "macro" lens may be necessary to preserve focus over the entire field if close-up viewing of an object is desired. There are also wide-angle, "zoom" telephoto, and "pinhole" lenses for special applications.

Since there are numerous combinations of cameras and lenses available, each specific application should be analyzed and evaluated to determine which combination best suits the system requirements. The analysis should consider the size of the imaged object, the resolution/detail desired, the distance from camera to the object, the color or grey level discrimination, and the ambient and controlled lighting available.

b. <u>System Bus</u>

Before continuing, it is necessary to define a digital architectural configuration common to all image processing systems: a bus. A bus is a group of wires carrying digital signals. In Figure III-2,

TABLE III-1. TYPICAL SPECIFICATIONS FOR VIDEO TUBES

| Specs / Tube | Dynamic Range (Light Ratio) | Resolution (# of lines) | "Burning"? | Response Time (minutes) |
|---|---|---|---|---|
| Vidicon | $2\text{-}5 \times 10^5 : 1$ | 500 - 800 | Yes | .25 - 1 |
| Newvicon | $2.6 \times 10^6 : 1$ | 700 - 800 | No | .25 - 1 |
| Ultricon | $2.6 \times 10^6 : 1$ | 700 - 800 | No | .25 - 1 |
| CCD Array | $2.6 \times 10^6 : 1$ | 1728X2846 (max) | No | .1 - .25 |

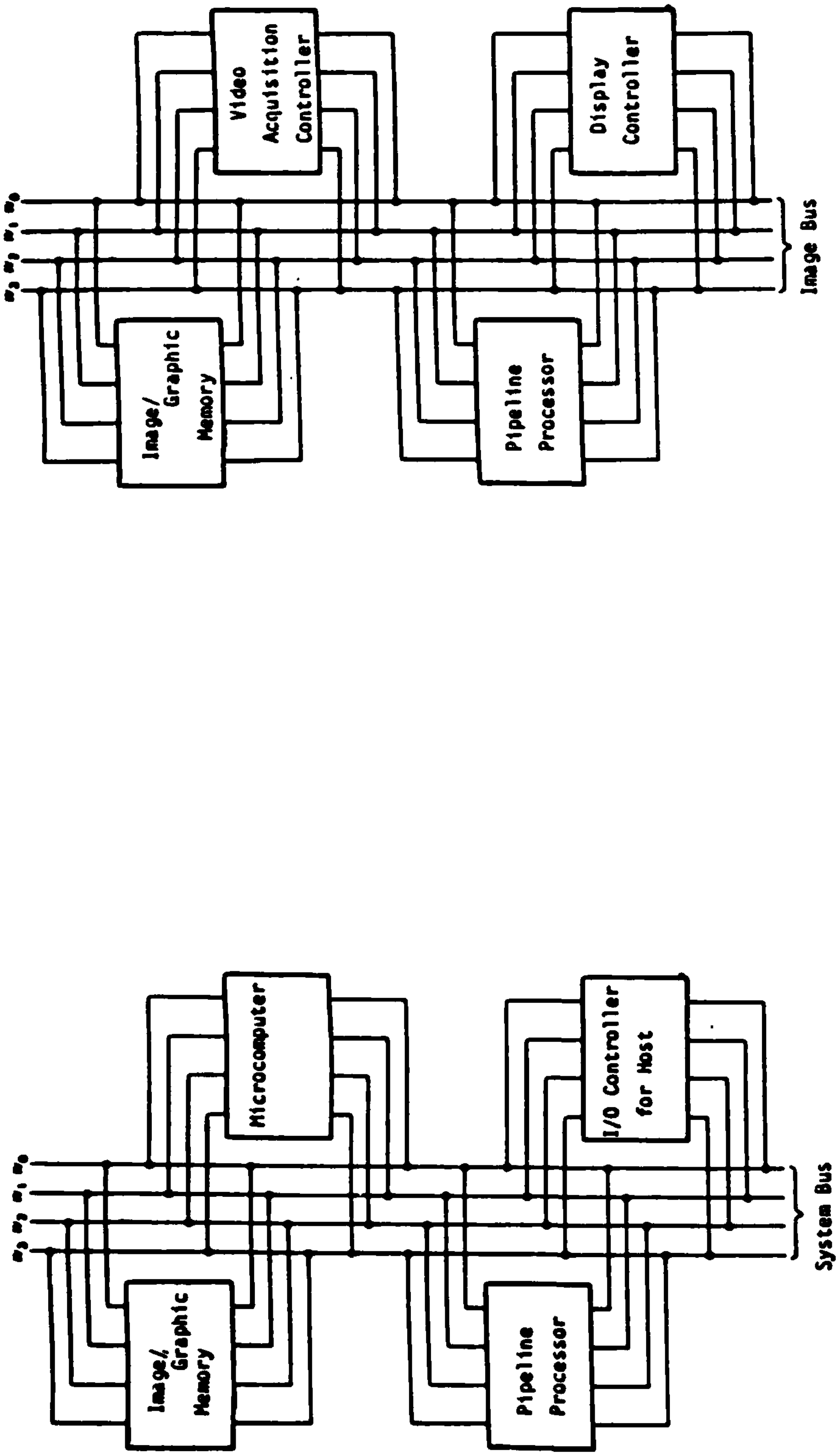



Figure III-2. Bus Structures

vertical wires $W_3$, $W_2$, $W_1$ and $W_0$ form a bus; these wires are a common transmission path between various memory registers in an image processing system. For example, on the system bus, the input data bits for the pipeline processor come from the "W" bus; at the same time, the output of the pipeline processor registers connects back to the "W" bus. Similarly, all other registers have their inputs and outputs connected to the system bus. The beauty of bus organization is the ease of transferring a word of data from one register to another. For comparative purposes, the most important characteristics of a bus architecture are data path width and data transmission rate. For instance, Aydin computer's "Mem Bus" has a data path width of 16 bits and a transmission rate of 1.25 million pixels/sec. Here, a pixel is just a "bit" of information traveling on the data bus. Path width and transmission rate indicate the relative speed at which a particular system can handle data. In turn, this is an excellent indicator of the system's capability to perform data manipulations and processing in "real-time" (results seem to be instantaneous in the user's time frame). A good example of this in image processing is the ability to view the results of a function - with little or no machine hesitation - immediately after it has been executed.

All commercially available turnkey image processing systems operate either in a stand-alone capacity or are interfaced to a host computer. In fact, several systems utilize both an internal microprocessor and a host computer (typically a minicomputer) for even more powerful image processing capabilities. However, the minimum requirement is only some form of processor module contained in either an internal processor chip or an external micro or mini computer system. This paragraph will describe a system with an integrated microprocessor and a link to an external "host" computer system.

1) Processor Module

A fully configured microcomputer based processor module would consist of a 16-bit or 32-bit microprocessor, a program memory, a disk controller, a magnetic tape controller, and communications interfaces (serial and parallel ports) all connected to the system bus (e.g., Motorola Versa Bus, Mem Bus, Intel Multi-bus, Dec Q-Bus, LSI Bus).

a) Microprocessor

A typical microprocessor used in image processing systems today is the Motorola MC68000. The characteristics of the MC68000 microprocessor complement the performance requirements of a real-time image processing system. The 68000's direct 16 mega-byte address space permits users to manipulate the large image memory quickly without resorting to "mapping windows". In convolution operations, for example, this dramatically speeds manipulations of parts of several images. In addition, the 68000 provides a large number of general purpose registers (15 of its 16 data/address registers are always available to the user). Hence, the user can perform more data manipulations without having to move intermediate results to a stack. This means an algorithm can be executed in less time and with fewer instructions.

b) Operating System

The operating system - the heart of any computer system - schedules tasks, allocates resources and handles file structures. It acts as a very basic interface between the hardware and the user. Its operation provides real-time, and very often multi-user/multi-tasking capabilities for a processor such as the 68000. The choice of operating systems generally falls into one of two categories: FOS or DOS.

With a FOS (firmware operating system), the image processing software is permanently stored on programmable read-only memory (PROMs) chips on a board within the microcomputer. On the other hand, a DOS (disk operating system) contains the software on hard or floppy disks and so a disk controller set (one or more boards) must be included. The DOS configuration allows great flexibility in the software package provided. In addition to image processing software, the DOS software package frequently includes an editor, a linking loader, assembler language, FORTRAN compiler, Pascal compiler and related utilities.

Typically, a microcomputer system can edit and compile programs while concurrently controlling the image processor operation. The system also can acquire, process, archive and display images simultaneously. This is made possible through the use of video and display processors which will be discussed in more detail later. While acquisition and display are taking place, image processing operations can be performed by the microcomputer system or a "pipeline image processor" (more about this later), or the image can be stored in image/graphics memory.

2) Host Interface/Direct Memory Access

A digital image processing system is commonly interfaced with a larger host computer to take advantage of the more powerful host features including higher speed processing, greater memory size, and access to a multitude of peripheral devices. This can be accomplished through the use of a host interface card and an I/O controller card. This interface is a digital data link capable of high-speed data transfer between the two machines over a finite distance. The maximum distance is usually limited by the length of the connecting cable; however, common phone lines can be utilized at greatly reduced transmission rates. Typically, the I/O controller board is installed in the image processing system, and the interface board is placed in the host computer. Then, these two boards are connected with a ribbon cable to achieve bidirectional parallel data transfer. This communications link allows commands and data transfer requests to be issued from either machine.

An alternate technique, direct memory access (DMA), allows for large amounts of data to be transferred at high speeds. Sequential data are transferred in large blocks from one machine to the other. In this way, complete images may be passed between the host and the microcomputer bus allowing for processing and subsequent storage. The typical DMA portion of a host interface will operate in two modes. For the system described to this point, the first mode might be the transfer of a full 16-bit word between the machines and placement in corresponding memory locations. The second and faster of the two modes can be used when pixel values are 8-bits or less. This is often referred to as "automatic data packaging", "byte-packed mode" or "data compression" because two bytes are concatenated to form one 16-bit word. When the image processor receives this word, it splits it into the two bytes which represent the value of two consecutive words. In this way, two 8-bit pixels are transferred at one time. The result of the data packing scheme is that data transfer is accelerated and storage requirements are reduced.

c. Image Bus

1) Video Acquisition Systems

In general, video acquisition systems are one of two types: a video controller or a video acquisition processor.

a) Video Controller

A video controller is a single board 8-bit/pixel analog to digital converter and memory control for digitizing and storing pixel images in image/graphics refresh memories. (A common image in this case might be a 512 x 512 pixel image). A video controller typically takes analog vector data (e.g., line, square, arc, color) and causes particular state changes in certain bits of the refresh memory.

b) Video Acquisition Processor

In a video acquisition processor, analog data from a video camera is translated into its internal machine format. The internal representation may be from one to 16 bits per pixel, but typically 8 bits (one byte) are used to describe each picture element. This will correspond to one of 256 colors or shades of grey. Data from the camera may be acquired sequentially or in interlaced form, usually in a format of 1024 x 1024 or 512 x 512 pixel image. For a 1024 x 1024 pixel image size, data can be acquired, displayed, and stored on a disk at approximately 7½ frames/second. At 512 x 512, data can be acquired, displayed, and stored on disk at approximately 30 frames/second. These figures represent data taken from a typical image processing system but can vary widely with different component configurations. (See Figure III-3 for the architecture of a typical video acquisition processor).

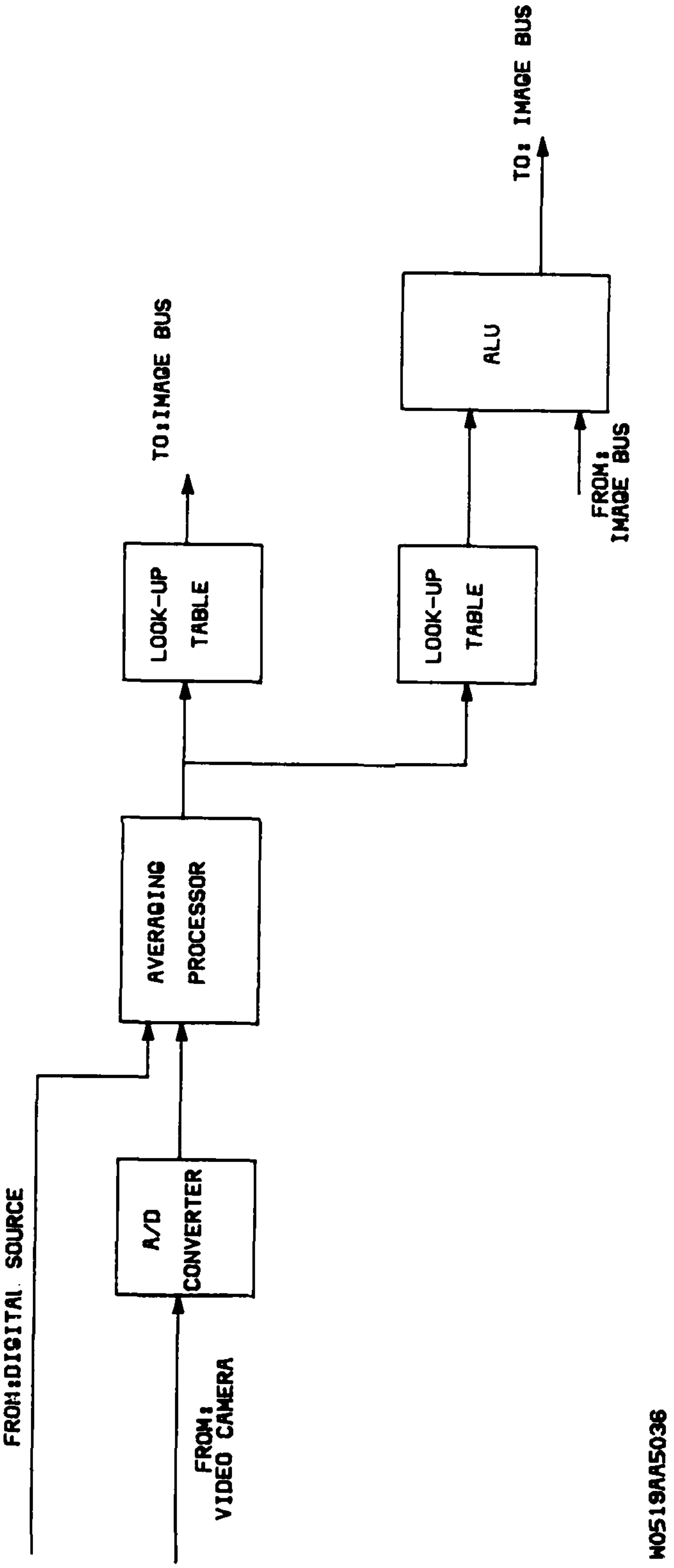



Figure III-3. Video Acquisition Processor Architecture

Image averaging is a useful application of the video acquisition processor. For example, to accommodate disk transfer rates, a sample 1024 x 1024 or 512 x 512 image could be averaged and subsampled to a 512 x 512 or 256 x 256 array, respectively, and stored within an image memory or display buffer. Although the resolution is reduced in the averaging scheme, the number of frames per second that can be saved on a hard disk is significantly increased.

Look-up tables (LUT) and the arithmetic logic unit (ALU) can also be used to produce interesting results within the video acquisition processor. For instance, by taking advantage of the LUT, the ALU, and the image feedback capabilities, a user could perform image addition, subtraction, pseudocolor, logarithm, and masking operations on a frame-to-frame basis, viewing the results as they occur. At the same time, raw or uncollected data could be transferred to the main image memory and subsequently stored on disk so the original data is available to the user at a later time.

2) Image/Graphics Refresh Memory

Refresh memories are used to store intermediate and final results of image processing. Memories can be combined to produce color images using red, green, and blue separations. A typical RAM memory architecture might be comprised of several 512 x 512 x 1 bit planes cycling at 10 megapixels/sec per bit. Four such planes packaged on a card four deep provides 1 million bits of RAM. A full 512 x 512 x 8 monochrome image requires two cards as in Figure III-4. Such a configuration defines an image plane, 64 of which would represent a 4096 x 4096 image in refresh at one time (see Figure III-5). The architecture is configured so that each bit plane can be interpreted as a raster graphics plane; and in addition, the read-out addressing can be configured so that the image planes can be stacked to represent true color (3 image planes or 6 cards), monochrome with 4 graphics (1½ image planes or 3 cards), or true color with 4 graphics (3½ image planes or seven cards). This final configuration is presented in Figure III-6.

Because all of the bit planes are in synchronization, it is possible to offset the (x,y) starting location, thereby providing

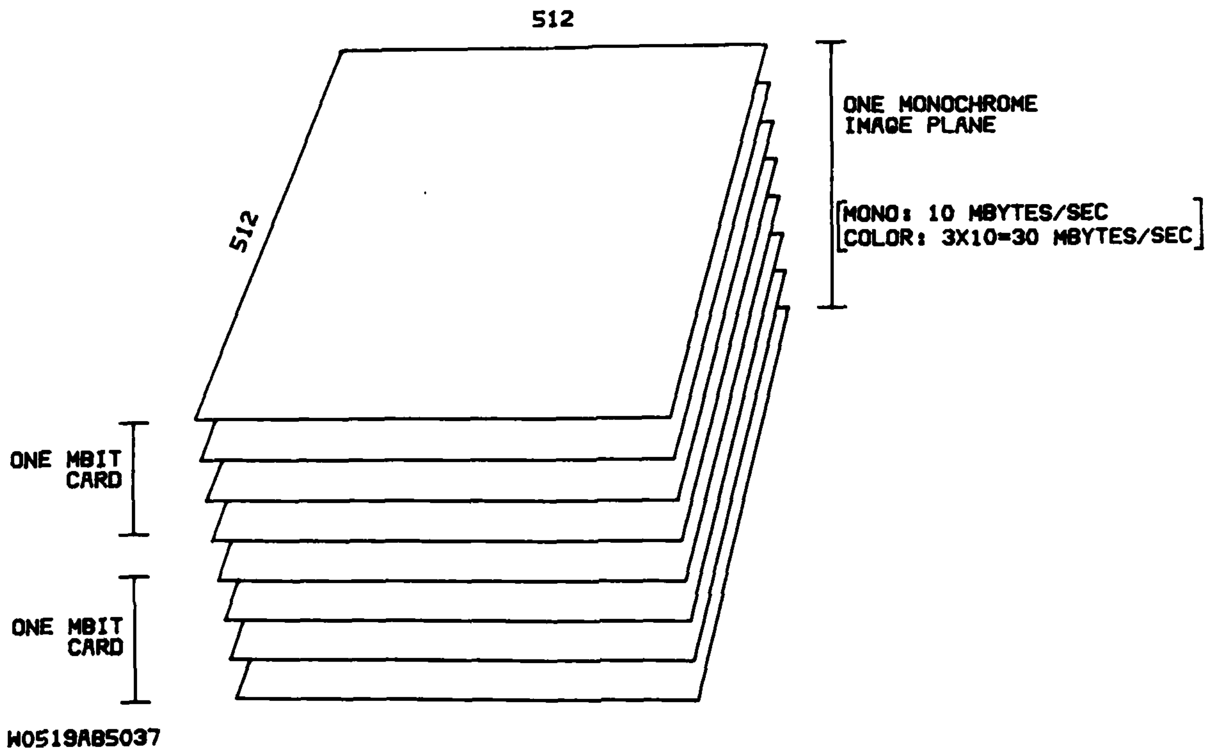



Figure III-4. Single 512 X 512 X 8 Bit Image Memory

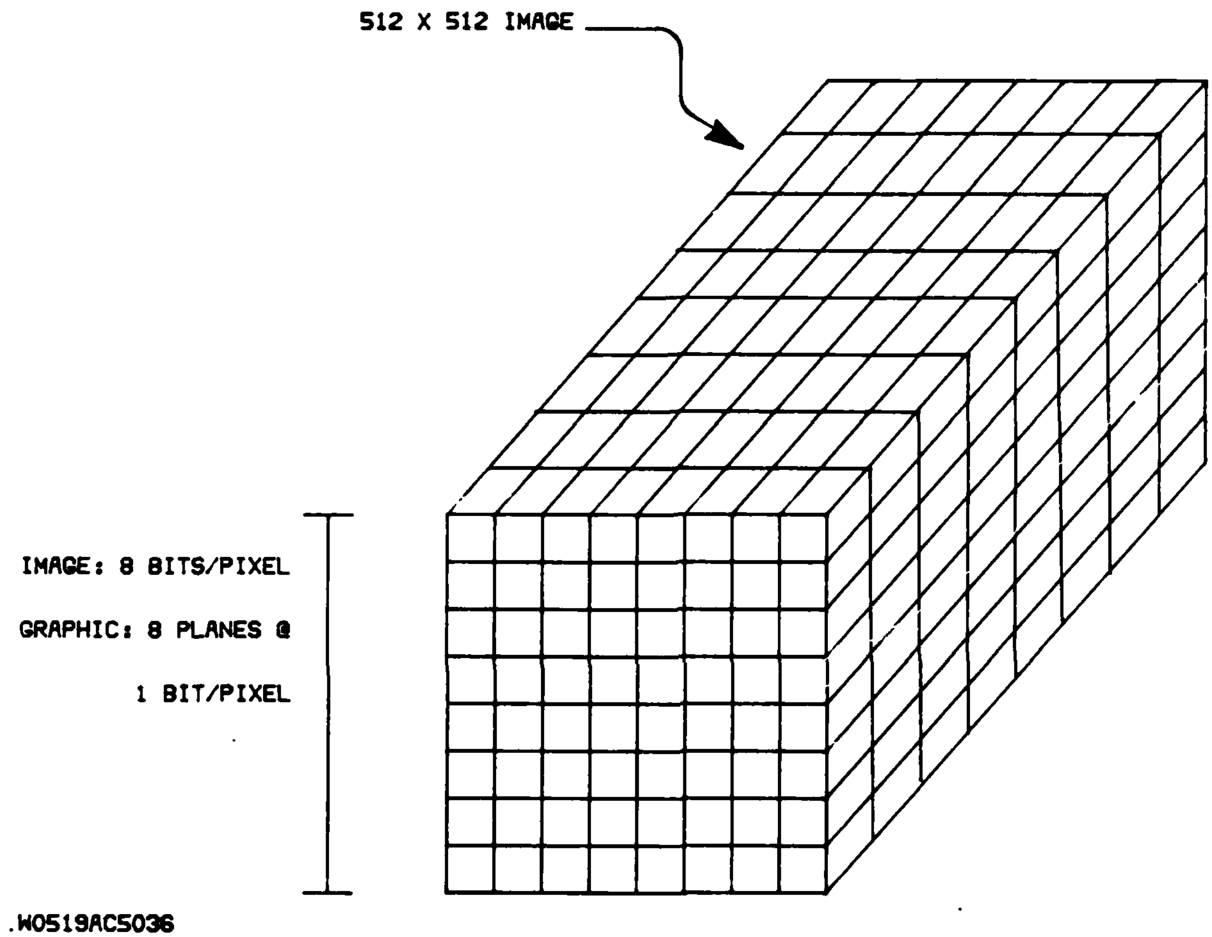



Figure III-5. RAM Memory Architecture

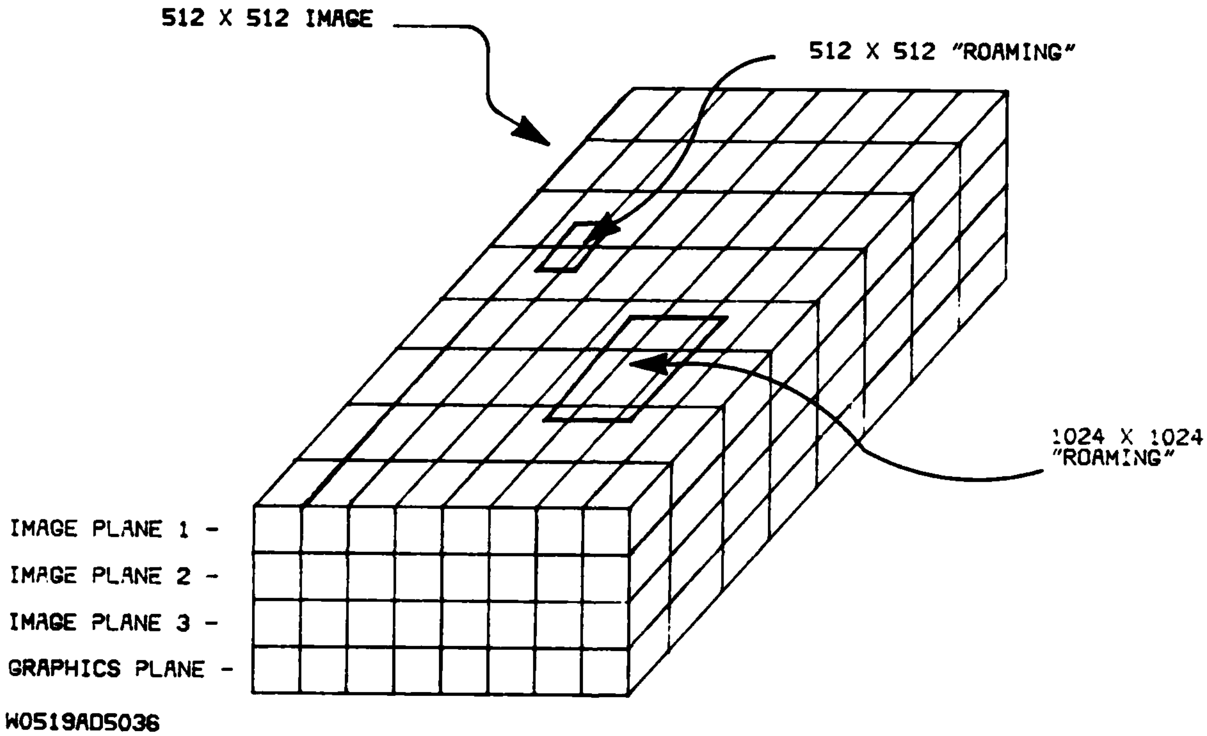



Figure III-6. "Roaming" RAM Memory

arbitrary "roaming" capability. For a 4096 x 4096 image data base a total of 64 separate 512 x 512 image planes would be "roamable." In general, the memory configuration can be dynamically allocated resulting in a software defined data base with a depth as follows:

Data base depth = (# of vertical images) x 512 x (# of horizontal images) x 512 x (# of 4-bit deep image/graphics planes) x 4

The user will typically have the option of either roaming the data base through a 512 x 512 or 1024 x 1024 window (see Figure 6) that is either in monochrome (8-bits deep) or color (24-bits deep). From the monitor viewpoint, thirty frames of imagery per second are presented to the viewer. This represents refresh rates as follows:

| Display Window | Refresh Rate |
|---|---|
| 512 x 512 Monochrome | 7.5 Mb/Sec |
| 512 x 512 Color | 22.5 Mb/Sec |
| 1024 x 1024 Monochrome | 30 Mb/Sec |
| 1024 x 1024 Color | 90 Mb/Sec |

In most systems, refresh memories will be controlled by a minimum of three separate I/O cards.

a) Image Refresh I/O Control

This card takes advantage of the dual porting of the memory (the ability to read from and write to memory in the same frame time) to allow arbitrarily slow loading into random locations at random rates thus allowing the refresh memory to be asynchronous loadable from any external device running slower than the particular refresh rate.

b) Image Refresh Data Base Control

This card defines the offset values on each memory card for arbitrary roaming. It employs a raster addressing circuit that addresses the starting pixel (usually upper left hand corner) of each image plane at the roam location. Because this address can change as often as 30 times/second, one could roam relatively quickly with respect to viewer response time. Roam addressing is changed during vertical blanking of the display; therefore, a smooth and continuous roam is possible without interference to the displayed scene. This refresh data base control card also provides for magnification (zoom) at any arbitrary location in the memory by pixel replication. The zoom factor can be in a variety of increments (i.e., 1x, 2x, 3x, 4x, etc.), and horizontal and vertical zoom may be independently controllable. The process of replication is implemented in real-time by slowing the memory down by a particular factor. For instance, a 2x replication slows memory by a factor of two, 4x by a factor of 4 and so on. Then, each pixel is read twice, or four times, or n times depending on the replication factor. Although the memory scan is slowed down by a given factor, the reading rate brings the output data up to the original digital TV refresh rate, accomplishing the zoom instantly. Because this control card also defines the roam offset, both roam and zoom can be combined quite easily here.

c) <u>Frame Write Card</u>

This card allows data entry into the refresh memory at a particular rate (for example: 80 M bits/sec) for a burst of 1/30 second. This allows the memory to be used as a video frame grabber, but more importantly it allows a feedback path back into the refresh memory for processed results coming out of the pipeline image processor. This feedback path allows truly iterative full frame array processing on the imagery where each iteration takes only 1/30 second for implementation and the progress of the algorithm is visually available.

3) <u>Pipeline Image Processor</u>

The pipeline processor functions are hard wired tables connected to each possible image and graphics plane in the refresh memory as illustrated in Figures III-7 and III-8. For example, given a 4096 x 4096 pixel image (64 image system), a total of 64 tables, each of which is an 8 x 8 bit (256 level by 256 level) function memory, can be loaded with arbitrary values under interactive operator control. These tables are loaded during horizontal and vertical retrace periods, thereby

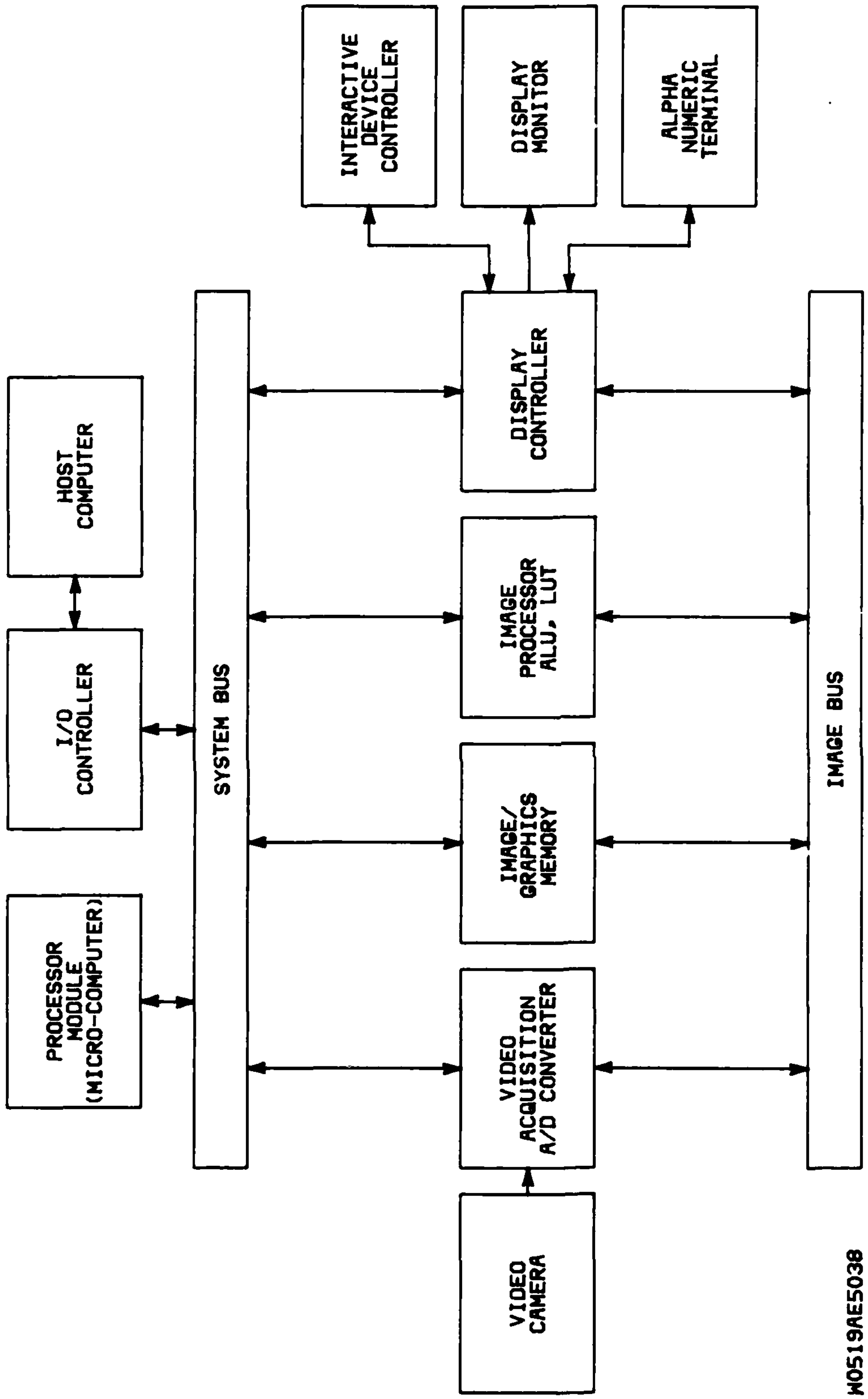



Figure III-7. Monochrome System Architecture

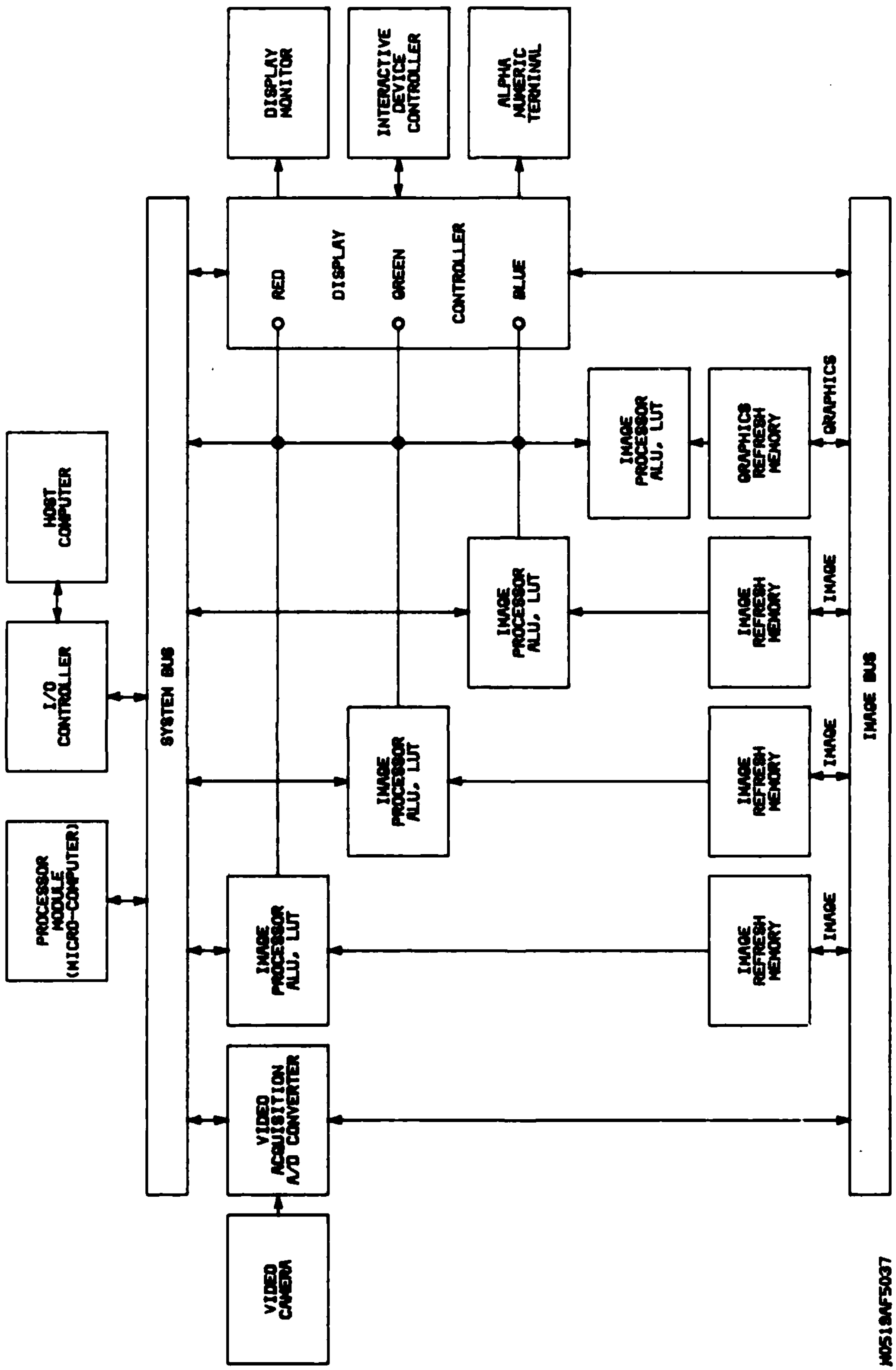



Figure III-8. Color System Architecture

guaranteeing no visual break-up of the displayed imagery. Then, an electronic selector switch provides for selection of at least three simple outputs from the 64 inputs (graphics or images). Monochrome is handled by switching one image to all three color outputs as in Figure III-7. When true color output is desired, the selector switch (which is updateable every 1/30 second) selects any three of the possible 64 images to be defined as any combination of red, green, and blue channels for final display (see Figure III-8). Thus, monochrome to color dynamic memory allocation is done simply at the parallel pipeline selector switch, and because all image planes pass through function memories, one then has 24-bit color correction control implemented by the respective three 8 x 8 function memory tables.

Most image processing algorithms can be decomposed into a sequence of primitive operators involving point, ensemble, and spatial processing for efficient execution in a recursive pipeline structure like the one in Figure III-9. In this structure, the microcomputer system initializes the pipeline controller and one of the processors. An original image is read from image memory, and the pixels are directed to a processor. The processed image is then stored in a work space in image memory.

The point processor performs operations on single pixels, a pixel and a constant, and pairs of pixels. Single pixel operations are executed by a look-up table (a bit table that can be dynamically programmed for linear or non-linear point-processing operations to modify the contrast of an image or to form its magnitude, square, square-root etc.) Two-operand operations are executed by an arithmetic logic unit (ALU) (performing addition, subtraction and logical combination of image pairs, or an image and an arithmetic or logical constant). The standard arithmetic and logic functions are:

A minus B
B minus A
A plus B
A xor B

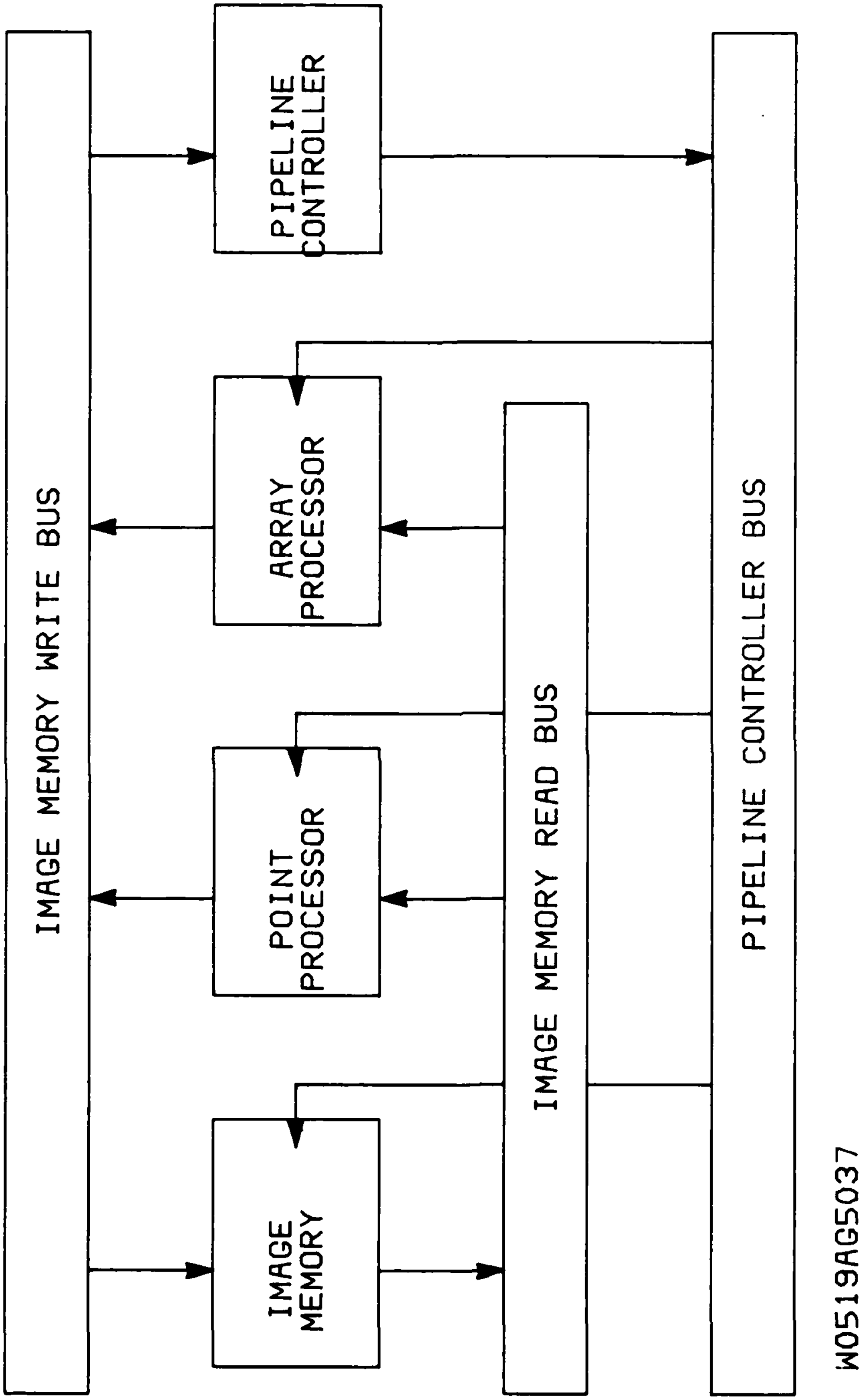



Figure III-9. Pipeline Image Processor

A or B

A and B

The array processor performs pixel by pixel multiplication of a pair of images, temporal or multi-spectral combination of image arrays, and the standard 3 x 3 pixel spatial convolution (more on this later).

The pipeline controller, the principal control element of any image processor, is a discrete logic state machine that will accept a microcode control program from the microcomputer system. It then sequentially steps through its stored instructions to grant bus access to memories and processors. On completion of a process, the pipeline controller will generate an interrupt to the microcomputer system, which then resumes program control.

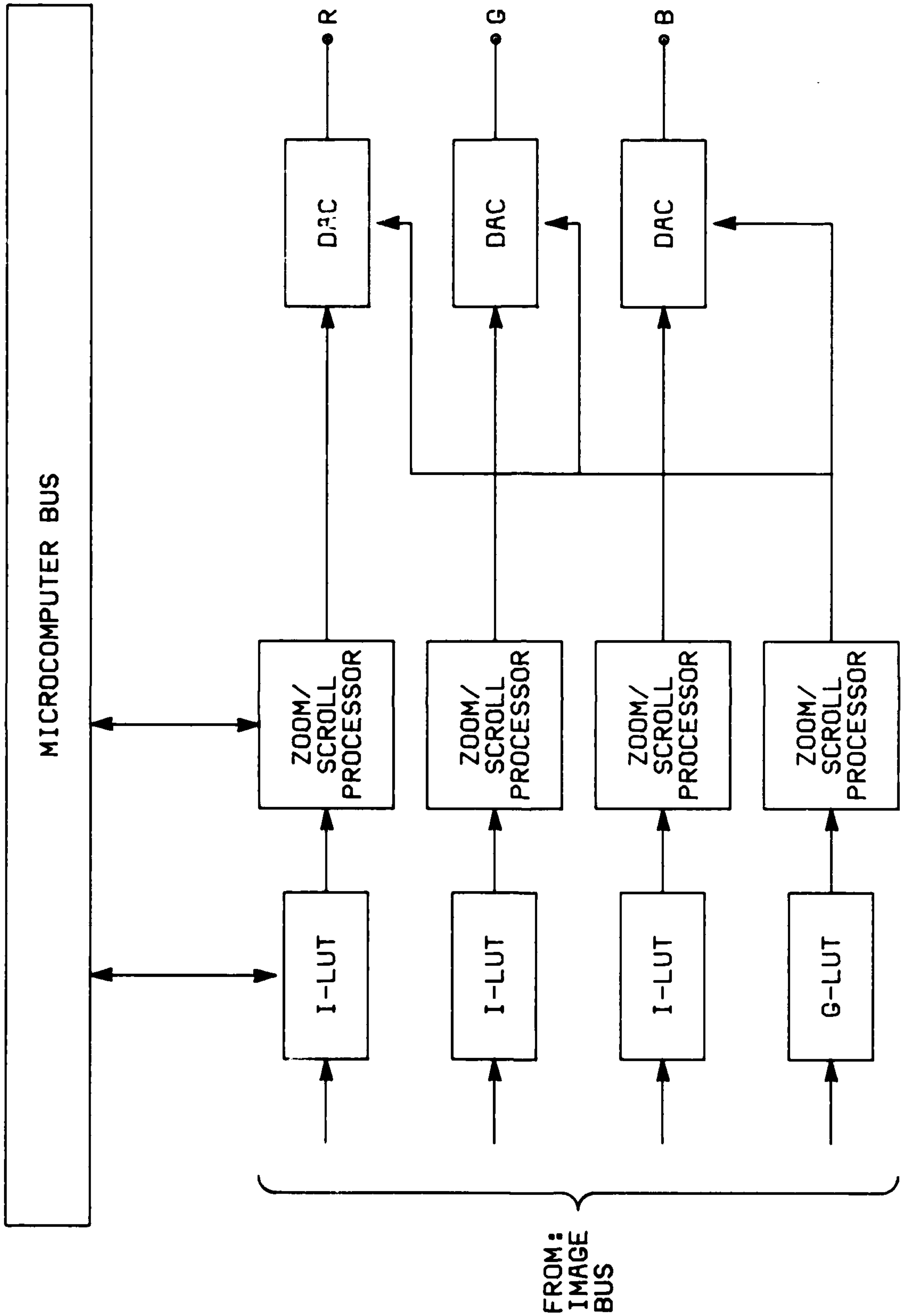
4) Display Systems

a) Display Controller

A typical display controller (shown in Figure III-10) contains three look-up tables (LUT) and three 8-bit digital to analog converters (DAC); the converters create three independent monochrome images or a single red, green, or blue color image for video generation. In addition, the display controller will usually support independent graphic overlay planes that can be individually programmed to display a variety of colors. Look-up tables perform graphic manipulation as well as image amplitude functions.

Generally, a display controller consists of a single board that can be configured for various video formats. Video formats consist of three basic parameters: refresh rate, interlace scan, and number of display lines. For example, here are three common video formats:

30 frames/sec., 559-line interlace scan, 512 display lines;
30 frames/sec., 525-line interlace scan, 480 display lines;
25 frames/sec., 625-line interlace scan, 512 display lines.




W0519AH5037

Figure III-10. Display Controller

Display manipulation within the controller can also include scrolling, panning or roaming, and independent (x,y) zooming functions. However, display manipulation is limited. In order to achieve a high degree of display manipulation and alteration, a multiple-board system called a display processor must be utilized.

b) Display Processor

The display processor consists of an interconnection of display buffers, display generators, and graphics generators. Figure III-11 illustrates the interconnection for a color image and a red/green/blue color graphics display. The modules can also be configured for pseudocolor image and color graphics display or for monochrome image and monochrome graphics display (Figure III-12).

Figure III-13 contains a functional diagram of the display buffer, which consists of a memory that stores a pixel image (i.e., 512 x 512 x 16 bit, 1024 x 1024 x 16 bit etc.) read in at the pipeline rate from the image bus. Then, the image data are read out at digital video rates from the buffer memory to the display generator through a zoom/scroll processor that performs zoom (integer magnification: 2X, 4X, etc.) and non-wrap around scrolling.

A graphics generator, shown in Figure III-14, provides graphics overlay planes, cursors, alphanumeric text characters, and graphics generation capability. Graphics can be superimposed over the entire image or just a portion of the image. Cursors are independent and programmable to various sizes. Character fonts are programmable and each text character generally can be programmed to blink, be underlined, or be inverted.

A graphics generator will commonly contain its own microprocessor, and possibly an arithmetic co-processor in conjunction with a special purpose controller that rapidly draws vectors and arcs and performs high-speed filling of closed contours. Invariably, the processor will also support several RS-232-C ports which allow for the connection of various interactive control devices (i.e., joystick, trackball, data tablet, mouse, light pen, etc.) The graphics bits are output at each

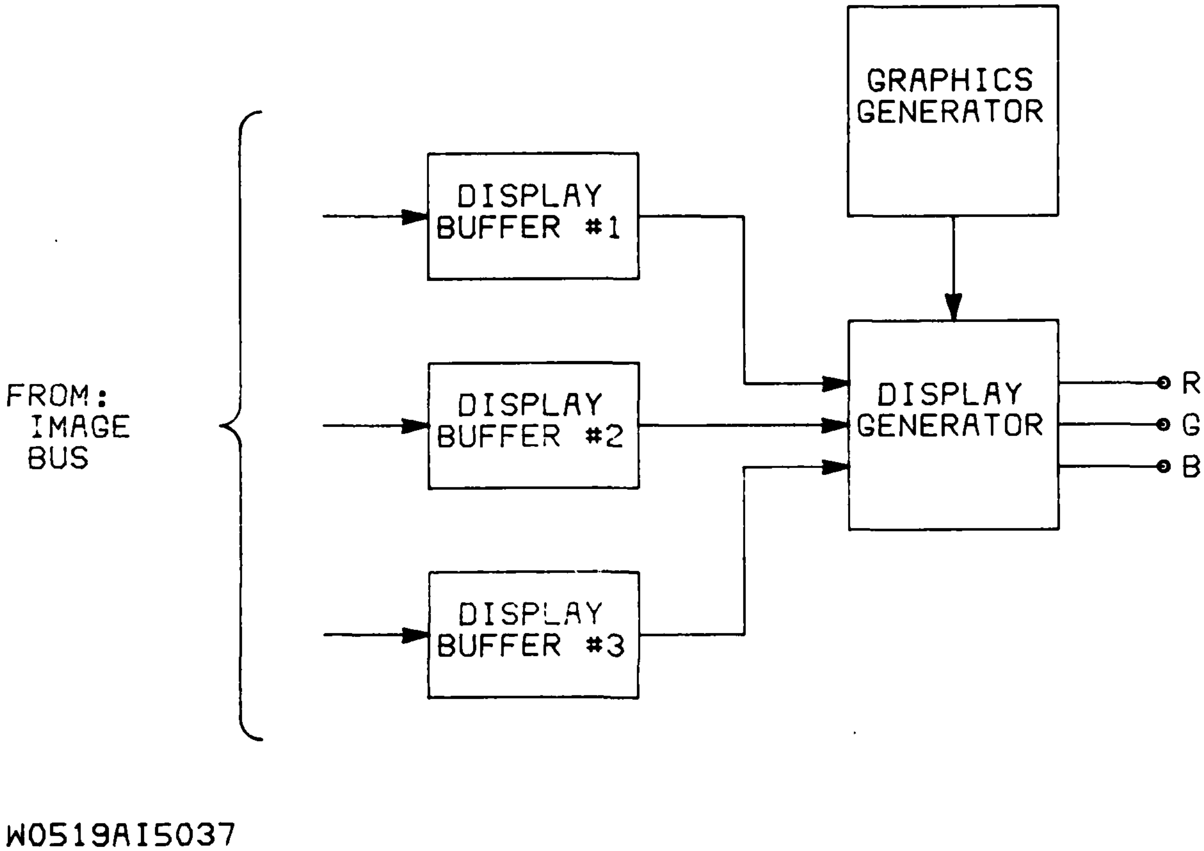



Figure III-11. Connection for Color Image, Color Graphics and Pseudocolor Capability

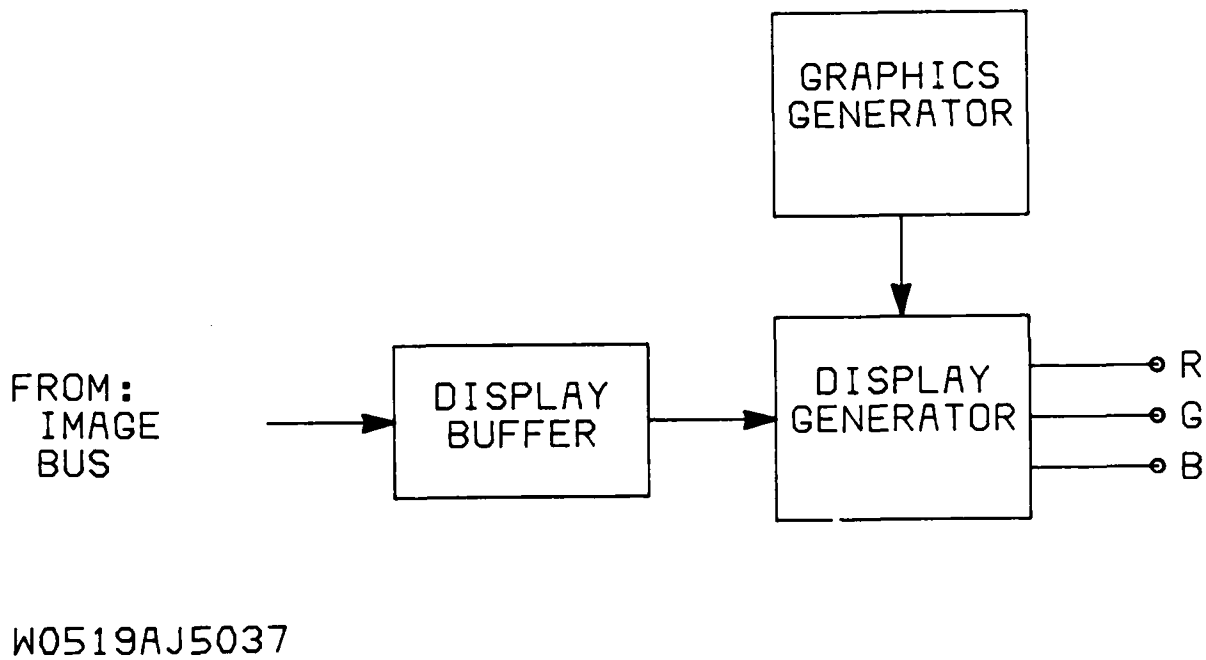



Figure III-12. Connection for Monochrome Image, Color Graphics and Pseudocolor Capability

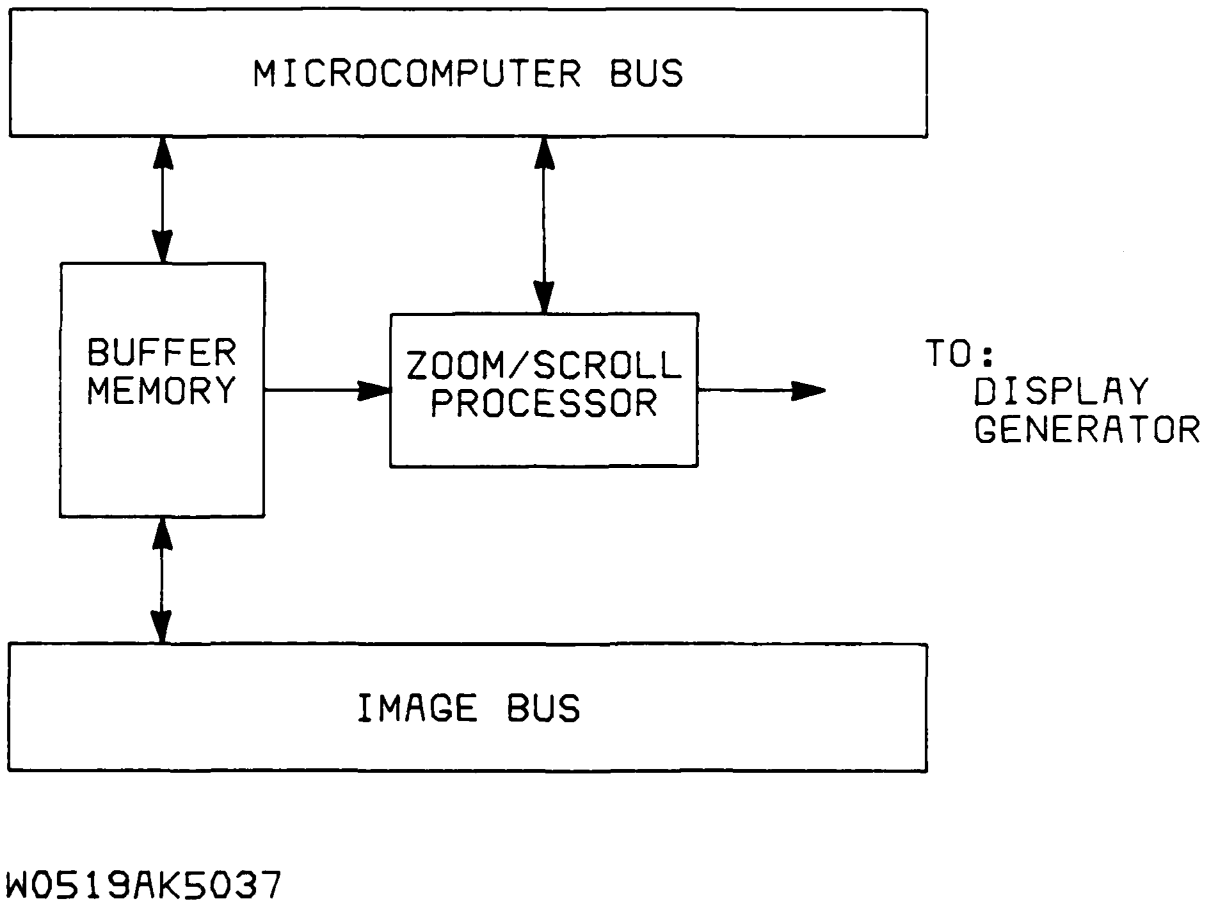



Figure III-13. Display Buffer

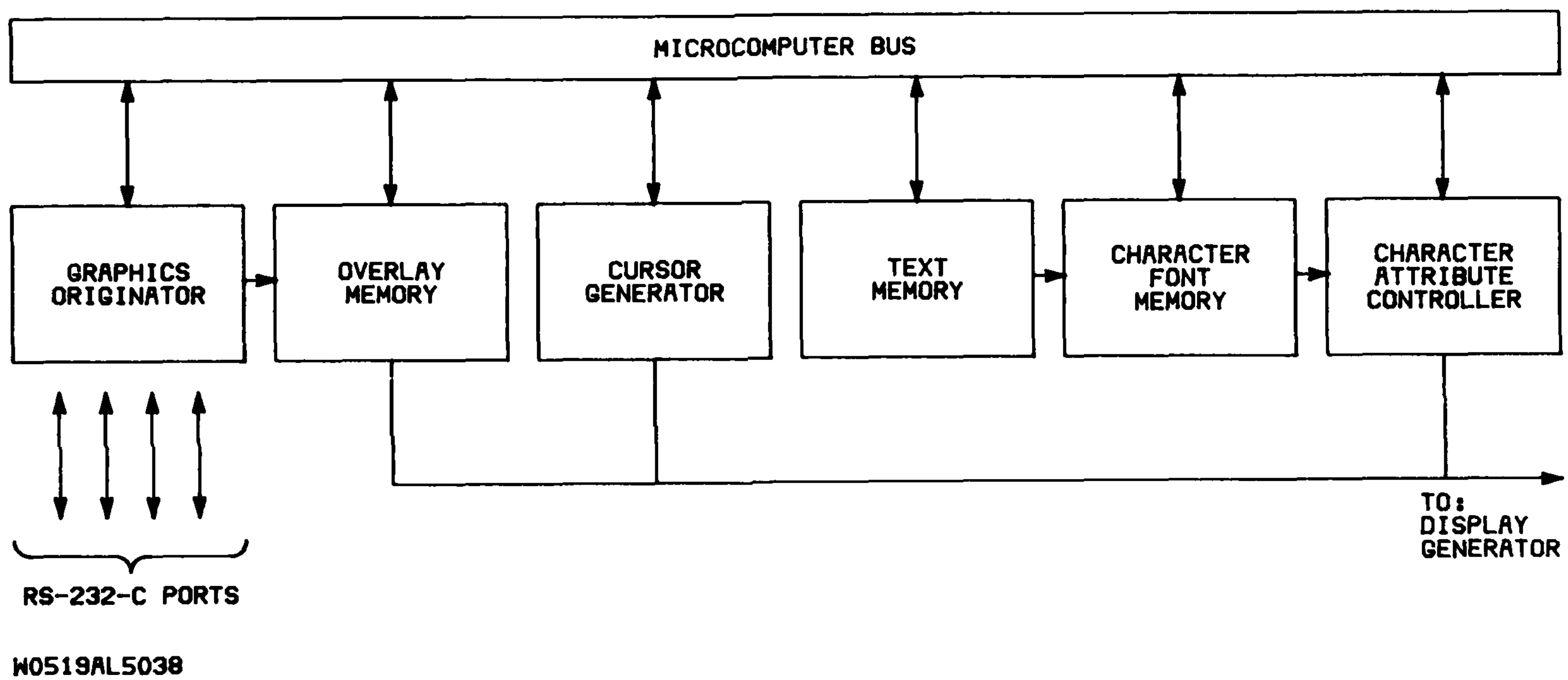



Figure III-14. Graphics Generator

pixel from the overlay memory(s), the cursor generator(s) and the character attribute controller at digital video rates into display generator(s).

Each display generator channel (see Figure III-15) contains a set of cascaded look-up tables which drive digital to analog converters. In a display buffer, a graphics look-up table (G-LUT) merges image-associated graphics data derived from main image memory with display associated graphics from the graphics generator. The pixel amplitude data, representing pixel luminance or color component values, enter the image look-up table (I-LUT) for amplitude modification, contrasting, windowing, etc. Finally, pixel amplitude and graphics data are joined in a merging look-up table (M-LUT) to create the input for the digital-to-analog converter (DAC). As in the display converter, the display generator can be software-switched between various video formats.

d. Display Devices

Although digitizing is a rather straightforward process, the conversion of the data into pictorial form is far more complex, and many devices are available for this purpose. The reason for the complexity is that the input to the device not only has to do a digital to analog conversion but a certain amount of data processing as well. First, however, one must distinguish between "hard copy" and "cathode-ray tube" (CRT) devices. CRT displays do not have to follow a strict order of display, but there is a need to repeat the process so that the displayed image may be viewed for any length of time. This process is called "refreshing the CRT". For hard copy, usually only one pass is made and therefore the input must be sorted to generate commands in an orderly fashion to direct the motion of the writing instrument.

The CRT devices available today contain a microprocessor and a memory buffer. The host computer loads the image information into the buffer and then the microprocessor reads the buffer and displays the contents as many times as necessary to generate a particular visual impression. The most primitive type of instruction stored in the buffer might be of the form: write (x,y,z). This instruction positions the

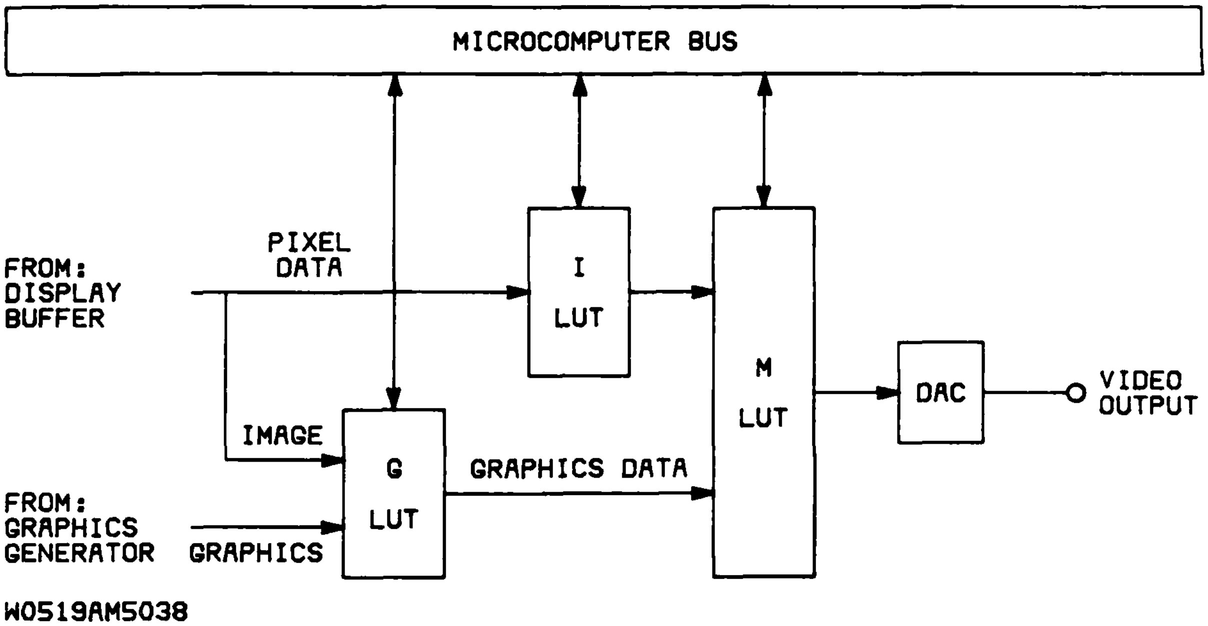



Figure III-15. Display Generator Channel cathode-ray (electron beam) at point (x,y) of the screen to produce a bright spot with intensity or color corresponding to z. Therefore, for a monochrome display, at least three D/A converters are needed: one for each of the x,y and z values. For color displays, the minimum number of D/A converters is five: two for coordinates (x,y) and three for each of the colors (z is now a vector of colors R,G,B).

Although the difference between CRT and hard copy might be important to the user, it is not very significant in terms of the conversion process or in the format of the memory buffer. The major distinction is between vector and raster graphics. In fact, much of the special preprocessing associated with hard copy devices involves the conversion from vector to raster representations and vice versa. The particulars of this subject will be addressed in a later discussion.

3. Commercially Available Image Processing Hardware

The survey of hardware technology is now concluded with lists of commercially available products, all of which are suitable for the most basic image processing. Depending on the specific requirements of the task, any of these might be appropriate for developing the automated camouflage pattern generation system.

a. Image Processing Systems

1) Turnkey Systems

| PRODUCT | MANUFACTURER | VENDOR | PRICE ($) |
|---|---|---|---|
| AUDRE | Audre, Inc. | Audre, Inc. | |
| AY52000 | Aydin Computer | Aydin | 35-70K |
| Digital I.P. | Vicom Systems | SSS Corp. | 30-150K |
| Eyecom III | Loge/Spatial Data | | 30-70K |
| PIC | Wang Laboratories | Wang | |
| System One | Via Video, Inc. | SMS Data Sys. | 45K |
| Vision One | 3M Inc. | Comtal Corp. | 30-130K |

2) Board-Level Products

| PRODUCT | MANUFACTURER | VENDOR | PRICE ($) |
|---|---|---|---|
| Datacopy 110 | Datacopy Corp. | Datacopy | 1000 |
| Ip-512 | Imaging Tech. | Dig. Image Sys. | 5-15K |
| PC-Eye | Chorus Data Sys. | Chorus | 500-1000 |
| PC-Vision | Imaging Tech. | Dig. Image Sys. | 3000 |

b. Commercial Video Cameras

| PRODUCT | TUBE | RESOLUTION | MANUFACTURER | PRICE ($) |
|---|---|---|---|---|
| AUDRE | CCD | 1720 x 2592 | Audre, Inc. | (*) |
| Datacopy 610 | CCD | 1728 x 2846 | Datacopy Corp. | 7850 |
| Eyecom Scanner | 1" Vidicon | 525 Lines | Logetronics | (*) |
| VC-1000 | 2/3" Vidicon | 500 Lines | Chorus | 270 |
| VC-2000 | 2/3" Vidicon | 600 Lines | Chorus | 368 |
| VC-3000 | 2/3" Vidicon | 700 Lines | Chorus | 510 |
| VC-4000 | 1" Vidicon | 800 Lines | Chorus | 1240 |
| VC-5000 | 2/3" Vidicon | 700 Lines | Chorus | 1199 |
| VC-6000 | CCD | 403 x 512 | Chorus | 1915 |

(*) These cameras are included in the price of the basic image processing system. Typically, these items may not be purchased separately.

c. Commercial Hard Copy Devices

1) Plotters

| PRODUCT | CONTACT | PRICE ($) |
|---|---|---|
| AlphaPlot | Alphamerics (213) 709-1155 | |
| CalComp M81/M84 | Calcomp (800) 556-1234 EXT 156 | |
| Gould Colorwriter | Gould (216) 361-3315 | |
| Hewlett-Packard | HP (800) 367-4772 | |
| 7470,7475,7220,7580,7585 | | 2-5K |
| Houston Instrument | HI (512) 835-0900 | |
| DMP-29,40,41,42 | | 1-4K |
| IBM XY 749/750 | IBM (800) 447-4700 | |
| Roland DXY-800,101 | Roland (213) 685-5141 | |
| Strobe 100,200,260 | Strobe, Inc. (415) 969-5130 | 500-1200 |
| Sweet-P | Enter Co. (619) 450-0601 | 595 |
| Western Graphtec | Western Graphtec (800) 854-8385 | 1200 |

2) Printers

| PRODUCT | CONTACT | PRICE ($) |
|---|---|---|
| Electrostatic | Audre, Inc. (714) 476-2214 | |
| Semiconductor Laser | Audre, Inc. (714) 476-2214 | |
| Helium-Neon Laser | Audre, Inc. (714) 476-2214 | |
| Argon Laser | Audre, Inc. (714) 476-2214 | |
| Epson Dot Matrix | Epson, Inc. (213) 539-9140 | |
| IBM Dot Matrix | IBM (800) 447-4700 | |
| Ink-Jet | HP (800) 367-4772 | |

## C. SOFTWARE TECHNOLOGY

### 1. Introduction to Software Technology for Image Processing

Generally, software for state-of-the-art image processing systems consists of three major components or layers. These three layers include some type of command interface module, an applications library, and a variety of device drivers. Together, the three layers provide a user with the capability for both high level interactive image processing and creation of high-performance customized applications.

In this section, the three basic layers are described. Then, in subsequent sections, the command layer and the library layer are surveyed in greater detail. Because of their relative importance to an image processing system, numerous command language modules are defined and a series of typical applications subroutines are investigated, processing techniques are reviewed, and algorithms are analyzed.

#### a. Command Interface Module

A command interface module is responsible for the interactive mode of image processing. It interprets commands from the keyboard and, in order to execute commands, instigates calls to the applications library. Typically, image processing system commands are entered using some form of a mnemonic and an associated argument list. In some cases, images are referenced by logical number and arguments are entered as constants and/or symbolic values. For example, the general form of a command might be:

```
COMMAND image#,...,image#   image#,...,image# (param,...,param)
```

In this case, the symbolic value, " ", is interpreted as a memory operand directing the input and output of images. Therefore, following the general form, an actual command to multiply an image stored in memory 1 (referenced as image#1) by a constant and place the result in memory 3 (referenced as image#3) might look like this:

```
MULTIPLY 1   3 (3.14159)
```

In this command string, the parameter (3.14159) denotes the value of the constant. In reality, a command statement would most likely be a three or four letter mnemonic like "MUL" rather than a full-word expression as shown in the above example. Of course, these particulars vary from system to system. For this reason, command modules have been surveyed in generic terms such as "Color Commands" and "Add Commands" rather than attempting to isolate the various mnemonics utilized by a particular manufacturer.

b. Applications Library

The applications library will typically consist of several hundred subroutines performing image processing arithmetic and algorithms. These subroutines will handle basic ALU operations, spatial filtering, convolution operations, image analysis, geometric transforms, graphics, input/output, point transformations, color operations, and a variety of other functions.

This layer of the software package is often referred to as the foundation for the entire image processing system because it links the user application or the command interface module to the hardware through the device driver layer. In order to implement image processing algorithms, the applications library translates a simple subroutine call into the proper sequence of hardware and software controlled operations. Furthermore, it provides subroutines for direct access to device drivers, so that a user application program can manipulate the image processing system without having to resort to assembly or machine language programming.

Because each subroutine conforms to a standard interface, the applications library can usually be easily extended by the user. In other words, specialized commands may be added to the system by writing new subroutines in the library and then labeling it with a personalized command descriptor or mnemonic.

c. Device Drivers

At the bottom layer of an image processing software system, the device drivers are found. A device driver is an assembly language program that manages all of the system devices (i.e. printers, plotters, communications ports, etc.) and provides a powerful, but simple, parameter block interface to the user software. In general, the user need not be concerned with the device drivers. However, for demanding applications which require the maximum processing speed, this layer makes it possible to call the drivers directly, minimizing the overhead for each operation. In fact, the user can extend the drivers themselves by writing customized assembly code for special types of devices with or without the use of system source code (high-level language).

2. Survey of Command Language Modules

The following is a list of the generic classes of commands common to most command interface modules. Of course, depending on the manufacturer, each command interface module is slightly different. This list represents the "minimum", as far as commands, that would be necessary for an effective automated camouflage pattern generation system. Therefore, this list should assist in isolating the image processing systems that are suitable.

a. Add Commands - Add/subtract elements of display.

b. Administrative Commands - Modify parameters, memory, and code; and display parameter values.

c. Assign Commands - Establish relationships between the user and various hardware options.

d. Clear Commands - Set zero values into various system components.

e. Color Commands - Change the color for specified system components.

f. Define Commands - Specify sizes for images, windows, and targets.

g. Display Commands - Specify which system component appears next on CRT display.

h. File Commands - Initiate transfer from attached host processor to system microprocessor.

i. Fill Commands - For an irregular enclosed area traced into a graphic plane, turn on all pixels inside the target-designated area to create a graphic control mask.

j. <u>Function Commands</u> - Manipulate a brightness transformation processor, known as function memory.

k. <u>Graphic Commands</u> - Manipulate the contents of the available 1-bit graphic overlay planes.

l. <u>Image Commands</u> - Manipulate the contents of image refresh memories.

m. <u>Initialize Commands</u> - Create initial conditions/values for various system components.

n. <u>Pseudocolor Commands</u> - Alter contents of the pseudocolor memories.

o. <u>Release Commands</u> - Reallocate various system resources.

p. <u>Roam Commands</u> - Enable real-time translation of the display window over the image data base.

q. <u>Roll Commands</u> - Enable interactive modifications of various processing memories.

r. <u>Set Commands</u> - Allocate specific system resources.

s. <u>Smooth Commands</u> - Provide interactive pixel averaging to smooth out blemishes.

t. <u>Table Commands</u> - Affect contents of temporary storage tables.

u. <u>Tint Commands</u> - Interactively affect displayed color values for truecolor image.

v. <u>Unassign Commands</u> - Terminate relationships established in assign commands.

w. <u>Miscellaneous Commands</u> - Commands unrelated to above modules.

3. <u>Software Controlled Image Processing Techniques/Algorithms</u>

Applications libraries often contain sophisticated display functions, applications software, and image capture utilities. The list below contains the most useful (and invariably the most common) software controlled techniques for sophisticated image processing systems. In subsequent paragraphs, each technique is described, and in some cases algorithms for these functions are discussed.

- Thresholding/Contract Enhancement
- Halftoning/Convolution
- Pseudocoloring
- Edge Detection
- Filtering
- Interpolation
- Minification
- Rotation
- Rectification
- Recenter
- Zoom
- Pan/Roam
- Image Combination

a. Thresholding/Contract Enhancement

Thresholding software reduces the number of bits per pel gray scale, thereby converting the image to high contrast. The threshold setting is usually programmable from the software level to any one of the possible levels of gray provided by a particular video camera. Furthermore, if the standard bit per pel of gray scale is desired, the threshold function is switched off by software command. By thresholding, one can compensate for an improperly exposed image by stretching and modifying the gray scale range over the image. Contrast enhancement can also be used to make vague features of an image stand out.

b. Halftoning/Convolution

Halftones are small dots of ink (or pixels on a monitor screen) of varying size and shape that, when viewed at a distance, give a visual approximation of a particular image such as a photograph. Convolution is the method by which the halftoning software produces different output image schemes. Convolution produces an output image whose pixels values are the sum of local input pixel values weighted by coefficients of a convolving kernel. Convolution is useful in producing various image processing effects such as: filtering, edge detection and edge enhancement.

c. Pseudocoloring

Whether an image is captured from a color or black/white camera, the displayed image may have several colors/levels of intensity. The process of color assignment for a particular intensity or mapping a monochrome image into a color image is called pseudocoloring. This software application can be especially useful in calling attention to significant features on a particular image. Of course, the colors available to select from will be limited by the specific graphics adapter board installed in the host computer. True color may be captured and processed by using multiple exposures and color separation filters. Although current graphics board technology will not permit true color display, there are methods to photographically reproduce these images using the reverse separation techniques mentioned above.

d. Edge Detection

Edge detection produces an output image containing only the edges of the input image. This process can be very valuable for vision and pattern recognition.

e. Filtering

Filtering applies a one-dimensional recursive filter to the rows or columns of a captured image. This process can aid in correcting blurring distortions associated with the camera (scanner) or movement of the object.

f. Interpolation

This software function scales an input image using bilinear interpolation with independent vertical and horizontal continuous scale factors. Interpolation allows for much smoother image magnification than pixel replication (simply repeating pixels in order to enlargen a particular portion of an image).

g. Minification

Reduces the size of an image by skipping intermediate samples or by averaging pixel values over a local area.

h. Rotation

Rotates an image by an angle.

i. Rectification

This allows for the viewing of a two-dimensional image from any three-dimensional viewpoint. Rectification is valuable for correcting distortion in aerial photographs.

j. Recenter

Resamples an image to shift it by fractions of a pixel. Recenter is used for template matching and edge detection.

k. Zoom

Enlarges the entire screen using pixel replication. Zoom factor can be in a variety of increments; horizontal and vertical zoom are independently controllable.

l. Pan/Roam

Slides an image in one-pixel increments and provides fast access to undisplayed regions of an image. Both zoom and roam are non-destructive, real-time functions.

m. Image Combination

Combines two input images with arithmetic (ADD, SUB, MUL, DIV) or logical (AND, OR, XOR) functions.

## D. IMAGE PROCESSING FOR AUTOMATED CAMOUFLAGE PATTERN PRODUCTION

### 1. Introduction/Application

The creation of new engineering drawings on CRT (Cathode Ray Tube) graphics terminals with the help of computer-aided design and drafting (CADD) systems not only results in an intelligent, compact graphic data base, but also makes it possible to rapidly retrieve and update the data base to reflect future changes. In order to obtain these benefits of CADD, manually prepared drawings are often entered into the system via a digitizing pad, mouse system, light pen, or keyboard. Many engineering drawings exist today that are still subject to change yet do not have the benefit of computer-aided design and drafting because of the cost and difficulties associated with placing them into automated graphics systems.

To this point, this chapter has surveyed commercially available systems which optically scan and digitize engineering drawings thus accelerating and simplifying the process tremendously. However, there is an inherent problem in the video digitizing process. By digitizing a drawing, a system breaks-down the line art into millions of dots (bits) or picture elements (pixels) which are either turned on (binary 1) or off (binary 0). Once this is completed, the vector character of lines is lost since a line is now just a row of bits set to binary 1. The difficulty here is that an output device such as a pen plotter cannot interpret raster data (dot information) as lines. The obvious answer to this problem is to convert the raster data back to vector data; however, this is not a trivial task. In fact, the process of "vectorization" is a real problem for the commercially available Image Processing/CAD systems, and a problem which at this time has not been solved effectively. Without line information, the quality of the output drawing is strictly dependent on the resolution of the matrix printer. Furthermore, no matter how high the resolution, a dot matrix printer will do just as its name indicates: print dots, not lines. Granted, at a high enough resolution the dots appear to be lines, but the resolution necessary to reproduce engineering drawings is extremely expensive and will never really have the quality of an inexpensive pen plotter.

For the Automated Camouflage Pattern Generation CAD system, vectorization software will not only be helpful; it will be a necessity. In light of the fact that little in the way of vectorization software is available on the market today, it is appropriate to suggest an algorithm or a method for the vector representation of engineering drawings.

2. Computer Aided Automated Digitizing (CAAD)

The basic steps in automatic digitizing and processing of engineering drawings are scanning, thresholding, encoding, character and element recognition, editing, and storage (see Figure III-16). Summarizing, scanning of the original is required to resolve the line and character information into picture elements or pixels. The optical characteristics of each pixel are used to control detection circuitry that generates digital output in the form of gray levels. A binary representation of the drawing is obtained by thresholding the gray level

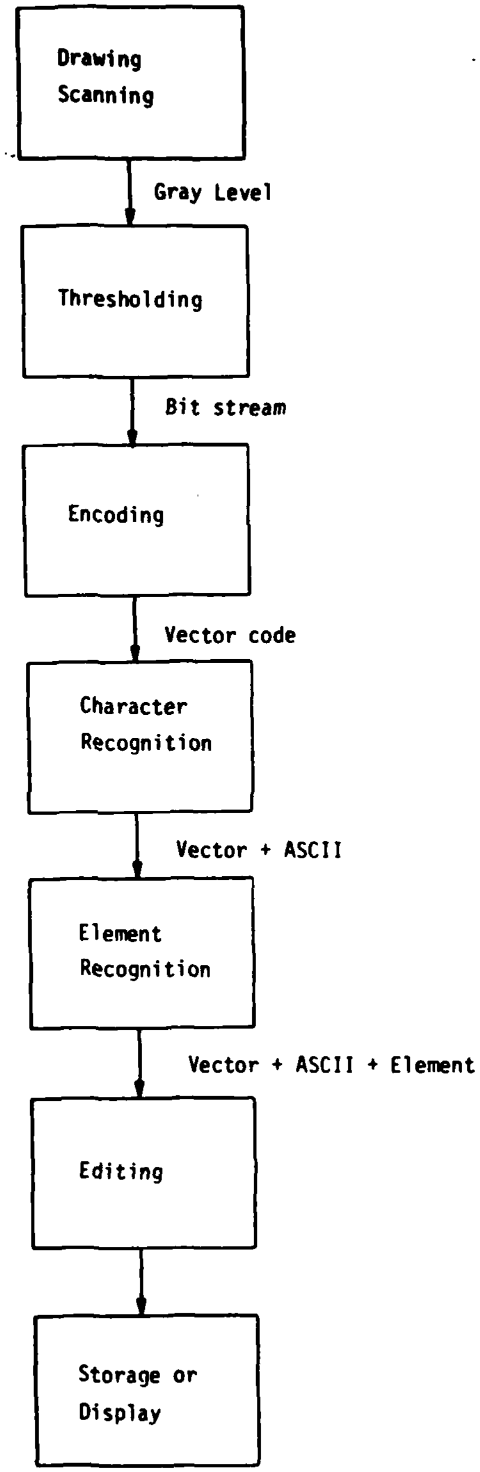



Figure III-16. The Basic Steps of Automated Digitizing image. For example, the binary representation of an "E" size engineering drawing of dimensions 34 in. x 44 in. at a resolution of 0.004 in. contains approximately 90 million bits. The purpose of encoding is to convert the enormous data stream into a compact data set. Encoding should also accomplish the task of coding raster data as if it were vector data by satisfying these goals:

- Real-time encoding/processing without intermediate data storage (low CPU processing time).
- Encoding that results in compact vector representation of engineering drawings which preserves line conventions (i.e. line thickness, dashed lines, etc.).
- Editable encoding scheme that allows for future revisions to engineering drawings (i.e. deletions, additions, etc.) via interactive display and editing on a CRT terminal.

a. Some Encoding Methods

Various algorithms to perform data reduction have been examined to identify a suitable algorithm to meet the above requirements. Four different methods will be discussed, all of which begin with raster scanning.

1) Chain Coding

The first method to be considered, chain coding, was developed for processing irregular line drawings such as weather maps and electrocardiograms. In this method, a coarse square grid with a resolution greater than the maximum line thickness is superimposed on the lines to be encoded and a set of chain points is generated by identifying the nearest grid intersections from the lines. The scheme has an inherent quantization error, depending on the grid size. Connecting the intersections produces a re-constituted line of irregular shape with a set thickness. Since chain coding does not reproduce line thickness information nor yield a compact data set because the grid size is determined by the short lines, it is not useful for encoding engineering drawings.

2) Line-Following

Automatic encoding of raster information from line drawings can also be provided by line-following codes. In line-following systems, a raster scan is used first to identify all lines on the drawing. This is followed by precise line-tracking of all lines to determine the end-points, thickness, and intersections. Line-following schemes, while generating vector notations directly, require large and very expensive scanning assemblies that are best suited for very high production environments.

3) Line Thinning

Line thinning, or skeletonization of thick patterns, is another data compression approach wherein a skeleton line representation is obtained from the bit pattern. A suitable window is centered successively at each individual data point in the binary pattern and the appropriate conditions given in the algorithm are tested. There are inherent difficulties with line thinning, however. First, line thinning algorithms cause the line thickness information to be lost. Secondly, the exhaustive analysis involved with using a window at every pixel on a large drawing requires considerable computer processing time. In addition, distortions occur at the intersections of lines which can only be corrected at the expense of extra processing time. For these reasons, line thinning is unsuitable for the digitizing of engineering drawings.

4) Message Coding

In message coding algorithms based on information and coding theory, the resulting bit stream is coded using techniques such as run-length coding, Huffman coding, Frank coding, and statistical coding. In run-length coding, each scan line is encoded by storing the locations of "0" to "1" and "1" to "0" transitions. In Huffman coding, a minimum-redundancy code is constructed such that the average number of coding bits per scan lines or message is minimized. In Frank coding, the runs of data of the same binary value on successive lines are combined to form a message. In statistical coding, data are collected on various image parameters such as run lengths for a one-dimensional coding method or connection patterns of successive elements in a contour line for a two-dimensional method, and variable length codes are constructed from the statistics.

These message coding schemes achieve significant reduction in the storage requirements for the scanned data, a valuable attribute in automatic encoding. Unfortunately, the form of representation of the data is coded messages which lack the information that is necessary for reconstructing regular line drawings. Although the methods cannot be used directly, some of them can serve as the basis for coding methods for CAAD. For example, they can be incorporated into schemes for searching the scanned data for the presence of vector and character information.

b. Vectorization Algorithm

In keeping with the criteria set forth, the following is a proposed but untested encoding algorithm to convert raster data to vector data on a CAD system designed to digitize and process engineering drawings.

1) Algorithm Concept

The process of vector encoding begins with noise removal and void filling. These processes are necessitated by the presence of extraneous microscopic dirt spots on the original drawing or minor deficiencies in the scanning and thresholding steps. In formulating the basis for criteria used in noise removal and void filling, the methods used for drafting engineering drawings must be studied. Data, such as minimum line or symbol width and minimum separation of lines on actual drawings, are needed to develop algorithms to identify noise and voids. For instance, if the minimum line width and minimum line separation are 0.012 inches, a scan resolution of 0.004 inches would translate the width and separation to approximately 3 pixels. In other words, groups of less than three pixels which are not identical can be disregarded:

Input: 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1 0 0 0

Output: 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

Noise Removal

Input: 0 0 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 0 0 0

Output: 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0

Void Filling

These steps figuratively clean-up the drawing and reduce the amount of data to be further processed. One method to accomplish noise removal and void filling might be iterated local averaging over a 3 x 3 neighborhood; however, this method will be inherently slow because of its exhaustive processing. An alternative might be to independently average the bit streams first parallel, then perpendicular, to the scanning direction. This averaging scheme will introduce greater error when compared to the 3 X 3 matrix averaging, but the simplicity allows for short processing time. Obviously, a speed/quality trade-off exists in this part of the algorithm and is dependent on the specific requirements of a particular scanning job.

The next step in the vector coding process is edge detection, whereby the edges of a black area on a given scan line are detected and the corresponding address is stored. The outline of edges on successive scan lines of a given shape on the drawing are coded, and the resulting outline contour is vectorized by piecewise linear segments. An outline consists of a number of run pixels of the same kind (i.e., "1" for black and "0" for white) on successive scan lines such that their left most edges are no more than a fixed distance (e.g. 3 pixels) apart and similarly their right most edges are no more than the same distance apart. Each outline can contain only one run on a given scan line. Next, the corresponding runs on successive scan lines are tested. If the conditions are satisfied, the run gets associated with the same outline as the corresponding one on the previous scan line. Whenever a new outline is started, the coordinates of the left most location on the run of the initial scan line are the (x,y) coordinates of one end of a vector, and the width of the vector corresponds to the width of the run. The number of runs in each outline is stored. Whenever the count is "m" runs, a vector is output by storing the (x,y) coordinates of the "mth" run, and the count is reset to one for this outline. Upon outputting a new vector, the slopes of the initial vector and the new vector are compared. If they are within a tolerance, a new longer vector is produced. This cascading of short vectors on the basis of slope comparison reduces the number of vectors significantly.

Figure III-17 illustrates the vectorization process, and Figure III-19 provides the Pascal code for the piecewise linear approximation. Figure III-18(a) shows the sequence of black runs corresponding to a curved line, whereas Figure III-18(b) shows the groupings of these runs into segments of m runs each where m equals 5. These segments are numbered in the drawing as s(1), s(2), s(3)...,s(10). Figure III-18(c) shows the vectors formed from these segments after concatenation on the basis of slope comparison and tolerance. The equations on the right side of the drawing indicate which segments have been joined to form the vectors v(1), v(2), v(3)...,v(7).

2) Advantages

Because this algorithm extracts vector notations from the scanner bit stream, it is amenable to line extraction in "real-time". The advantages are fast processing due to exploiting line structures present in engineering drawings and the availability of line thickness information, which is vital for drawing interpretation.

3) Potential Risks

This algorithm is untested; therefore, it is necessary to identify any problems which may arise if it were to be implemented. The following list summarizes the potential risks associated with this particular algorithm:

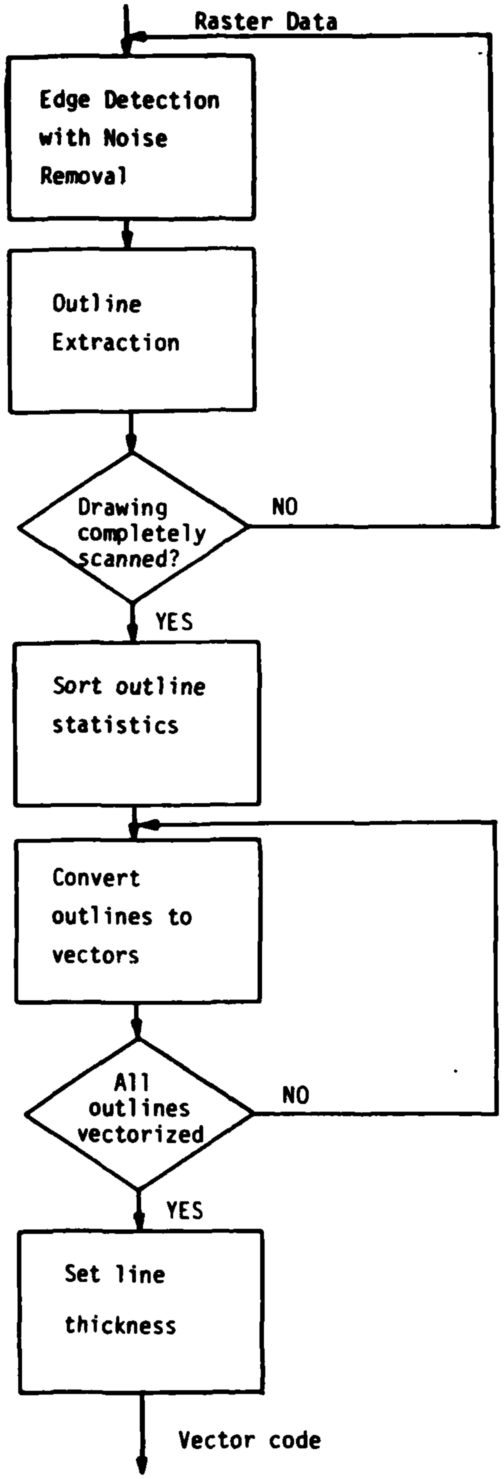



Figure III-17. Flowchart for Vector Encoding Scheme

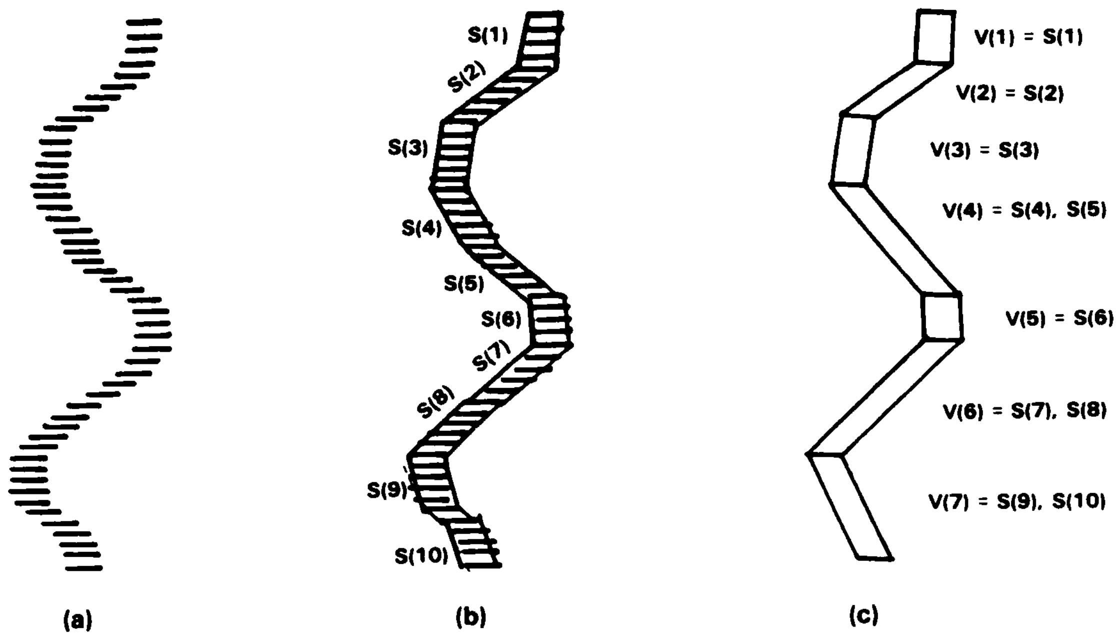



Figure III-18. Vector Encoding Process: (a) Curve Representation, (b) Run-Length Coding, and (c) Outlining of Contours

```
                then testing := false
                else begin
                     theta := dy/s;
                     if (thetamin <= theta) and (theta <= thetamax)
                        then begin

                             {point within bounds... update bounds}
                             s := delx + sqrt (rsq - dsq);
                             if (s > 0.0)
                                then begin
                                     thetamin := thetamin + ((dy-d)/s - thetamin)
                                                 * ord (thetamin < (dy-d)/s);
                                     thetamax := thetamax + ((dy+d)/s - thetamax)
                                                 * ord (thetamax > (dy+d)/s);
                                     end;
                             end
                        else testing := false;
                     end;
             if (not testing)
                then begin

                     {point outside angle bounds ...}
                     {end old line segment and begin new one}
                     x[m] := x[i-1];
                     y[m] := y[i-1];
                     dx := x[i] - x[m];
                     dy := y[i] - y[m];
                     rsq := (dx * dx) + (dy * dy);
                     m := m + 1;
                     end;
             end;
     if (not testing) and (rsq > dsq)
        then begin

             {initialize angle bounds... begin testing}
             testing := true;
             reflect := (dx < 0.0);
             if (reflect)
                then sign := -dx
                else sign := dx;
             s := sqrt (rsq - dsq) + sign;
             thetamin := (dy - d)/s;
             thetamax := (dy + d)/s;
             end;
     end;

 {end points of approximation are to match,originals}
 x[m] := x[n];
 y[m] := y[n];

 end;  {of procedure segment}
```

Figure III-19. Pascal Embodiment of Piecewise Linear Approximation Algogithm

```
procedure segment (n:integer; var m:integer; d:real; var x,y:arraytype);

{ This procedure performs a piecewise linear approximation of a planar curve. }

{                                                                             }
{ Economize on the number of line segments needed to approximate the curve    }
{ represented by the n points (x,y) subject to the restrictions:              }
{                                                                             }
{           1. The resultant m points define line segments which are          }
{              no further than distance d from the originals.                 }
{                                                                             }
{           2. The original terminal points are preserved.                    }
{                                                                             }
{ Parameters:                                                                 }
{                                                                             }
{ n         Number of original points... n > 1.                               }
{ m         Number of approximating points... m <= n.                         }
{ d         Error tolerance.                                                  }
{ x,y       Cartesian coordinates of points.                                  }
{                                                                             }

  var
        sign,dx,dy,dsq,rsq,theta,thetamin,thetamax,delx,s : real;

{                                                                             }
{ dx,dy     Cartesian components of polar distance measured from              }
{           start of line segment.                                            }
{ dsq       Square of tolerance.                                              }
{ rsq       Square of polar distance r.                                       }
{ theta     Tangent of one-half the polar angle.                              }
{ thetamin  Minimum allowable value of theta.                                 }
{ thetamax  Maximum allowable value of theta.                                 }
{                                                                             }

        testing,reflect : boolean;

{                                                                             }
{ testing   True if theta bounds are to be tested.                            }
{ reflect   True if initial angle in 2nd or 3rd quadrants implying            }
{           curve is to be reflected about the y-axis.                        }
{                                                                             }


begin
m := 2;
testing := false;
dsq := d * d;
for i := 2 to n do
    begin

    {calculate square of polar distance}
    dx := x[i] - x[m-1];
    dy := y[i] - y[m-1];
    rsq := (dx * dx) + (dy * dy);
    if (testing) and (rsq > dsq)
       then begin

            {check if point is inside angle bounds}
            if (reflect)
               then delx := -dx
               else delx := dx;
            s := delx + sqrt(rsq);
            if (s <= 0.0)
```

Figure III-19. Pascal Embodiment of Piecewise Linear Approximation Algogithm (Continued)

- Distortions in horizontal lines
- Staircasing of diagonals (if tolerance is incorrect)
- Possible requirement of up to 4Mb of RAM
- Possibility of incomplete curve closure
- Errors in radius of curvature (big vs. small curves)
- Different interpretations of identical entities
- Need for character recognition program (convert to ASCII)
- Need for element recognition program (CPU speed-up)

# BIBLIOGRAPHY

Barrett, R.C. and B.W. Jordan Jr., "A Cell Organized Raster Display for Line Drawings," Communications of the ACM, Vol 17 Num 2, pp. 70-77, February 1974.

Barrett, R.C. and B.W. Jordan Jr., "Scan Conversion Algorithms for a Cell Organized Raster Display," Communications of the ACM, Vol 17 Num 3, pp. 157-163, March 1974.

Buscher, D.J., "Digital Representations of Engineering Drawings," U.S. Army Tech Report, November, 1977.

Jain, A.K., "Advances in Mathematical Models for Image Processing," Proceedings of the IEEE, Vol 69 Num 5, pp. 502-528, May 1981.

Pavlidis, Theo, Algorithms for Graphics and Image Processing, Rockville, Md.: Computer Science Press, 1982.

Pferd, W. and K. Ramachandran, "Computer Aided Automatic Digitizing of Engineering Drawings," Proceedings of the IEEE Computer Software Applications Conference (Chicago, Ill.), pp. 630-635, Nov., 1978.

Pratt, William K., Digital Image Processing, New York, New York: Wiley-Interscience, Inc., pp 631-637, 1978.

Ramer, Urs, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," Computer Graphics and Image Processing, Vol 1, pp. 244-256, 1972.

Rosenfeld, Azriel, "Image Pattern Recognition," Proceedings of the IEEE, Vol 69 Num 5, pp. 596-605, May 1981.

Rosenfeld, A., and A.C. Kak, Digital Picture Processing 2nd edit., New York, New York: Academic Press, Inc., 1982.

Williams, Charles M., "An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves," Computer Graphics and Image Processing, Vol 8, pp. 286-293, 1978.

# CHAPTER IV
# ROBOTIC SURVEY

## A. INTRODUCTION

This chapter presents the results of a survey of robotic controlled painters to determine digital control interface requirements that will ensure maximum compatibility and flexibility of the existing automated camouflage pattern generation system with the camouflage painting facilities of Army production contractors and depots. The survey included a review of the robotics technologies presently being implemented and their applications in the industrial workplace. Additionally, numerous robotics manufacturers were contacted to obtain specific information on applicable finishing robots. This data was used to determine the interface requirements which arise from the use of robotic painters.

Industrial robots are already in use in a wide variety of applications, and the number of industries which are putting them to use is increasing every day. The advantages of using robotic painters in this application are numerous. Some of these are as follows:

(1) Productivity. Robotic painters can operate 24 hours a day with no change in speed or quality of their output. Also, in many cases, a robot can work faster and more consistently than a human.

(2) Safety. Many paints and their fumes are highly toxic to humans. The use of robots will minimize the need for human contact with this paint.

(3) Uniformity. Robots are able to operate repetitive cycles with little or no change. This will facilitate uniform painting for all vehicles.

(4) Paint savings. Since the robots can be programmed to deliver an even coating across the entire vehicle, the amount of paint which is wasted will be minimized.

The following sections will discuss several important areas of interest relating to robotic painter technology.

## B. PRESENT TECHNOLOGIES

### 1. Robots

An "industrial robot," as defined by the Robot Institute of America, is a programmable, multifunction manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. Robots differ from other types of automation in that they can be reprogrammed for different applications. An example of a typical robot is shown in Figure IV-1. An industrial robot consists of three basic components: the manipulator, the controller, and the power source.

### 2. Manipulators

The manipulator is a series of linkages capable of movement in various directions. Robots are capable of movement on several axes simultaneously. All industrial robot arms fall into one of the following classifications: cylindrical, spherical, or jointed spherical. The most complex of these arms, the jointed spherical arm configuration, operates in much the same manner as the human arm. In this configuration, the arm extends from the base, or trunk, and is jointed at the "elbow" and at the "shoulder" where the arm and base meet. The base provides rotary motion. The area which a robot can reach is referred to as its "work envelope." The jointed spherical arm configuration permits the robot to access any point within its work envelope.

### 3. Controllers

The controller actually directs the movements and operations performed by the manipulator. Depending on the degree of control, robots are classified as either non-servo-controlled or servo-controlled.

#### a. Non-Servo-Controlled

Non-servo-controlled robots are limited to discrete positions in their control of movement. A non-servo-controlled robot's sequence-controller initiates action by sending signals to control valves located on the axes to be moved. The valves open, admitting air or oil to the actuators, which drive the axes, and the axes move until they are physically constrained by end stops. Limit switches then signal the

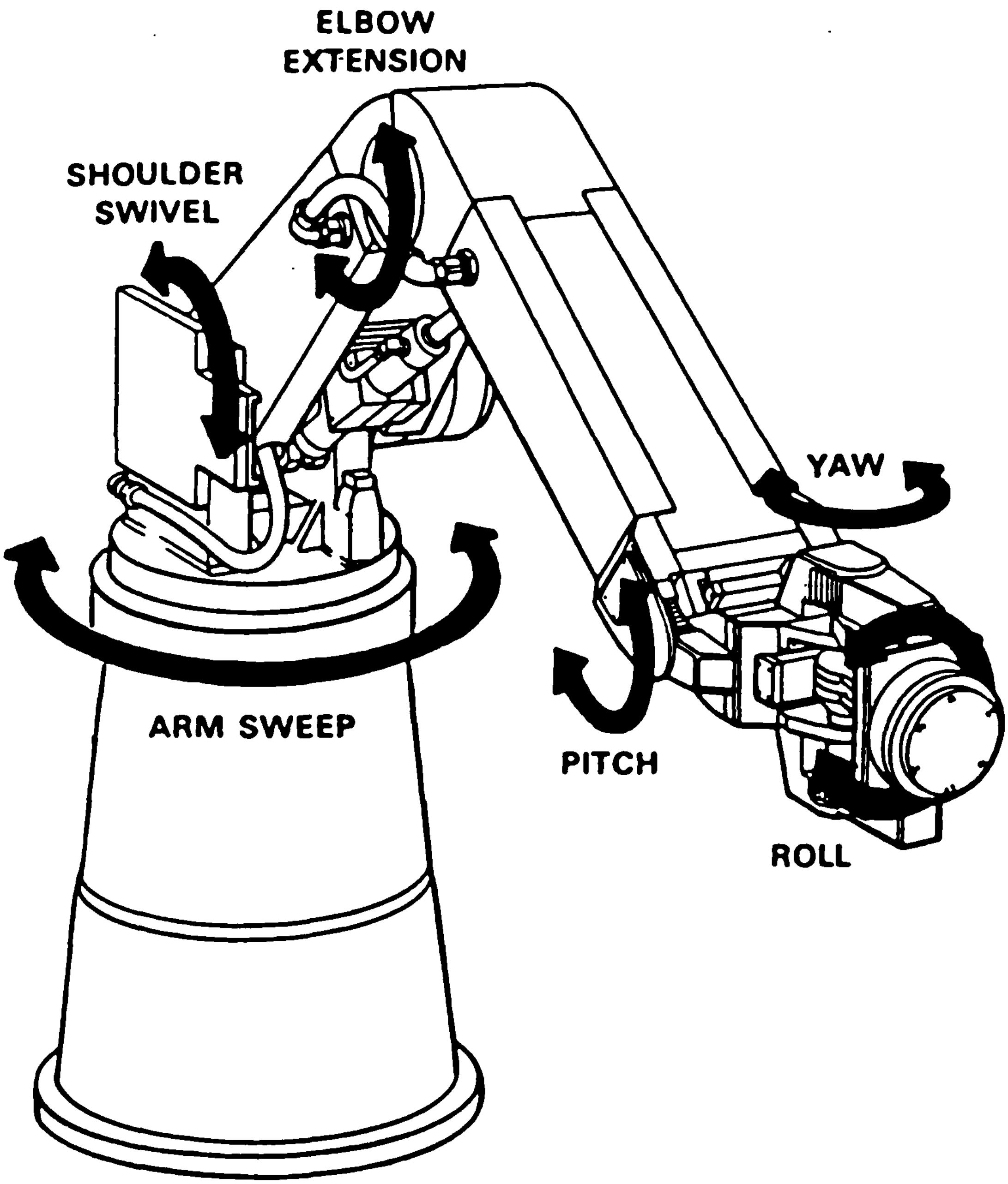



Figure IV-1. Illustration of an Industrial Robot controller to close the valves and send out new signals. This process is repeated until the sequence of moves has been executed.

b. Servo-Controlled

Servo-controlled robots, on the other hand, provide maximum positioning capability because of their ability to position each axis anywhere within its limits of travel. These robots operate by recalling prerecorded positional data from memory, using these data to generate motion command signals for each arm axis. As the individual axes move, the feedback devices are continuously read to determine the amount of error between the actual and desired positions. When the feedback indicates that the axes are approaching their destinations, the axes are brought to a controlled stop. The route that the arm takes to reach its destination can be controlled in one of two ways: point-to-point or continuous path.

1) Point-to-Point Control

Point-to-point control is the simplest control method. Teaching is performed by moving the arm to its desired position for each robot axis. When recalling these stored points, each axis runs at its maximum rate until it reaches its final position. Therefore, some axes reach their final value before others. Because there is no coordination of motion between axes, the path and velocity of the arm is not easily predictable. For this reason, point-to-point control is used for applications in which only the final position is of interest.

2) Continuous Path Control

Continuous path control is used when the path of the arm is of primary importance to the application, such as is required for spray painting. Typically, robots are taught by physically grasping the unit and leading it through the desired path in the exact manner and speed in which it is to repeat the motion. However, other types of teaching are available.

4. Power Sources

Robots may be hydraulically, electrically, or pneumatically driven. Briefly, the advantages of hydraulically driven robots include mechanical simplicity, higher load capacity, and high speed. However, hydraulically driven robots generally offer lower repeatability than their electrically driven counterparts.

In most cases, electric robots are not as strong or as fast as hydraulic robots, but they generally show better accuracy. Electrically driven robots save floor space and decrease noise levels since no hydraulic power unit is necessary to their operation.

Some non-servo-controlled robots may be pneumatic. For applications involving simple programs, such robots are often the best solution; however, their movement limitations make pneumatically driven robots extremely limited in their range of application.

## C. COMMAND OF ROBOTS

The greatest asset that robots possess is their versatility. Robots can be applied to a wide variety of jobs with a minimal amount of change in their hardware configuration. However, this flexibility cannot be properly exploited if the robot cannot be easily programmed. There are different methods of programming robots, and their applicability to this project will be discussed below.

### 1. Guiding

The earliest and most widespread method of programming robots involves manually moving the robot to all desired positions and recording the internal joint coordinates corresponding to these positions. In addition, operations such as turning the spray gun on or off are specified at some of these positions. This method of robot programming is sometimes known as guiding or teaching by showing. This method is simple to perform. Because it does not require the use of a peripheral computer, it was used for many years before it became cost effective to incorporate computers into industrial robots. Because the paint spraying application does not require the robot to make any conditional branches within its program, the guiding method of programming is ideally suited to it. For this reason, most, if not all, robotic painters can be programmed by using the guiding methodology. The only liability to the use of guiding is that each robot must be individually programmed. Typically, one master program cannot be produced centrally and distributed to participating locations.

2. Pseudo-Programming

A second method which is used to program painting robots uses the guiding method to program specific points of interest. It then uses a computer to control the robot's motion between these points. The computer can control the path, velocity, acceleration, etc. of the robot. This method is useful; however, it still requires an operator to manually move the robot to teach it.

3. Computer Programs

Another method of programming industrial robots uses only a computer program. A computer can instruct the robot to move to various locations and perform its functions. The resultant motion is identical to that of the guiding method. Since robotics is a relatively young industry, there are as yet no standard programming languages used throughout the industry. Two languages that have been developed for machining applications are APT and Compact, but these have not received broad acceptance. When a computer is in use, the robot must use a defined location as a reference point and move with respect to that point. This type of programming would be ideal for this application; however, it is not presently available for painting robots.

Sensors are frequently used in the programs which robots use; however, they are seldom used in the paint spraying application. One type of sensor which might be used in this application would determine the exact location of the object to be painted. This point could then be used as a reference point to begin the application of paint. Additional reference points could be used to record the location of other parts on the vehicle to avoid possible collisions with those parts.

Another problem which must be considered in certain applications is the possible use of a conveyor to transport the items being painted. When conveyors are used, the object is continuously moving. This changes the necessary movements of the robot. Once this is considered, the robot can be programmed to move horizontally with the object.

## D. THE PAINTING APPLICATION

The robot configurations described above are suited for a wide variety of applications; however, some configurations are not suited for specific applications. The nature of spray painting requires robots to possess some specific attributes. These attributes will be discussed in detail below.

### 1. Power

A major constraint on the types of robots that can be used is the paint itself. Most types of paints used in industrial applications are extremely flammable in their liquid state. Additionally, the use of a spray gun to apply the paint creates a mist which is even more highly ignitable. This safety hazard rules out the use of an electrical power source because possible sparks could cause a fire or an explosion. Also, pneumatic power sources are generally suited for only simple applications. Consequently, spray painting robots are almost exclusively hydraulically powered. Hydraulically powered robots are capable of performing at high speeds, and their repeatability of approximately ± 1 to 2 mm is sufficient for this application.

### 2. Programming

The method by which the robot can be programmed is also critical. For this application, the robot should be programmable using a high level computer language. This would reduce the amount of labor required for the programming of the robot since the program would be written once and copied for use in each painting facility. The use of a program to direct the robot would also reduce the possible error in painting which could arise if the robot were taught using the other common method, guiding.

A necessity for the robot controller is the ability to communicate with a peripheral computer. The camouflage pattern is generated on an external computer and the controller must be able to communicate with it. For this reason, most controllers are equipped with an RS232 port. Since the method of distribution for camouflage patterns will be a floppy disk, the controller must be able to read its instructions from the disk, or have some small dedicated microcomputer connected to the RS232 which can translate the floppy disk instructions.

Since the guiding technique requires a human operator to lead the robot through the pattern, a possibility of inaccuracies exists in the programming. On the other hand, it is not desirable to limit the robot's teaching to just a computer program. There may be situations where the guiding method is required. For this reason, the robot should be programmable by either method.

Additionally, the robot must have continuous path control. The painting application requires the manipulator to be controlled at all times; point-to-point control will not suffice.

3. <u>Size</u>

The robot's "work envelope" is another constraint on the system. Since vehicles of all sizes will be painted using this system, the robot's "work envelope" must be large enough for the robot to paint all vehicles. Another alternative would be to couple robots to paint different portions of the vehicles.

4. <u>Multiple Colors</u>

The three color pattern in use in this application adds another constraint. If the robot cannot be equipped with multiple nozzles, the nozzle would require changing while painting each vehicle. This would add significant time and cost to the painting process.

5. <u>General</u>

In addition to the factors stated above, there are several factors which affect the interface requirements associated with these robots. First, the optimal distance between the spray gun and the object to be painted (usually six to twelve inches) is a function of two things: the viscosity of the paint, and the pattern to be painted. Secondly, sensors which are more sophisticated than the ones that are normally used on painting robots may be required to detect irregularities in modified vehicles. Also, painting robots generally traverse the vehicle while painting one color per pass. This requires the robot to make three passes to paint each vehicle. Finally, it must be remembered that a significant effort will be required to interface a robot to this application if it is to be programmed only by a computer.

In summary, the robot(s) which should be used must be large enough for the job, hydraulically powered, multi-nozzled, and programmable by both a peripheral computer and the guiding technique.

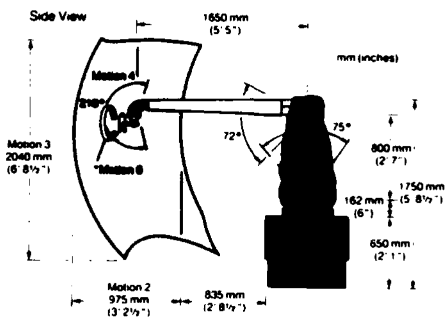
## E. COMMERCIALLY AVAILABLE SYSTEMS

The robotics industry is presently experiencing phenomenal growth worldwide, and this growth is expected to continue for some time. There are presently over 300 robotic systems manufacturers worldwide, and approximately twenty of these produce painting robots. Several manufacturers were contacted to determine what robots are presently available for this specific application. This section will discuss the attributes of some of these robots. This survey is intended to provide a list of several manufacturers that produce robots applicable to this project. It is not intended to be a complete list of robot manufacturers. Technical information and illustrations were obtained from the manufacturers' brochures.

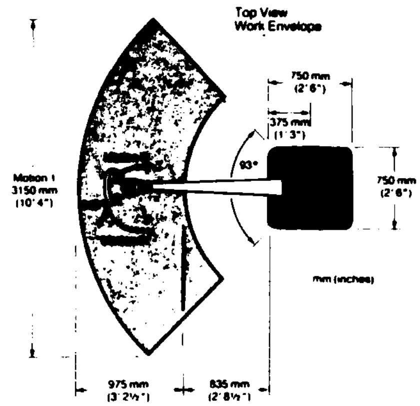
1. <u>DeVilbiss TR-3500/4500</u>

| | |
|---|---|
| MANUFACTURER: | The DeVilbiss Company<br>Toledo, Ohio 43692 |
| MODEL: | TR-3500, TR-4500 |
| PROGRAMMING: | Guiding or Pseudo Programming |
| POWER: | Hydraulic |
| KEY FEATURES: | RS-232 Port<br><br>Must be coupled to obtain sufficient work envelope.<br><br>Ideal for multiple colors. |

## DEVILBISS TR-3500/4500




*Optional 6th Motion 420° Spray Fan Rotation




*Optional 6th Motion 420° Spray Fan Rotation
3 Joint FLEXIARM Provides 176° Movement in Motion 4 and 5

### Physical Characteristics

| | Depth | Width | Height | Weight (Approx.) |
|---|---|---|---|---|
| Control Center | 482 mm (19") | 914 mm (36") | 1829 mm (72") | 148 kg (325 lbs.) |
| Manipulator | 750 mm (29.5") | 750 mm (29.5") | 650 mm (25.5") | 454 kg (1000 lbs.) |
| Hydraulic Power Unit | 610 mm (24") | 711 mm (28") | 762 mm (30") | 91 kg (200 lbs.) |

### TR-3500 FEATURES

Program Capacity:
- Standard 64 individual programs
- 128 minutes (depending on sampling rate)

Optional
- RAM 256 k Bytes memory available. See catalog Sheet I-8219.

Program Functions:
- Five (5) on/off (expanded to 15 optional)

Standard Inputs:
- Six (6) binary (for automatic program selction)
- One (1) program toggle (per automatic program selection)
- One (1) start signal
- One (1) conveyor synchronization
- One (1) oil level
- One (1) oil temperature sensor

### TR-4500 FEATURES

Program Capacity
- Standard
- 999 individual programs
- 45-60 minutes memory time (depending on sampling rate & function activity)
- Up to 99 modules per program

Optional: Error printer

Program Functions
- Five (5) on/off, standard
- Up to 20 on/off functions optional

Standard Inputs:
- Three (3) digit BCD (for remote program selection)
- One (1) program toggle (for remote program selection)
- One (1) start signal
- One (1) direction sensing conveyor synchronization
- One (1) oil level fault
- One (1) oil temperature fault

Optional Inputs
- Oil level warning
- Oil temperature warning
- Return filter backpressure warning fault

Optional Outputs
- 12 status lines for remote monitoring

Figure IV-2. DeVilbiss TR-3500/4500

2. Thermwood Paintmiser

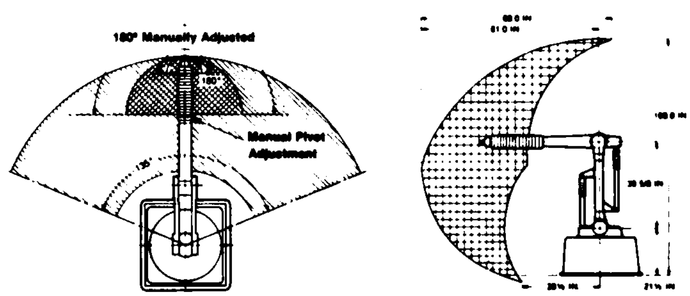
| | |
|---|---|
| MANUFACTURER: | The Thermwood Corporation<br>Richardson, Texas 75081<br>1-800-527-4417 |
| MODEL: | Paintmiser |
| PROGRAMMING: | Guiding only |
| POWER: | Hydraulic |
| KEY FEATURES: | Good repeatability |
| | Large Memory |
| | Cannot be programmed by a peripheral computer. |
| | RS-232 Port |

# PAINTMISER Specifications

| | |
|---|---|
| Number of Axes | 6 (additional axes optional) |
| Configuration | Jointed Arm |
| Mounting Position | Floor |
| Coordinate System | Jointed Arm |
| Drive System | Intrinsically safe hydraulic/mechanical |
| Load Capacity | 18 pounds - full speed |
| Horizontal Stroke/Speed | 48 inches - 30 inches/second |
| Vertical Stroke/Speed | 84 inches - 30 inches/second |
| Rotary Stroke/Speed | 135° - 60°/second |
| Wrist Roll/Travel Speed | 270° 150°/second |
| Wrist Movement | 180° cone |
| Repeatability | ±.125 inches |
| Control System | Closed loop servo analog feedback<br>Microcomputer based solid state |
| Memory Type | Semiconductor/disc memory option |
| Memory Size | 4.2 minutes continuous path (Expandable)<br>point to point dependent on program |

| | |
|---|---|
| Number of Programs | 9 (Expandable) |
| Random Selection | Yes |
| Operating Modes | Continuous path/Point to point |
| Programming | Lead through teach |
| Editing | Yes |
| Adjustable Playback Speed | Yes |
| Outputs | 8 |
| Inputs | Remote Program Select Lines |
| Interface Hardware | Terminal strip for external wiring<br>RS232C selectable baud rate<br>asynchronous RS232C synchronous |
| Ambient Conditions | 40° to 120°F 5% to 95%<br>humidity non-condensing |
| Power Required | Maximum 10 kw |

Thermwood Corporation reserves the right to make changes in the specifications at any time




Rotational Sweep

Vertical Working Envelope

Figure IV-3. Thermwood Paintmiser

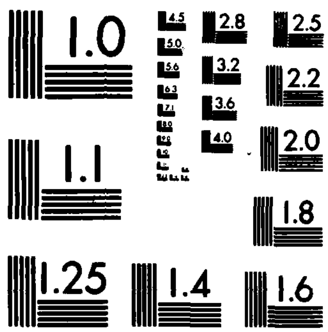
AD-A158 903 AUTOMATED CAMOUFLAGE PATTERN GENERATION TECHNOLOGY SURVEY(U) BDM CORP MCLEAN VA G N ROMSTEDT ET AL. 07 AUG 85 BDM/W-85-0135-TR DAAK70-83-D-0019 2/2

UNCLASSIFIED F/G 15/7 NL

END

FILMED

DTIC



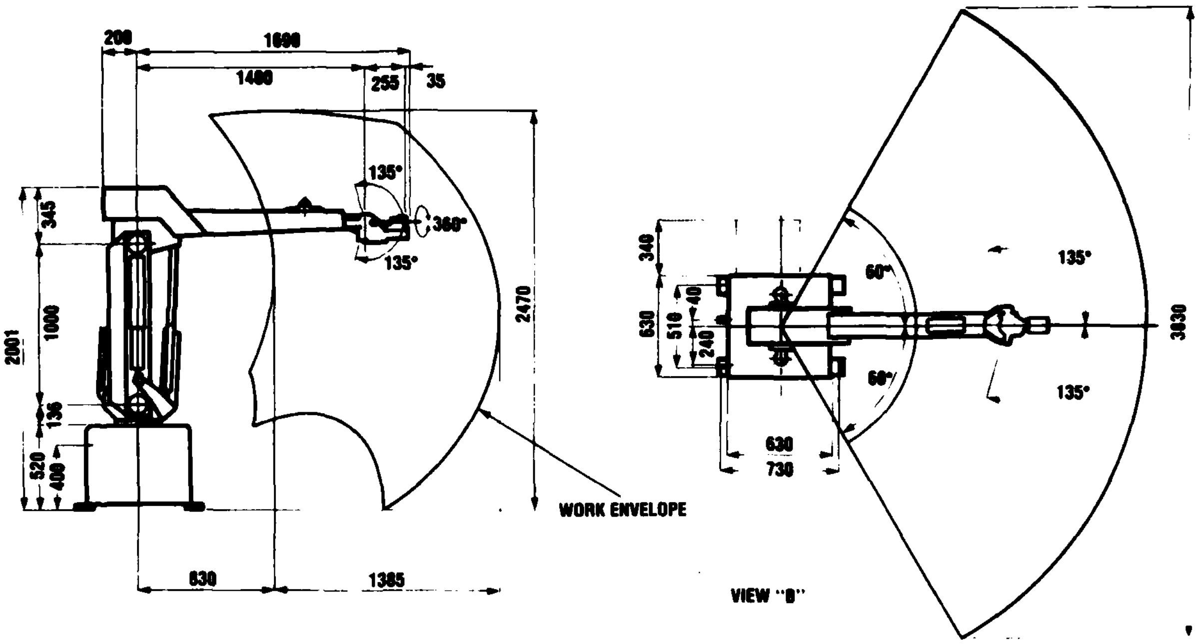

MICROCOPY RESOLUTION TEST CHART

DARDS-1963-A

3. Cybotech TP-15

| | |
|---|---|
| MANUFACTURER: | Cybotech Corporation<br>Indianapolis, Indiana 46208<br>(317) 298-5890 |
| MODEL: | TP-15 |
| PROGRAMMING: | Guiding or Pseudo Programming |
| POWER: | Hydraulic |
| KEY FEATURES: | TP-15 has good repeatability<br><br>H-530 has a larger work envelope. |

All dimensions are stated metric




**Drive System**
Direct hydraulic cylinder, Hydraulic motor

**Operating Environment:**
with fan cooling: 5° – 38° C
(41°-100° F)

**Control System**
The TP15 Control System is the most flexible and advanced available for any painting robot system in the world, with both continuous path and point-to-point execution, and direct or remote teaching. The standard RAM/floppy disc combination stores data for 34,000 points of working memory. The Controller's user-friendly display and simplified hand-held editing/modification control offer unequalled ease of programming and editing.

**Weight**
545 kg (1200 lbs)

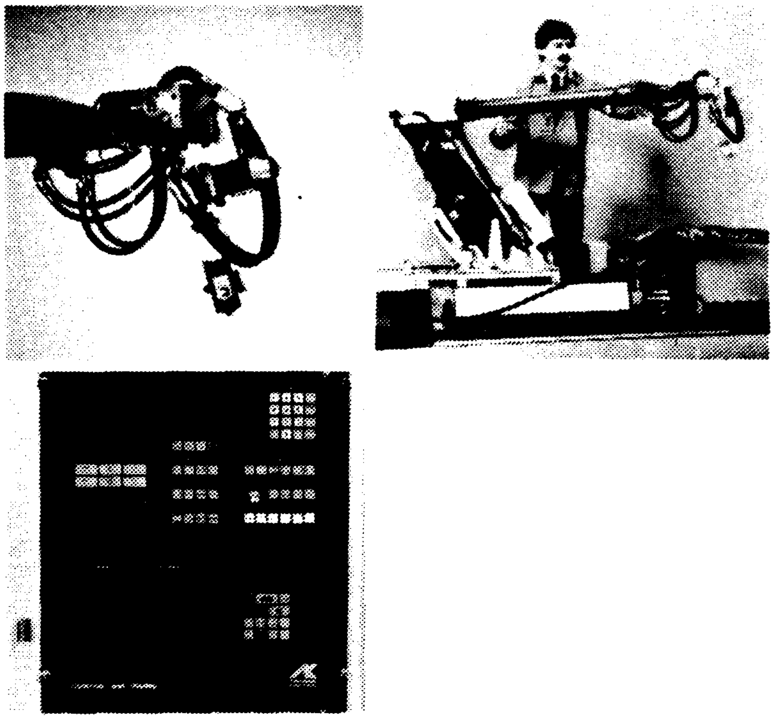
**Power Requirements**
460 VAC, 10k VA 3 Phase

Applications engineers are available to assist you in selecting the industrial robot system that best suits your job requirements.

Figure IV-4. Cybotech TP-15

4. AKR 3000

| | |
|---|---|
| MANUFACTURER: | AKR Incorporated<br>Subsidiary of AKR Robotique, France<br>Livonia, Michigan 48150<br>(313) 261-8700 |
| MODEL: | AKR 3000 |
| PROGRAMMING: | Guiding or Pseudo Programming |
| POWER: | Hydraulic |
| KEY FEATURES: | Excellent speed<br><br>Excellent repeatability<br><br>Multiple paint gun capability<br><br>Coupling system already exists.<br><br>RS-232 Port |



## FEATURES

- 5 to 7 servo-controlled axes
- End of arm payload 15 kg max (33 lbs)
- Max wrist torque 46 N M (336 ft lb) allowing 1-3 spray guns typically
- Speed above 2 m/s (65 ft/s)
- Static repeatability less then ± 2 mm (end of arm)
- Work envelope for the normal length arm (shown above)
- Anti-collision safety system on the horizontal arm
- Dust proof control cabinet (IP55)
- 16/32 Inputs
- 16/32 Outputs
- RS232 Port

**Programming**
The AKR 3000 robot provides the broadest choice of programming methods

1 Direct lead-through teaching the robot is guided directly by the operator

2 Syntaxor lead-through teaching the AKR teach arm, which is extremely light and easy to manipulate, is substituted for the robot during programming. The use of the Syntaxor permits the operator to teach the system with a natural technique unhindered by the weight and inertia

3 Telemanipulation teaching the robot follows the movements of the teach arm being manipulated by the operator and the movement of the robot is recorded. This method allows programming while leaving the operator out of a dangerous or inaccessible area, as sand blasting, for example

4 Point-to-point lead-through teaching the operator moves and orients the robot arm (or the teach arm) to teach certain key points, while leaving the definition of the final trajectory between these points up to the system controller

5 Advanced Analytic Programming ($A^2P$) the operator describes a trajectory through definition of points, speeds, accelerations, and device controls in a written program that can be edited with all the capabilities of a high level language

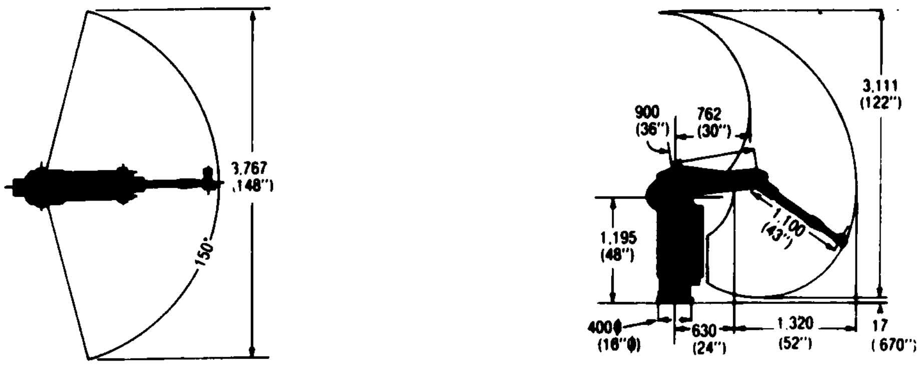
6 CAD programming the robot programs are generated automatically based on data received from an external CAD system

Figure IV-5. AKR 3000

## 5. General Electric S6

| | |
|---|---|
| MANUFACTURER: | General Electric<br>Bridgeport, Connecticut 06602<br>(203) 382-2876 |
| MODEL: | S6 |
| PROGRAMMING: | Guiding only |
| POWER: | Hydraulic |
| KEY FEATURES: | Excellent speed<br><br>Must be coupled |

## VERTICAL TYPE S6




| | | |
|---|---|---|
| Speed | | Max. 1,750 mm/sec (69 in/sec)<br>Painting 1,000 mm/sec (39"/sec) |
| Weight | | 500 kg (1,100 lbs.) |

**• CONTROL SYSTEM**

| | |
|---|---|
| Teaching method | Remote teaching through the teaching box |
| Control method | Continuous Path by Point-to-Point linear interpolation |
| Memory | Core memory |
| Number of programs | 15 |
| Modes of operation | Converting operation of multi-joint coordinates into cartesian coordinates<br>Wrist correction operation<br>Linear-speed control operation<br>Conveyor synchronizing operation |
| Fault detection function | Sensor failure<br>CPU failure<br>Hydraulic power source failure<br>Painting machine failure<br>Conveyor failure |
| External input/output points | 7 inputs, 7 outputs |
| Electric power source | AC 115V ±10%, 50/60 Hz, 1 kVA |
| Weight | 150 kg (330 lbs.) |

Options 1. Memory storage system (cassette deck)
2. Additional memory 512 steps (maximum 3)

Figure IV-6. GE S6

6. Graco Robotics OM-5000

| | |
|---|---|
| MANUFACTURER: | Graco Robotics, Incorporated<br>Livonia, Michigan 48150 |
| MODEL: | OM-5000 |
| PROGRAMMING: | Guiding or Pseudo Programming |
| POWER: | Hydraulic |
| KEY FEATURES: | Can be synchronized with a conveyor system.<br><br>Large work envelope<br><br>RS-232 Port |

## FEATURES

- Floating start points characteristic of each program
- Automatic homing to start point
- Continuous path/point to point
- Editing on/off functions
- Editing path points
- Extensive self-diagnostic with self-test
- CRT display
- Optional remote CRT monitor
- Random program access
- Conveyor synchronization
- Conveyor speed read-out
- Register memory for automatic part identification
- Programmed color change
- Speed control up to 200%
- Programming linking/merging
- Servo position displayed on CRT (LCD display on 816-400 RCP)
- Servo optimized in real-time
- Output forcing for testing and troubleshooting
- Mirror Imaging
- Programmable, retentitive/non-retentitive, outputs
- Copy/editing programs

**Physical Characteristics**

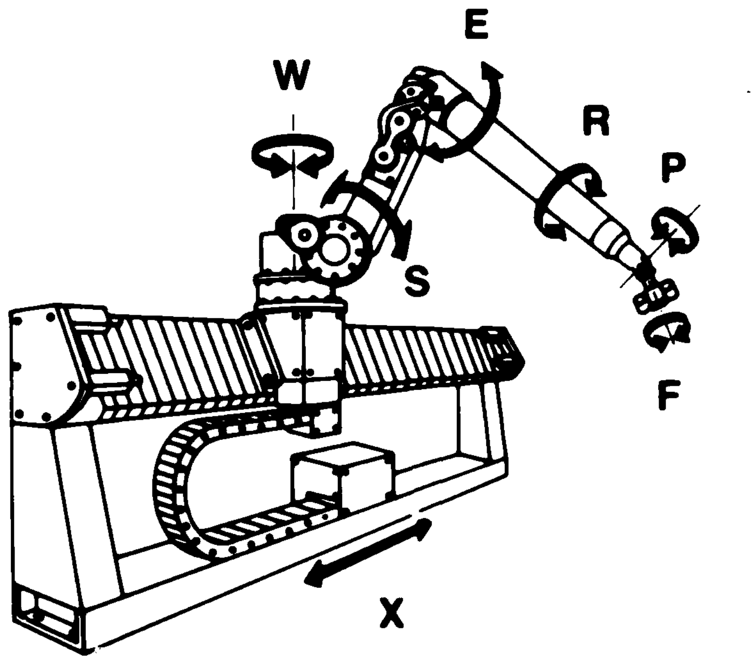
| Space Requirements Description | Depth | Width | Height | Weight (Approx.) |
|---|---|---|---|---|
| Manipulator Base, 830-200 | 42" | 30" | 26" | 1259 lbs |
| Manipulator Base, 816-200 | 42" | 30" | 32" | 1000 lbs |
| Computer Console, 830-100 | 24" | 26" | 66" | 378 lbs |
| Computer Console, 816-000 (less 20" legs) | 17" | 25" | 54" | 250 lbs |
| Accessory Cabinet | 17" | 25" | 20" | 50 lbs |
| I/O Panel | 12" | 37" | 66" | 800 lbs |
| R.C.P., 830-404 | 3 | 8" | 18" | 3 lbs |
| R.C.P., 816-400 | 2" | 8" | 18" | 2 lbs |
| Hydraulic Supply | 30" | 36" | 30" | 250 lbs |

| Environmental Requirements Description | Temperature | |
|---|---|---|
| Manipulators, Computer Panels, I/O Panel, R.C.P.'s | 0-49° C | 32-120° F |
| Hydraulic System (max. oil temp.) | 65° C | 149° F |

Figure IV-7. Graco Robotics OM-5000

7. GMF NC Painter

| | |
|---|---|
| MANUFACTURER: | GMF Robotics<br>Troy, Michigan 48098<br>(313) 641-4242 |
| MODEL: | NC Painter |
| PROGRAMMING: | Guiding or Pseudo Programming |
| POWER: | Hydraulic |
| KEY FEATURES: | Poor repeatability<br><br>Large work envelope |




FEATURES:

- ± 6.4mm REPEATABILITY
- AXES TRAVELS

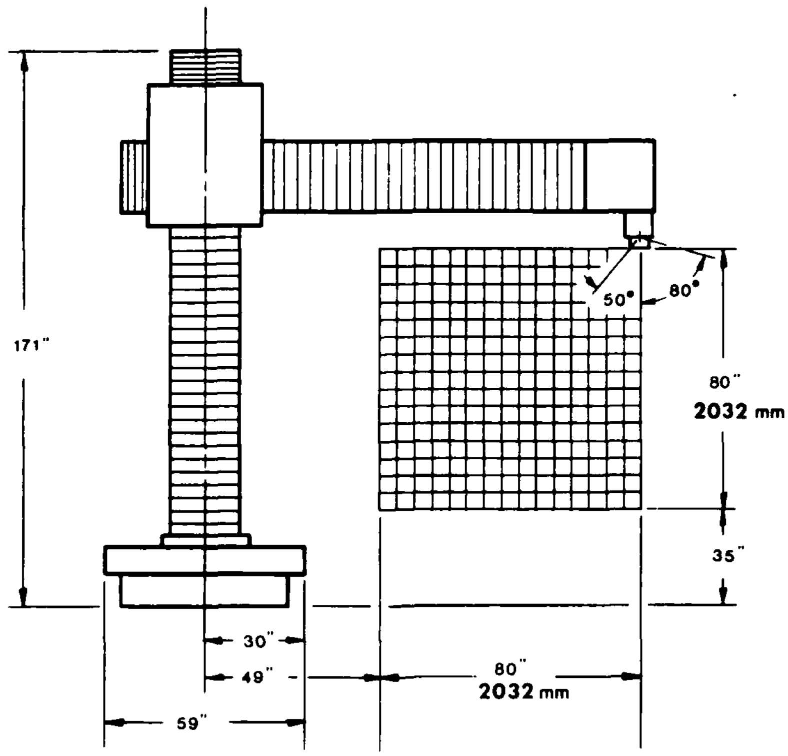
X = 16 ft. (4.9m)
W = 160° (waist rotation)
S = 128° (shoulder rotation)
E = 140° (elbow rotation)
R = 1080° (wrist rotation)
P = 180° (wrist yaw)
F = 1080° (fan rotation)

- HYDRAULIC SERVE DRIVE

Figure IV-8. GMF NC Painter

8. Advanced Robotics Cyro 2000

| | |
|---|---|
| MANUFACTURER: | Advanced Robotics Corporation<br>Newark Ohio Industrial Park<br>Building 8<br>Route 79<br>Hebron, Ohio 43025<br>(614) 929-1065 |
| MODEL: | Cyro 2000 Advanced Generation |
| PROGRAMMING: | Computer Programming |
| POWER: | Electric |
| KEY FEATURES: | Excellent Repeatability<br>Easily Programmable<br>Large Work Envelope |




**Cyro® 2000 Robot Specifications**

**Motions**
5 axes—powered by electric Servo Motors
Travel:
X axis—80 in. (2032 mm.) expandable
Y axis—80 in. (2032 mm.)
Z axis—80 in. (2032 mm.)
A axis—130°
C axis—720°

**Repeatability**
Repeatable to ± 0.016 in. (± 0.40 mm.)

**Maximum Speed**
X, Y axis—300 in./min. (7620 mm./min.)
Z axis—150 in./min. (3810 mm./min.)
A, C axis—90°/sec.

**Floor Space**
Robot Base: 54 sq. ft. (4.86 sq. m.)
Control Cabinet: 13 sq. ft. (1.17 sq. m.)

**Power Requirements**
480 Volts, 3 phase, 60 Hz, 15 KVA

**Environments**
40°F to 120°F (5°C to 50°C)

**Weight**
Robot —12000 lbs. (5454 kg.)
Control— 1800 lbs. (818 kg.)

**Programming Methods**
Teaching with a pendant
Numerical programming via terminal

**Programmable Features**
Linear interpolation
Circular interpolation (3 types)
Acceleration/deceleration
Analog function (2 outputs)
Dwell
Inch/Metric units
Absolute/Incremental dimension
Program shift
Programmable tooling point
Oscillation
Program editor
8 I/0 expandable to 32 I/0

**Memory Capacity**
64K Bytes
Permanent program storage on tape cassette

**Options**
Coordinated Positioning Robots
X axis expansion
Welding process packages including computer programmable weld parameters, voltage, wire speed, dwell, burnback
ArcScan™ welding seam tracker

Specifications subject to change without notice

Figure IV-9. Advanced Robotics Cyro 2000

## F. SUMMARY

The phenomenal growth of the robotic industry is due in part to the flexibility of the robot. In addition to painting, robots are used in assembly, inspection, forging, welding, material handling, and a wide variety of other applications. Because of this growth, the technology used in this industry is also improving rapidly. As this growth continues, new models of robots will be constantly introduced to the marketplace. For this reason, this survey should be reviewed and updated in the future to determine the existence and applicability of future models.

# BIBLIOGRAPHY

1. Ottinger, Lester V., "Robotic Applications - The Next Wave," Robotics World, Vol. 1, No. 3, March 1983, pp 18-21.

2. Perez, Tomas Lozano, "Robot Programming," Proceedings of the IEEE, Vol. 71, No. 7, July 1983, pp 821-841.

3. Susnjara, Ken, "Robotic Painting Solutions," Robotics World, Vol. 2, No. 2, February 1984, pp 20-22.

4. Tarvin, Ronald L., Merton D. Corwin, and Wayne E. Mechlin, "Industrial Robotics," Handbook of Industrial Engineering, New York: John Wiley & Sons, Inc., 1982, pp 7.6.1 - 7.6.7.

# CHAPTER V
# ASSESSMENTS AND RECOMMENDATIONS

## A. COMPUTER SYSTEMS

The survey of computer systems considered those computers already owned by BRDC as well as several commercially available CAD systems. The survey showed that the one capable in house system was much more expensive to operate and less practical to manage than any of the dedicated commercial systems. As a result, the most capable and least expensive commercial CAD system, the Autotrol AGW-III, was recommended for the production camouflage pattern generation system.

The Autotrol is an integrated package of extensive computer aided design and drafting software, unique operator interface hardware, and a state-of-the-art computer workstation. It is based on a ring network architecture to share peripherals and data among the workstations and other computers. The product is a complete system, and yet it offers great flexibility for incorporating the inevitable improvements in computer technology because of its modular structure. Relatively low cost components can be attached and upgraded as necessary to keep pace with new camouflage requirements.

## B. OPTICAL DIGITIZATION

Optical digitization is a subject of much concern to the camouflage pattern generation program since the bulk of the labor in creating patterns by computer will be in the transferring of data from existing mechanical drawings into the computer. It is hoped that an optical process will be able to automatically capture the vehicle dimensional information from a sequence of plane view mechanical drawings and thus be able to satisfy the system need for two and three dimensional system representations.

Optical digitization technology is rapidly advancing. Over the course of this survey, many manufacturers have introduced new and substantively upgraded capabilities. Image capture systems with up to 2000 x 2000 point resolution are being marketed that can function in approximately 15 seconds. They include built-in image-enhancement software that operates interactively to facilitate image-cleanup and correction. Charged coupled device (CCD) arrays have been the primary sensing technology, but recently laser scanning has been exploited for the digital sampling of the image. Laser devices promise even higher resolution and faster scanning as they become more developed.

Unfortunately, the critical piece to the optical digitization requirement for camouflage is missing. While high resolution images can be obtained of the original mechanical drawings, they are internally represented in the computer as a sequence of on and off points. There is no computer understanding of the vector or dimensional information (e.g., lines, circles, size, scale, hidden lines) that is the real content of the mechanical drawing. There exist only crude algorithms that attempt to recognize the drawing information, categorize it, and build a drafting file of data for the image. These algorithms are slow and quite prone to misinterpretation and so require a lot of operator supervision and correction. The problem is a fundamental one of pattern recognition to which humans are well suited and computers currently are not. Great strides are being made in developing computers and algorithms to address pattern recognition problems (e.g., artificial intelligence, supercomputers, optical character readers). Major CAD system manufacturers are working feverishly on developing this capability so that their customers can enter their existing drawings into their CAD system's data base for revision. But as of this writing no such system exists.

The automatic extension of several two dimensional images into a three dimensional shape is similarly difficult. Work was begun in the 1960's on this problem as a means to create 3D computer aided design models. A projection and intersection technique was employed which was able to handle simple orthogonal shapes. Curved surfaces and doubly sloped surfaces, however, were not handled well. Two dimensional drawings don't adequately represent them in an unambiguous fashion. Furthermore, the many surface details and hidden lines present in a 3D drawing create an enormous decision burden for the automatic process. The result is often a confused 3D model with the computer making rational but incorrect assumptions about the true shape. Since then, the problem has not been resolved since the projection capabilities of computer aided drafting systems have been enhanced. The operator's intervention into and control over this process is deemed easier than trying to clean up the mess created by automatic systems.

## C. ROBOTICS INTERFACE

A wide variety of robotic painters are available in the commercial marketplace. All of these robots are programmable to repeat a specific sequence of axis movements and spray operations. Typically, these robots are taught by manually moving the arm through its course and indicating where the spray is to start and stop. In the teaching mode, the robot records these movements (on diskette or tape) in each of its rotation axes/joints and spray heads. Then, these instructions are played back and executed as required. Programs may be changed by substituting a different instruction tape or diskette.

Some painting robots are equipped with RS232 interfaces to connect the external computers and device controllers. These links are used predominately to synchronize several robots within an assembly line or to share a single device controller among several robots. Unfortunately, there is currently no standard way of communicating movement instructions to these robots. Each uses its own peculiar "language".

The interface of the Automated Camouflage Pattern Generation System to painting robots will have to be specifically designed for the robot itself. If the robot is programmed by teaching then each arm/joint position needs to be computed along the spray path. This may also be the case for some of the RS232 equipped robots if the device controller does not perform that computation automatically. The computation must be responsive to the arm lengths, axes of rotation, and degrees of freedom of the robot. In addition the generated spray path needs to account for the number of spray heads on the robot arm, the spray pressure, and the spray dispersion. Finally, for those cases in which a synchronized set of robot sprayers are employed, the spray paths for each painter must be segregated to prevent collisions. All of these specific instructions must then be formatted and provided in an electronically and mechanically compatible medium to the robot.

END

FILMED

10-85

DTIC