

# NASA Technical Memorandum

NASA-TM-86457 19840025005

NASA TM 86457

## COMPUTER VISION FOR REAL-TIME ORBITAL OPERATIONS

Center Directors Discretionary Fund Final Report  
(Project No. 82-27)

By Frank L. Vinz, Linda L. Brewster, and  
L. Dale Thomas

Information and Electronic Systems Laboratory

August 1984

**LIBRARY COPY**

**JUL 29 1985**

LANGLEY RESEARCH CENTER  
LIBRARY, NASA  
HAMPTON, VIRGINIA

# NASA

National Aeronautics and  
Space Administration

George C. Marshall Space Flight Center



NF00626

1. REPORT NO. NASA TM-86457		2. GOVERNMENT ACCESSION NO.		3. RECIPIENT'S CATALOG NO.	
4. TITLE AND SUBTITLE Computer Vision for Real-Time Orbital Operations — Center Directors Discretionary Fund Final Report (Project No. 82-27)				5. REPORT DATE August 1984	
				6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S) F. L. Vinz, L. L. Brewster, and L. D. Thomas				8. PERFORMING ORGANIZATION REPORT #	
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812				10. WORK UNIT NO.	
				11. CONTRACT OR GRANT NO.	
				13. TYPE OF REPORT & PERIOD COVERED Technical Memorandum	
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546				14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES Prepared by Information and Electronic Systems Laboratory, Science and Engineering.					
16. ABSTRACT  This report describes machine vision research at Marshall Space Flight Center which has potential benefit for the NASA Space Station program and its associated Orbital Maneuvering Vehicle (OMV). Initial operation of OMV for orbital assembly, docking, and servicing will be manually controlled from the ground by means of an on-board TV camera. These orbital operations may later be accomplished autonomously by machine vision techniques which use the TV camera as a sensing device. Classical machine vision techniques are described in this report. An alternate method was developed and is described which employs a syntactic pattern recognition scheme. It has the potential for substantial reduction of computing and data storage requirements in comparison to the Two-Dimensional Fast Fourier Transform (2D FFT) image analysis. The method embodies powerful heuristic pattern recognition capability by identifying image shapes such as elongation, symmetry, number of appendages, and the relative length of appendages.					
17. KEY WORDS Machine vision techniques			18. DISTRIBUTION STATEMENT Unclassified-Unlimited		
19. SECURITY CLASSIF. (of this report) Unclassified		20. SECURITY CLASSIF. (of this page) Unclassified		21. NO. OF PAGES 40	22. PRICE NTIS

1184-3.3.076 #

## TABLE OF CONTENTS

	Page
INTRODUCTION .....	1
MACHINE VISION REQUIREMENTS .....	1
SCENE ANALYSIS BY FAST FOURIER TRANSFORMS .....	6
TWO DIMENSIONAL FAST FOURIER TRANSFORM IMPLEMENTATION .....	7
Camera Interface .....	7
Two Dimensional Fast Fourier Transform .....	8
Image Recognition .....	9
REAL TIME VIDEO INTERFACE .....	9
SCENE ANALYSIS BY SYNTACTIC RECOGNITION.....	10
APPLICATION OF "AI BOSS" TO REPRESENTATIVE TARGET VEHICLE.....	22
SCENARIO OF AUTONOMOUS DOCKING MISSION.....	29
CONCLUSION.....	31
REFERENCES.....	32
APPENDIX — DESIGN DESCRIPTION OF REAL TIME VIDEO INTERFACE.....	33

## LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Orbital Maneuvering Vehicle .....	2
2.	Candidate OMV missions .....	4
3.	Orbital docking simulation facility .....	5
4.	Development/evaluation of machine vision for real time automated assembly .....	7
5.	Video camera to array processor real time interface .....	10
6.	Tree graph of "T" image .....	11
7.	Tree graph of "T" image rotated 90 deg CW .....	14
8.	Tree graph of "T" image rotated 180 deg CW .....	15
9.	Tree graph of "T" image rotated 270 deg CW .....	15
10.	Tree graph of "T" image rotated 315 deg CW .....	16
11.	Tree graph of "T" image with two pixels shifted .....	16
12.	Performance measurement for chaser roll search of "T" pattern .....	18
13.	Tree graph of irregular pattern .....	19
14.	Tree graph of irregular pattern rotated 90 deg CW .....	19
15.	Tree graph of irregular pattern rotated 180 deg CW .....	20
16.	Tree graph of irregular pattern rotated 270 deg CW .....	20
17.	Tree graph of irregular pattern rotated 315 deg CW .....	21
18.	Performance measurement for chaser roll search of irregular pattern .....	22
19.	Typical reference views of target vehicle .....	23
20.	Space Telescope image .....	24
21.	Binary image of Space Telescope .....	24
22.	32 x 32 pixel representation of the Space Telescope .....	25
23.	Sequential box scan .....	25
24.	Tree graph of Space Telescope .....	26

## LIST OF ILLUSTRATIONS (Concluded)

Figure	Title	Page
25.	Tree graph end point selection .....	27
26.	Tree graph end points .....	28
27.	Implementation of "AI BOSS" for autonomous docking .....	30

## LIST OF TABLES

Table	Title	Page
1.	Comparison of Box Scan and Line Scan Methods.....	17
2.	Performance Measurements of "T" Patterns.....	17
3.	Performance Measurements of Irregular Patterns .....	21

## LIST OF ACRONYMS

AI BOSS	Analysis of Images by Box Scan and Syntax
A/D	Analog to Digital
DMA	Direct Memory Access
GPIOP	General Purpose Input Output Port
I/O	Input/Output
MSFC	Marshall Space Flight Center
OMV	Orbital Maneuvering Vehicle
2D FFT	Two Dimensional Fast Fourier Transform

## TECHNICAL MEMORANDUM 86457

### COMPUTER VISION FOR REAL-TIME ORBITAL OPERATIONS

#### INTRODUCTION

The development of machine vision techniques for industrial automation and other robotic systems has been a long standing goal which has only been recently met with restricted success in manufacturing applications. Machine vision is used to denote the ability of a device to process visual information so that scene interpretation can be made and decisions result which allow the accomplishment of non-trivial tasks. Video systems are a natural choice for the sensing element of a machine vision system because they are commonly available, can be made to function in diverse environments and are relatively cheap. However, a TV camera can acquire such a vast amount of visual data in such a short period of time that processing systems may not have the ability to store all visual data nor could they perform calculations fast enough for real time scene analysis and subsequent decision making. The solution of these problems is fundamental to the development of machine vision.

The advancement of microelectronics, computer processing speed, and memory capacity coupled with enormous decreases in equipment prices places science and engineering at the threshold of developing substantial machine vision capability. At this time computing systems are marketed for less than \$100,000 which can store 2 to 5 million bits of data and process this data at the rate of 10 million floating point operations per second. As impressive as these computing hardware advances are, there is the potential for an equally dramatic advance in machine vision capability through the application and development of innovative artificial intelligence techniques.

This report describes a particular application of machine vision which has potential for NASA Space Station program benefit. A classical 2D FFT signal processing technique for possible use in a machine vision system is summarized. An alternate method is also described which was developed during this two-year research. It employs a syntactic pattern recognition scheme. It has the potential for reducing data storage requirements by perhaps one order of magnitude and providing an even greater decrease in computation requirements.

This research was accomplished by personnel in the Data Management Branch of the Software and Data Management Division, Information and Electronic Systems Laboratory of Marshall Space Flight Center. Funding for this two-year project was provided by the Center Director's Discretionary Fund program.

#### MACHINE VISION REQUIREMENTS

A likely initial application of machine vision capability is for orbital docking, servicing, and assembly for advanced NASA Space Station operations. A teleoperator type vehicle is currently being designed to accomplish these tasks. It is designated the Orbital Maneuvering Vehicle (OMV) and is depicted in Figure 1. It would be carried to low Earth orbit by the Space Shuttle Orbiter and manually operated from

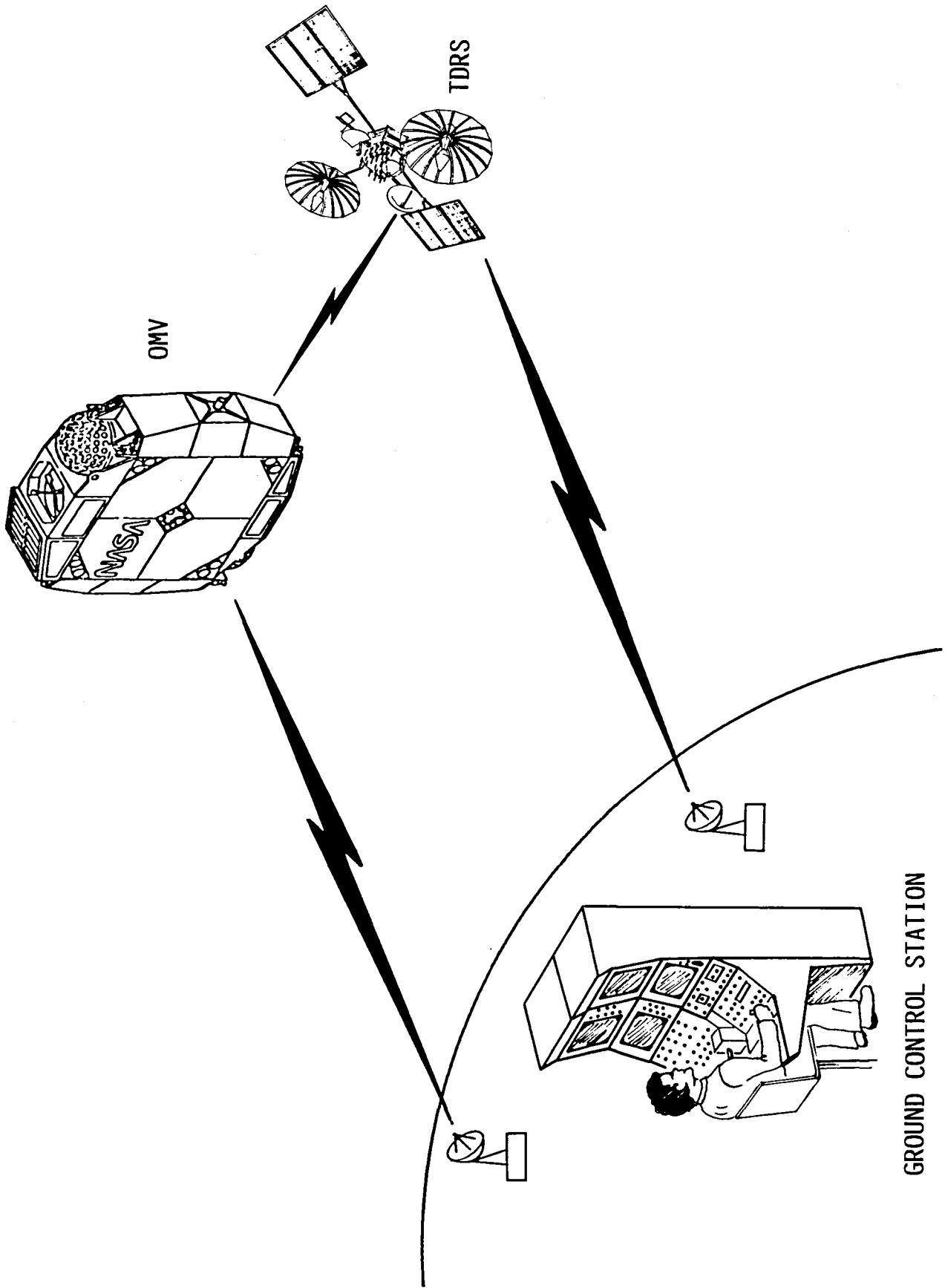


Figure 1. Orbital Maneuvering Vehicle.

GROUND CONTROL STATION



a remote ground station during critical station keeping, docking and servicing operations. Visual data for the ground operators will be provided by onboard TV cameras whose video signals are transmitted to the remote control station by means of several communication links. Candidate missions for the OMV include servicing and satellite viewing and debris capture (Fig. 2).

Remotely controlled docking maneuvers can be a difficult task for a human operator to perform for a number of reasons such as:

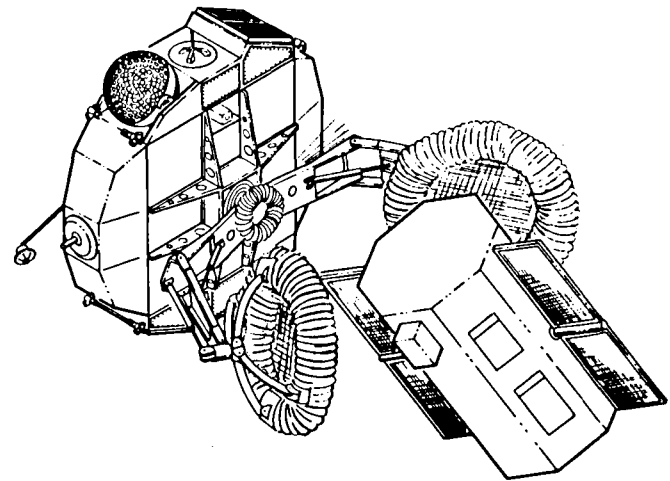
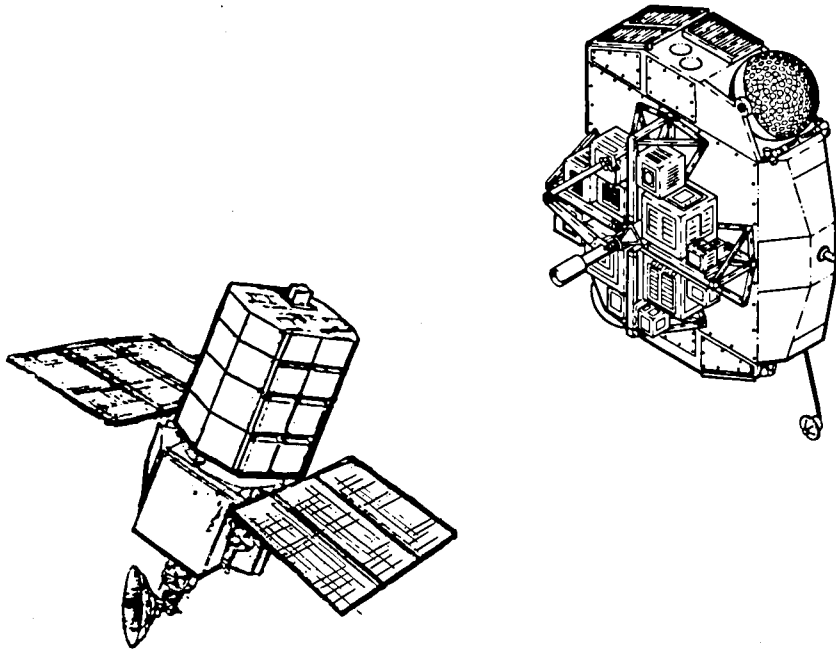
- 1) The OMV must be flown in six degrees-of-freedom (i.e., x, y, and z translations and roll, pitch, and yaw rotations). Simultaneous control of six variables can be a demanding task for a human operator.
- 2) The target vehicle may be disabled so that docking aids such as visual alignment devices, transponders or light patterns are not usable.
- 3) The target vehicle may have lost its attitude stabilization capability and could be spinning, coning, or tumbling. This is almost certain to be encountered on debris capture operations.
- 4) Anticipated time delays of up to 2 sec needed for round-trip transmission of video data and command signals can seriously complicate remote control during critical alignment operations just prior to docking mechanism engagement.

Because of these difficulties machine vision techniques were investigated in order to provide enhanced capability of the OMV for automated docking and servicing operations. The onboard TV camera to be provided with the initial OMV could be used as the sensor input to the machine vision system. Thus machine vision for automatic docking is a viable evolutionary growth possibility providing the additional onboard processing requirements are reasonable. A machine vision capability for the OMV offers the following autonomous operation advantages:

- 1) Independence from the operation of docking alignment aids such as transponders or light patterns.
- 2) Independence from communications links including TDRSS.
- 3) Communication time delays for vehicle control are eliminated.
- 4) Large costs associated with mission control operations and remote control operator training are either eliminated or substantially reduced.

Machine vision techniques which were developed by this MSFC research project will be evaluated by use of an orbital docking simulation facility as shown in Figure 3. This facility includes a TV camera system which represents the view from a chaser vehicle, such as the OMV, and either a scale model or full size model of the target vehicle. This simulation facility provides relative translation and rotation dynamics between both vehicles in order to represent typical orbital docking maneuvers. The analysis and investigation of candidate machine vision techniques on this orbital docking simulator will provide realistic visual and dynamic environments for this challenging development effort.

SATELLITE VIEWING/DOCKING/SERVICING



DEBRIS CAPTURE

Figure 2. Candidate OMV missions.

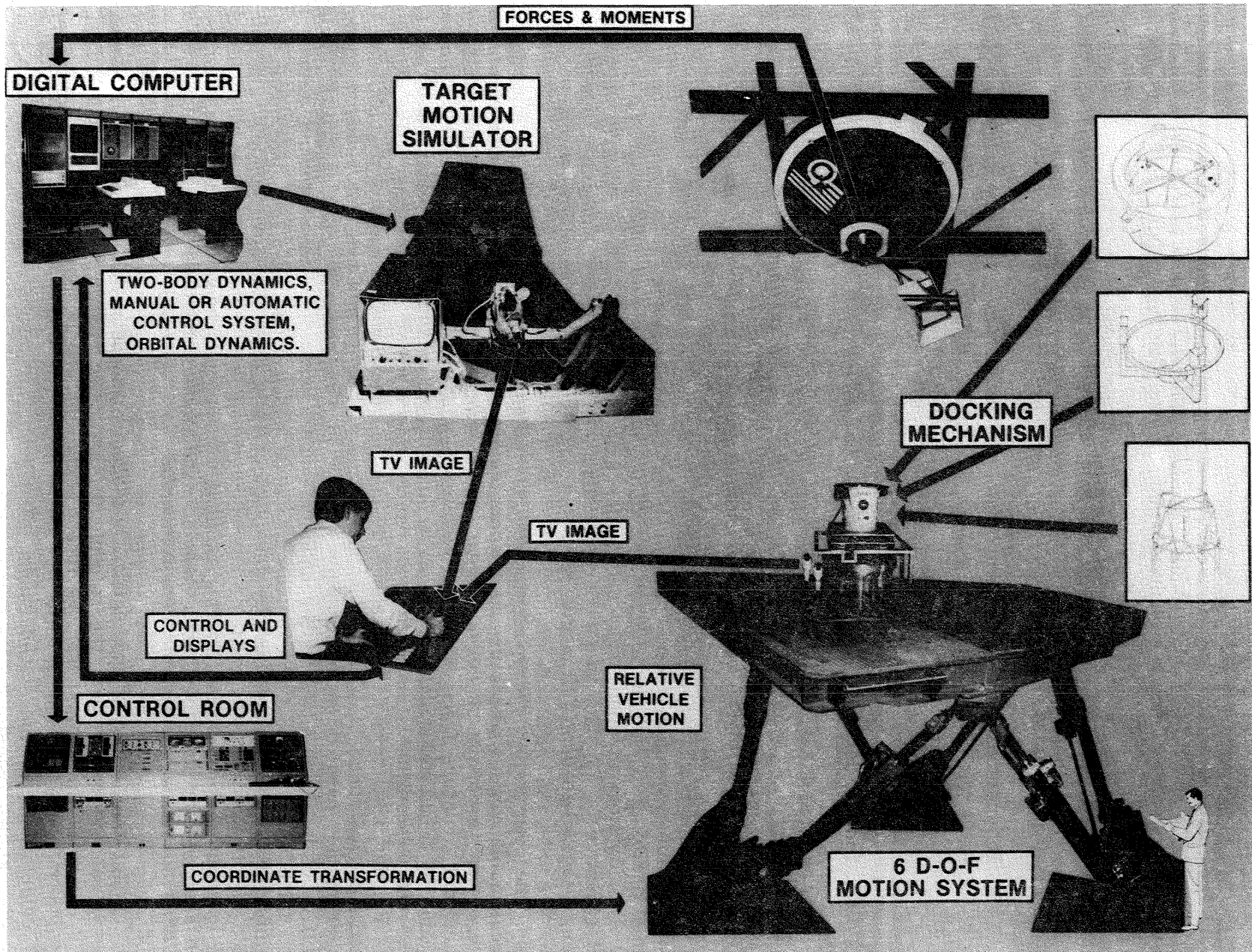


Figure 3. Orbital docking simulation facility.

## SCENE ANALYSIS BY FAST FOURIER TRANSFORMS

The initial investigation of machine vision project concentrated on the use of signal analysis techniques as provided by the classical frequency transform methods. Performance goals of an ideal machine vision system for recognizing orbital target vehicles include:

- 1) Independence of the target image position in the field of view (i.e., the image may have horizontal and vertical displacements of the line of sight from the optical bore sight axis).
- 2) Independence of orientation (i.e., the image may have any relative rotation including being upside down or sideways).
- 3) Independence of relative size of the image (i.e., distance to the target is not critical for identification).
- 4) Independence of the image illumination intensity.
- 5) Independence of the apparent distortion of a 3-D object when viewed in 2D by oblique or perspective viewing angles.
- 6) Immunity to the background against which the image is viewed.

Elementary machine vision techniques such as template matching used in assembly line inspection do not provide these capabilities [1]. However, Fourier transforms of an image are independent of object position and orientation (1 and 2 above). By normalization, they can be made to be independent of relative size as well (3 above). The use of Fourier transforms will reduce the difficulty of recognizing images because of this invariance to displacement, orientation, and size.

The computation requirements for use of Fourier transforms can be substantial. A reasonable amount of picture resolution for identification of orbiting vehicles requires on the order of 256 horizontal by 256 vertical picture elements or pixels. A single TV picture frame with this resolution would contain 65,536 pixels. The calculation of two-dimensional (2D) discrete transforms of this many points requires 4,295,000,000 multiplications and a similarly large number of additions. Use of the Fast Fourier Transform (FFT) technique reduces these multiplications to 1,049,000 or a reduction of about 1/5000 [2,3]. The number of additions are also reduced by a similar ratio. Image identification for objects with dynamics as slow as orbital docking requires computation times on the order of once a second. The computing device used for 2D FFT calculations for this research was a model FPS 5205 array processor manufactured by Floating Point Systems, Inc. It is a special purpose device which has a 38-bit word length, contains optimized routines for image analysis calculations, and is capable of 12 million floating point operations per second. This was equipped with a second input/output port, denoted as a General Purpose Input/Output (GPIOP), which would provide direct interface to a computer compatible TV camera.

Despite its speed of calculation, the array processor was not capable of processing each video image at the rate of 30 frames per second normally generated by a TV camera. However, near real time calculations can be performed by "grabbing" a single TV frame in real time by the array processor memory and performing the 2D FFT calculations on this stored image. The transform calculations and comparisons of this with known transform data of the target vehicle should be accomplished in

approximately 1 sec if the performance goals of this research are met. The implementation of this signal processing technique for developing an automatic docking capability is shown in Figure 4.

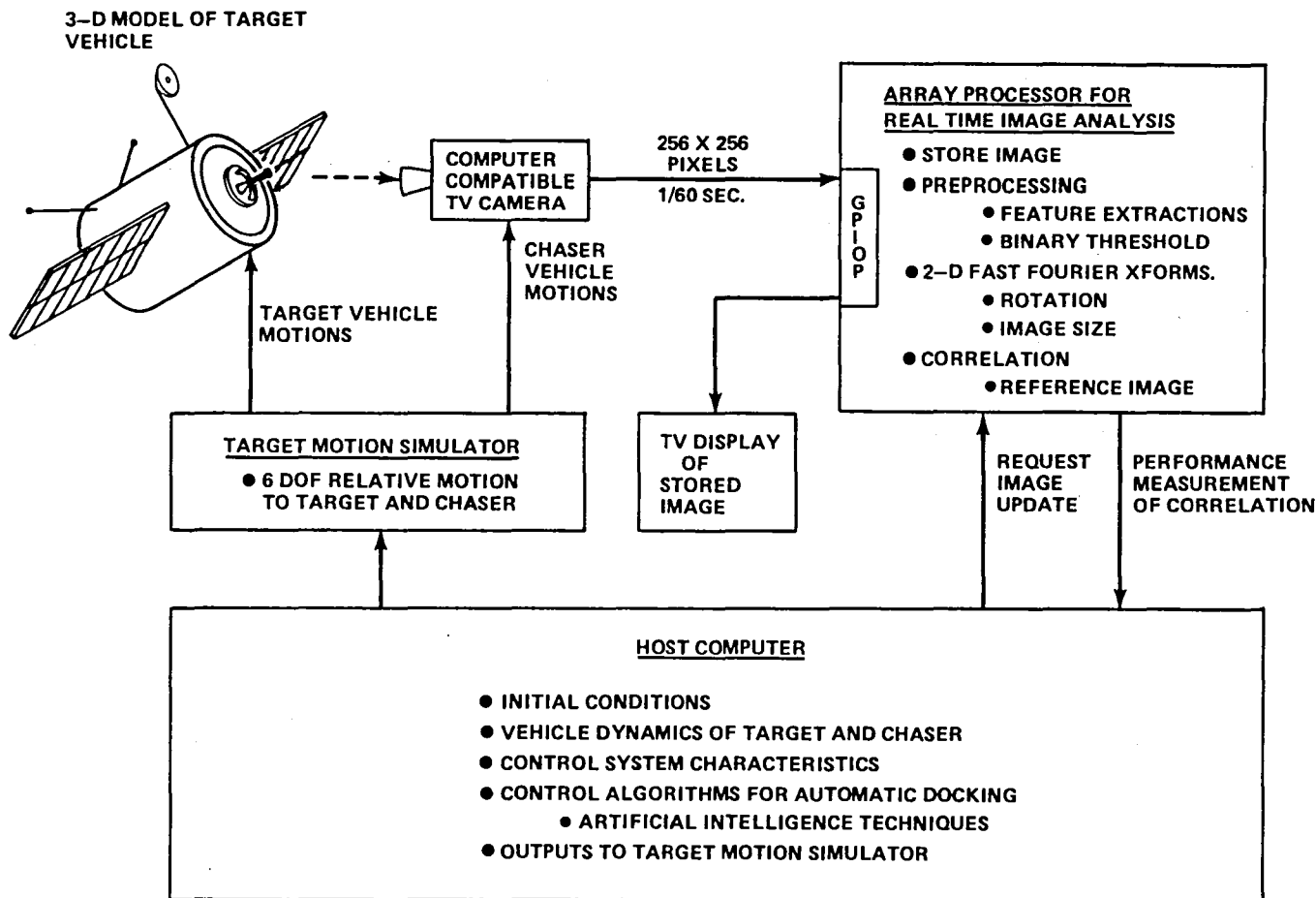


Figure 4. Development/evaluation of machine vision for real time automated assembly.

## TWO DIMENSIONAL FAST FOURIER TRANSFORM IMPLEMENTATION

Three distinct tasks were accomplished in implementation of 2D FFT's for scene analysis. The initial task was to interface a Hamamatsu model C-1000-00 computer compatible TV camera to the VAX 11/750 simulation computer. The second task was to implement array processor programs which could properly receive and manipulate the camera scene data. The third task consisted of developing software techniques for image recognition.

### Camera Interface

The interface between the Hamamatsu TV camera system and the VAX computer was to be accomplished in two methods. One method was to use the DR11-W interface board. This would allow Direct Memory Access (DMA) data transfer from the camera

to the VAX computer. The second method was to use the GPIOP on the array processor for storing the data directly into the array processor and then transfer the results into the computer. Time constraints prevented implementing this second method. The first method, using the DR11-W interface board was accomplished. To improve the transfer speed and to prevent the camera from timing out, the DR11-W interface board was placed at the front of the VAX unibus, thereby giving the camera system a higher priority than the majority of the other devices.

The Hamamatsu TV camera system was delivered with operating software. It included a demonstration program and a series of Fortran callable subroutines to scan and sample the picture. This software allowed use of one-to-sixteen scan lines and a resolution from 256 to 1024 pixels. For this project, a resolution of 256 x 256 and four scan lines was used for this software. However, when these numbers were used, modification to the Hamamatsu software was necessary. The calculations that the program used to unscramble the intensities was found to be incorrect. To verify the location of the image, the intensities were stored in a 256 x 256 matrix, a program was written then developed to read the intensities from the camera and translate them into a hardcopy via an overprinting matrix. The overprinting matrix consisted of a 6 x 36 matrix of various characters. These characters, when overprinted, displayed 36 gray scale levels. The output from the camera was scaled down to 36 gray scale levels from 256 gray scale levels. This program successfully showed the matrix transfer from the camera to the VAX 11/750 retained all information and that the orientation was correct.

#### Two Dimensional Fast Fourier Transform

There are many different methods available for image recognition. The first approach taken in this project was to use a 2D FFT on the 256 x 256 matrix. The program flow consisted of the following:

- 1) Scan the picture.
- 2) Store the intensities.
- 3) Create a complex matrix.
- 4) Multiply by a centering matrix.
- 5) Run the 2D FFT routine.
- 6) Scale the real and imaginary numbers
- 7) Convert to magnitude and phase.
- 8) Use the magnitudes for image recognition.

The picture was scanned using four sampling lines and a resolution of 256 x 256 intensities was stored. This resolution was reduced to a 32 x 32 matrix so that the intensities could be easily viewed at a computer CRT display terminal. Multiplication of a centering matrix was needed so that the output of the Fourier Transform could be easily interpreted. The method used would shift the origin of the transform to the center frequency point (16, 16) of a 32 x 32 frequency square. This shift did not affect the magnitudes of the output of the Fourier Transform. These centered values were stored into the array processor and a 2D FFT was

implemented. The output was scaled and then converted to magnitudes using sub-routines provided with the array processor. After it was determined that the program would work, the resolution size was changed to 64 x 64 and then to 128 x 128. A 256 x 256 image could not be processed due to temporary restrictions of 256K words of common memory in the model 5205 array processor used.

### Image Recognition

Several images were processed through the program and the values entered into a data base to be used for real time recognition. Six images were selected and processed. Recognition by use of 2D FFT data on each of these images was accomplished in 7 to 13 sec; depending upon the location of the image in the data base. The faster recognition times corresponded with the image being at the beginning of the data base, whereas the slowest times correspond with the last image in the data base. The real image could be positioned at any point within the field of view of the camera and it would still be recognized. If the image was rotated, it would be recognized. It was noted that 2D FFT data for 180 deg rotation was identical and made recognition easier. If project time permitted, size normalization would have been developed so that image size would be immaterial. Also, the recognition times could have been reduced if portions or all of the program were written in Assembly Language and if direct transfer of the image intensities to the array processor were used.

### REAL TIME VIDEO INTERFACE

Initial investigations of image recognition indicated that faster solution times were required to achieve the goal of 1 sec recognition of typical OMV target vehicles. These initial investigations utilized a DMA input/output (I/O) device which is the fastest input method available to the VAX simulation computer. The input time for a 256 x 256 pixel image from the Hamamatsu camera is approximately 1/2 sec by this approach. Although the DMA interface is capable of a throughput rate of 500K words per second (one 256 x 256 video frame with 8 bits per pixel gray scale is 64K bytes, 32K words), the bus arbitration greatly limits data transfer. Sliced video data can be transferred at a higher rate because the data is compressed to one bit gray scale per pixel (black or white) resulting in 1/8 the data per video frame.

The GPIOP of the FPS-5205 array processor is a dedicated high speed (6 MHz) input/output processor. By utilizing the GPIOP, data may be input at a rate of 3,000,000 38-bit words per second. In addition, the data is transferred directly into the FPS-5205, saving the time required to transfer data from the VAX into the array processor.

The Hamamatsu camera updates at 60 frames per second in the 256 x 256 pixel mode, hence, it takes 1/60 of a second to generate one video image. This time of 1/60 of a second is the lower limit on the signal conversion and transmission lag time. In order to minimize the signal conversion and transmission lag time, a real time interface was designed to input one video frame into the GPIOP in the minimum time possible — 1/60 of a second. A block diagram of the interface is shown in Figure 5. Design of this real time interface is described in the appendix.

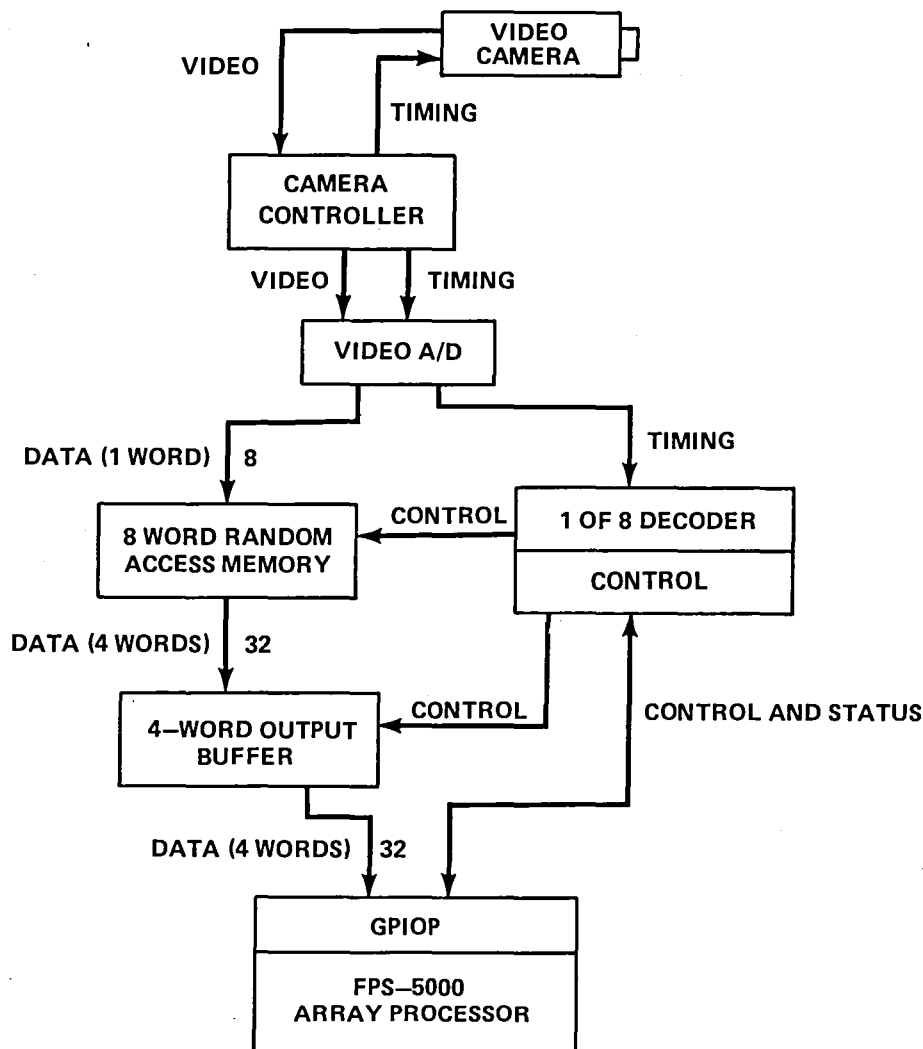


Figure 5. Video camera to array processor real time interface.

### SCENE ANALYSIS BY SYNTACTIC PATTERN RECOGNITION

An alternative to the Fourier transform approach to image analysis and target vehicle identification has been developed by Frank Vinz (one of the authors of this report) using techniques derived from syntactic pattern recognition. This technique employs a tree graph for representation and description of a scene. It is considered to be a very efficient way to characterize an object in a scene since only a small fraction of the total image data need be permanently stored. There is a corresponding reduction in computation requirements as compared to the 2D FFT technique.

This method is unique in that it does not have the limitations of possibly omitting major sections of the image which may be the case with vertical or horizontal line scanning reported in the literature. To prevent this possible omission, a box scan is employed which starts at the centroid of the image and increments outward



in larger and larger layers until the complete scene is examined. An example of this scan process on a 8 x 8 scene matrix is shown in Figure 6. The scene includes a "T" image consisting of 16 binary "1's" while all other possible pixels are background elements having low level video intensity and have been pre-processed as binary "0's". The complete scene is examined by a 3 x 3 scan window using the box scan technique and the resultant tree graph is also shown in Figure 6. The longest branches of the tree have end points which correspond to the major end points of the "T" image on the 8 x 8 scene. This tree graph representation has a capability of uniquely describing an image without requiring storage capacity for the complete 8 x 8 matrix scene. This capability has significant reduction in memory for high resolution scenes having a large number of pixels such as 256 x 256 matrix. An even greater significance is the reduction of computations required for image recognition as will be shown. The method is referred to as Analysis of Images by Box Scan and Syntax or "AI BOSS" and it is felt to have a good potential for autonomous control of orbital servicing vehicles such as the OMV during critical docking and assembly operations.

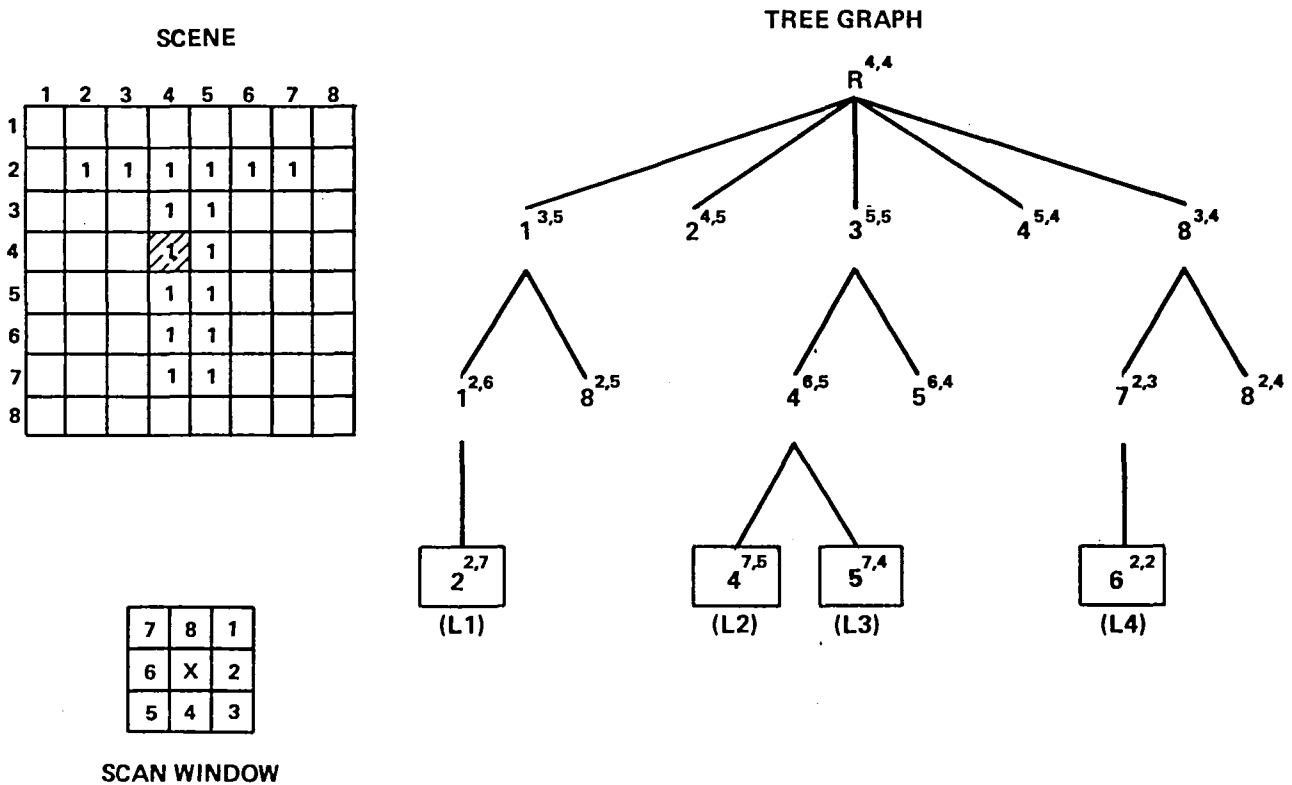


Figure 6. Tree graph of "T" image.

Specific procedures for implementing the "AI BOSS" technique involves the following steps:

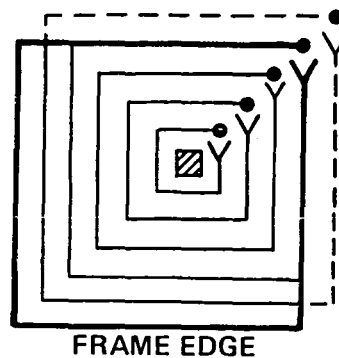
1) By means of a computer-compatible TV camera having pre-processing for binary slicing of a video image, obtain a digitized representation for the scene. Each pixel representing the target image is denoted by a binary "1" and the background pixels are a binary "0." This black and white silhouette of the target is referred to as a binary image.

2) Compute the centroid pixel of the target image.

3) Create the root and initial branches of the tree graph by centering a 3 x 3 pixel scan window at the centroid. Figure 6 illustrates this process. Proceed to examine in numerical sequence each of the 9 pixel locations of the scan window. The centroid pixel will be the root "R" of the tree graph providing it is a binary "1." The initial branches of the tree graph are any of the surrounding pixels of the scan window which are also a binary "1." The branches are placed on the tree graph in a left to right order in the sequence in which they were scanned. Each tree graph element is identified by its corresponding scene matrix row and column. In case the centroid is not a binary "1" then the root of the tree graph is the first binary "1" which occurs in the box scan sequence after the centroid is examined.

4) The box scan is then moved so that the scan window is sequentially centered about the 8 pixel locations identified in (3) above. All binary "1" pixels on this relocates scan window which are horizontally, vertically, or diagonally adjacent (i.e., neighbors) to previously identified tree elements are included on the graph as branches of these earlier identified elements. Each pixel so identified is drawn on the tree graph and denoted by its row and column on the scene matrix as in (3) above.

5) The box scan examines the next outer layer by incrementing one pixel to the right and one pixel up from the starting point of the preceding box scan layer and then performing a new box scan. This box scan procedure is depicted to the right. The scan process continues to grow until all pixels on the complete scene matrix are examined. In the case that the image centroid does not coincide with the center of the scene matrix, then the box scan must continue as though there existed additional pixel layers outside the scene. This insures that all scene pixels will be examined for potential inclusion in the tree graph.



6) Each box scan should proceed to investigate all possible binary image pixels in that particular layer of scanning, even if the center of the scan window at any pixel location is not previously assigned to the tree graph.

7) In the event a pixel location which should be the center of a scan window but has not been identified as an element of the tree graph, assign it as a tree graph descendant to its neighbor based on the box scan priorities of 1 to 8.

8) If an unassigned pixel cannot be identified for a tree graph connection as in (6) above, perform an auxiliary layered box scan to establish the tree graph connection. Terminate this auxiliary box scan when either unidentified pixels are connected or else the auxiliary scan has grown in size equal to the current box scan.

9) After a pixel is identified as an element of the tree graph it cannot appear elsewhere on the tree.

10) Once elements of a tree graph are identified, subsequent scan window searches are made only from these elements. The order of scanning from these identified elements is determined by the sequence in which they were identified by the rules of the box scan. In this way, there will be fewer problems with disjointed elements in the process of building the tree graph.

11) In the case pixels are found which are not neighbors to existing tree graph elements, then they are temporarily tagged for auxiliary searches to determine where they should be placed. Each "tagged" pixel will be examined after each pixel search in the box scan procedure. If a candidate pixel does not have a neighbor that is an element of the tree, it will continue to be tagged for auxiliary searches. In case several pixels are tagged for auxiliary searches, they will be examined for possible incorporation into the tree in the sequence by which they were originally tagged. After completion of all box scans and auxiliary searches, if tagged elements remain which have no neighbors that are elements of the tree graph, then these tagged elements are discarded and are considered to not be a part of the image. They may be the result of noise in the video picture or they may be bright background objects.

12) An image is then characterized by the length of the major branches on the tree graph and by the branch end point locations (i.e., the row and column numbers from the scene matrix). For the examples studied in which the frame matrix is 8 x 8, the four longest distinct branches were selected as major branches and were sufficient to identify the image. Distinct branches are separated from each other by a maximum number of tree elements and are also positionally spread out over the tree shape. Refer to Figure 13 for an example of the use of distinct branches. In the case where distinct branches of equal length are in contention as major branches, the order of selection is left to right as they appear on the tree.

13) For more realistic cases where the scene matrix may have many more elements such as 256 x 256, then more than four major branches may be selected to better characterize the image. To provide distinction between end points in this case, the major branches selected should have at least 10 or 20 unique elements at the end. Otherwise the branch and points may correspond to adjacent elements of the image in the scene matrix and would not be very effective describers.

14) Images of orbiting targets may be recognized by the relative length of each major branch. This constitutes a powerful pattern recognition capability in that general shapes are characterized by some of the same features a person uses such as: elongation, compactness, symmetry, relative length of appendages, etc.

15) The method exhibits a generally consistent mapping property that is somewhat analogous to conformal mapping. The clockwise sequence of the image extremities in the scene matrix is usually identical to the left to right sequence of major end points on the tree graph. This characteristic is clearly shown in Figures 6 through 17.

16) For those targets selected for closer inspection, servicing, or docking, the chaser vehicle must rotate itself for relative alignment by the use of the length and the end point locations of the major branches. Comparison of both the length of the major branches and the location of the end points of each major branch will provide a template matching for unique identification of the image.

The "AI BOSS" technique for tree graph construction has several advantages over the horizontal or vertical line scan methods that are presented in the literature. The pros and cons of both methods are compared in Table 1. Examination of these advantages and disadvantages leads to a dominate case for the box scan procedure.

The "T" image shown in Figure 6 was examined for several orientations by use of "AI BOSS." The following conditions were investigated:

Figure 7 - "T" pattern rotated 90 deg CW.

Figure 8 - "T" pattern rotated 180 deg CW.

Figure 9 - "T" pattern rotated 270 deg CW.

Figure 10 - "T" pattern rotated 315 deg CW.

Figure 11 - "T" pattern with 2 pixels shifted.

All of these cases consistently show four major tree graph branches and all are approximately equal in length. The end points of each of the tree branches correspond correctly with the extremities of the "T" image on the scene matrix. This correspondence between branch end points and image extremities is a strong feature of the "AI BOSS" method and is a direct result of the box scan technique.

Data obtained from comparisons of tree graph test cases with the reference tree graphs may be used to provide proper orientation between the chaser and target vehicles. Comparisons of branch length ratios and branch end points are used for performance measurements as shown in Table 2. The performance measurement may have plateaus and local minimums which must be contended with in the chaser vehicle roll search control law. An indication of the performance measurement for the "T" patterns is depicted in Figure 12 for the five test cases studied.

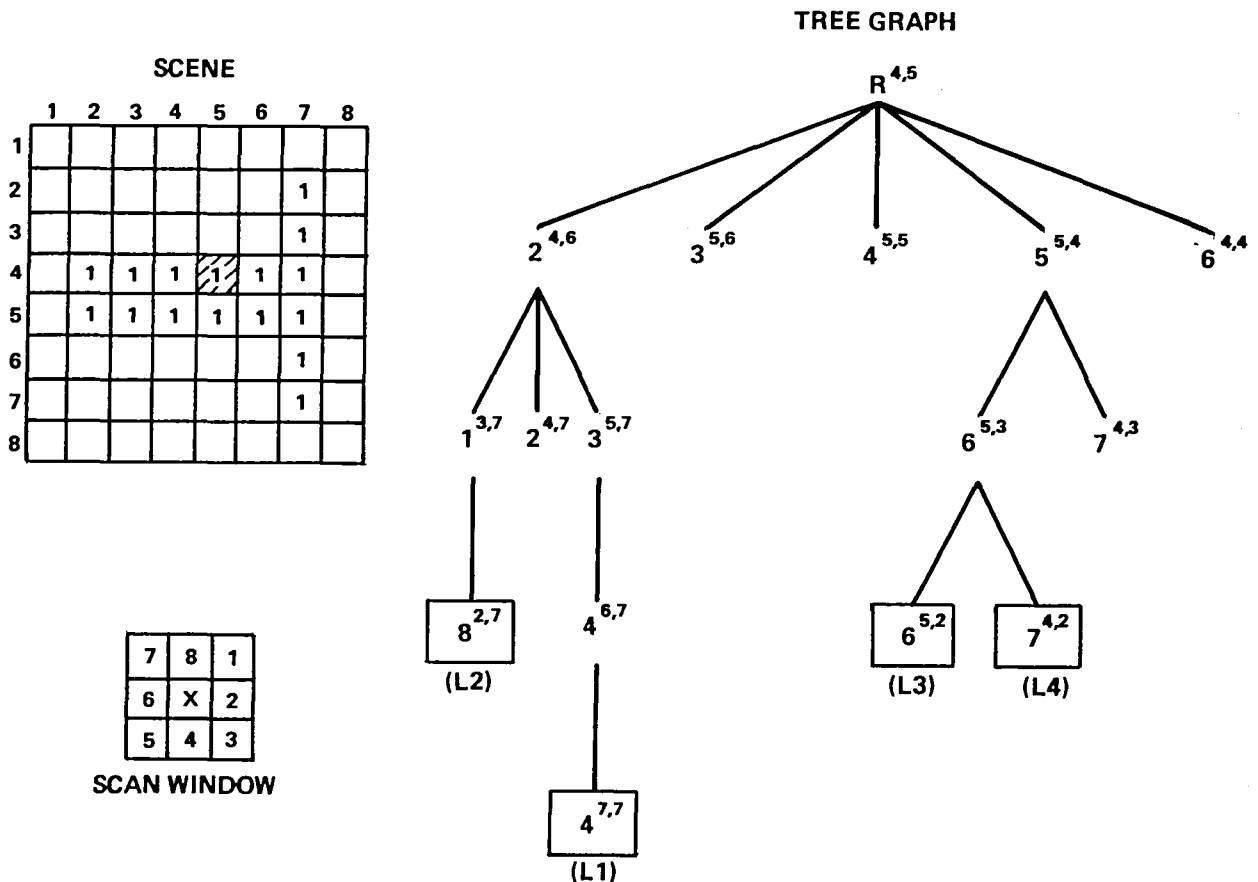


Figure 7. Tree graph of "T" image rotated 90 deg CW.

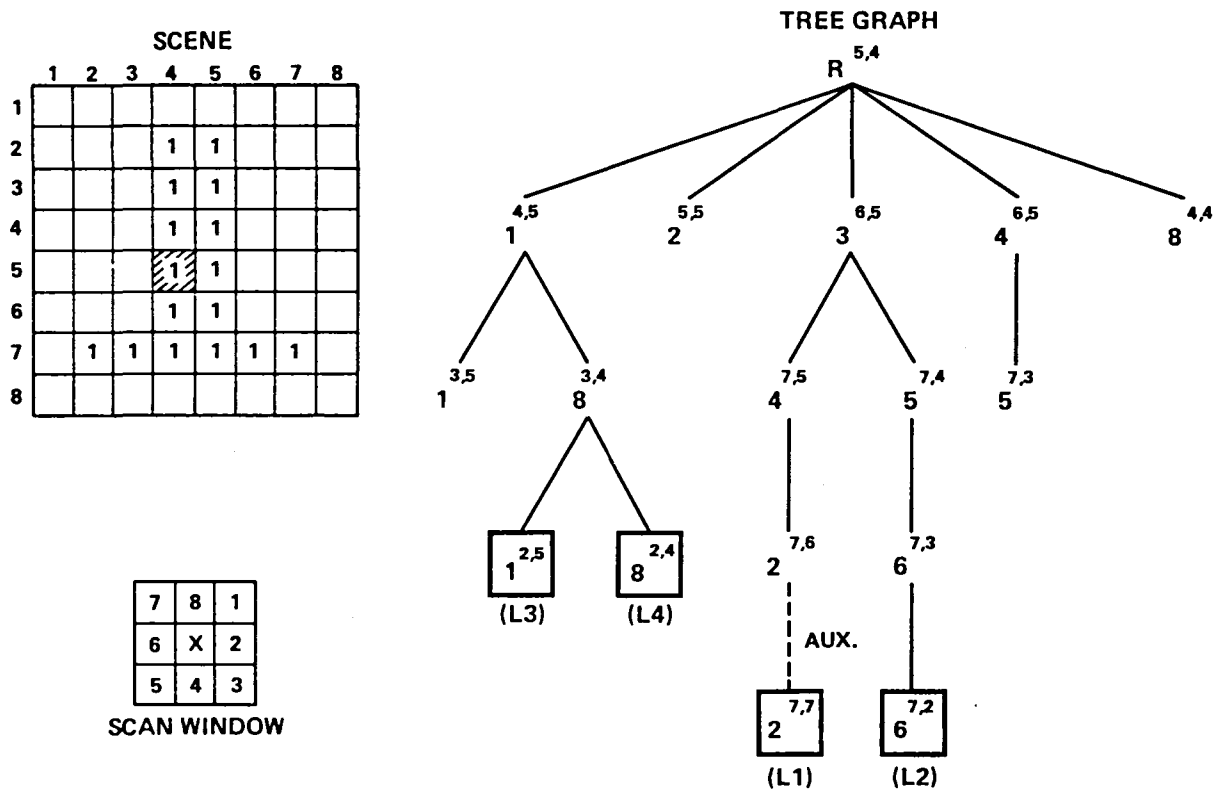


Figure 8. Tree graph of "T" image rotated 180 deg CW.

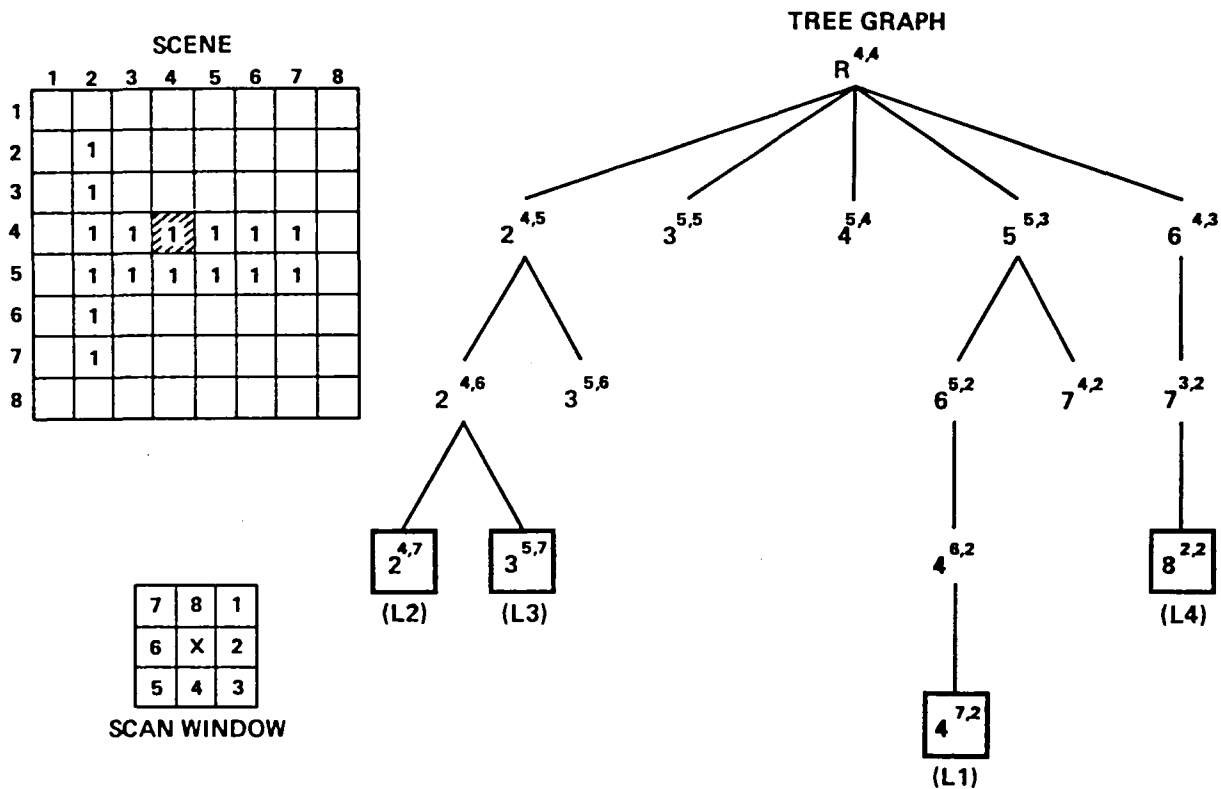
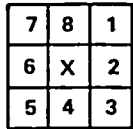
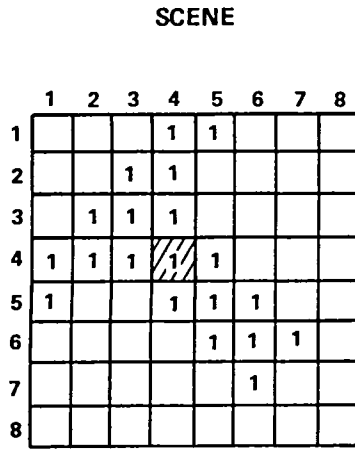


Figure 9. Tree graph of "T" image rotated 270 deg CW.



**SCAN WINDOW**

**TREE GRAPH**

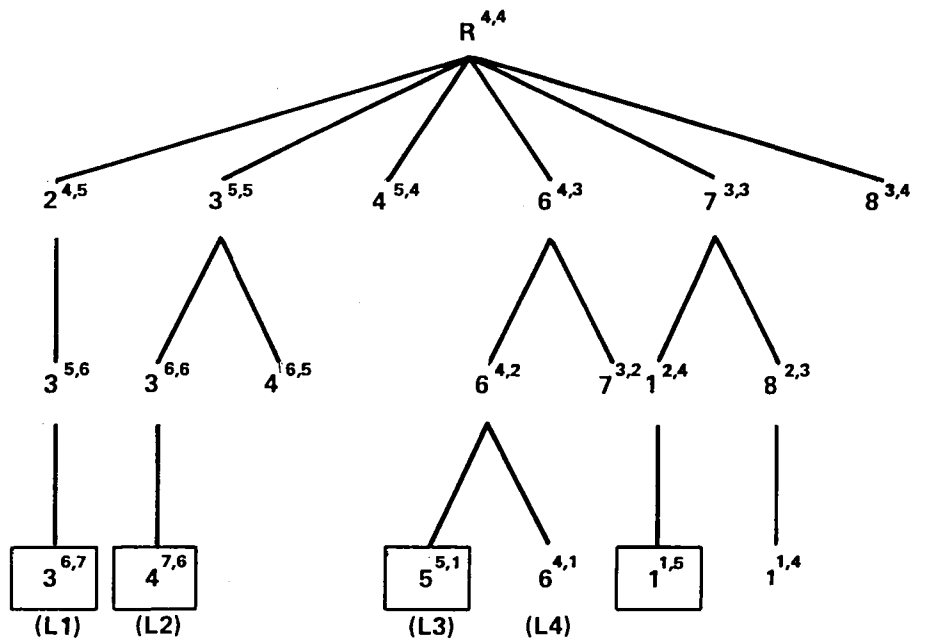
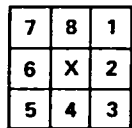
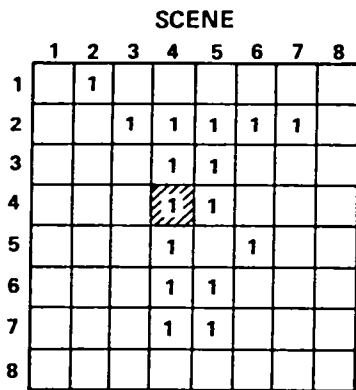


Figure 10. Tree graph of "T" image rotated 315 deg CW.



**SCAN WINDOW**

**TREE GRAPH**

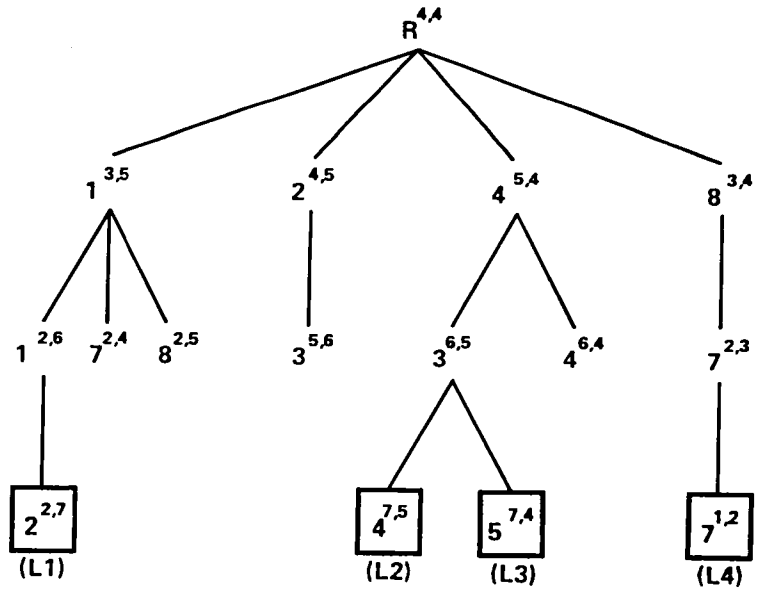


Figure 11. Tree graph of "T" image with two pixels shifted.

TABLE 1. COMPARISON OF BOX SCAN AND LINE SCAN METHODS

**ADVANTAGES**

**BOX SCAN ("AI BOSS")**

- ALL MAJOR FEATURES OF IMAGE WILL BE REPRESENTED ON THE TREE GRAPH
- CONSISTENT TREE GRAPHS ARE GENERATED REGARDLESS OF IMAGE ROTATION
- IN GENERAL, CONFORMAL MAPPING LIKE CHARACTERISTICS ARE EXHIBITED
- DUE TO THIS MAPPING CHARACTERISTICS, A POWERFUL PATTERN RECOGNITION CAPABILITY RESULTS IN THE ABILITY TO DISTINGUISH SHAPES SUCH AS SYMMETRY, COMPACTNESS, ELONGATION, RELATIVE LENGTHS OF APPENDAGES, ASPECTS RATIOS, ETC.

**HORIZONTAL LINE SCAN**

- IN THEORY, ONLY ENOUGH MEMORY IS REQUIRED TO STORE 3 TV RASTER SCAN LINES (768 WORDS FOR A 256 x 256 FRAME SIZE)
- REAL TIME PROCESSING OF TV FORMAT MAY BE POSSIBLE IF 25 NANOSECOND MEMORY TIME CAN BE IMPLEMENTED

**DISADVANTAGES**

**BOX SCAN ("AI BOSS")**

- REQUIRES STORAGE OF COMPLETE VIDEO FRAME (65,536 WORDS FOR 256 x 256 TV FRAME) PRIOR TO START OF SCAN
- REQUIRES CALCULATION OF IMAGE CENTROID FROM WHICH THE SCAN EMANATES (HOWEVER, THIS CALCULATION IS ALSO USED FOR PITCH AND YAW POINTING DURING CHASER SEARCH AND DOCKING OPERATIONS)

**HORIZONTAL LINE SCAN**

- MAY OMIT MAJOR BRANCHES ON THE TREE GRAPH BY FAILING TO DETECT CONNECTIVITY PROPERTIES OF THE IMAGE
- TREE ROOT IS USUALLY AN EXTREMITY OF THE IMAGE THUS CAUSING INCONSISTENT TREE GRAPHS
- TREE GRAPH IS HIGHLY DEPENDENT ON ORIENTATION OF THE IMAGE
- THERE IS NO CONFORMAL MAPPING LIKE CHARACTERISTIC

TABLE 2. PERFORMANCE MEASUREMENTS OF "T" PATTERNS

	DATA & CALCULATIONS	ΣA. (2)	ΣB. (3)	ΣA. (2) + ΣB. (3) PERFORMANCE MEASUREMENT
I. REFERENCE IMAGE "T" A. TREE LENGTHS L1, L2, L3, L4 B. END POINTS	4, 4, 4, 4 (2,7), (7,5), (7,4), (2,2)	-	-	0
II. TEST CASE - 90° CW A. (1) TREE LENGTHS (2) $10 \left  \text{RATIO} \frac{IA}{IIA(1)} - 1 \right $ B. (1) END POINTS (2) REF. END POINTS (3) $ II B(1) - II B(2) $	5, 4, 4, 4 2.5, 0, 0, 0  (7,7), (2,7), (5,2), (4,2) (2,7), (7,5), (7,4), (2,2) (5,0), (5,2), (2,2), (2,0)	2.5	18	20.5
III. TEST CASE - 180° CW A. (1) TREE LENGTHS (2) $10 \left  \text{RATIO} \frac{IA}{IIIA(1)} - 1 \right $ B. (1) END POINTS (2) REF. END POINTS (3) $ IIIB(1) - IIIB(2) $	5, 5, 4, 4 2.5, 2.5, 0, 0  (7,7), (7,2), (2,5), (2,4) (2,7), (7,5), (7,4), (2,0) (5,0), (0,3), (5,1), (0,2)	5	16	21
IV. TEST CASE - 270° CW A. (1) TREE LENGTHS (2) $10 \left  \text{RATIO} \frac{IA}{IV A(1)} - 1 \right $ B. (1) END POINTS (2) REF. END POINTS (3) $ IV B(1) - IV B(2) $	5, 4, 4, 4 2.5, 0, 0, 0  (7,2), (4,7), (5,7), (2,2) (2,7), (7,5), (7,4), (2,2) (5,5), (3,2), (2,3), (0,0)	2.5	20	22.5
V. TEST CASE - 315° CW A. (1) TREE LENGTHS (2) $10 \left  \text{RATIO} \frac{IA}{VA(1)} - 1 \right $ B. (1) END POINTS (2) REF. END POINTS (3) $ VB(1) - VB(2) $	4, 4, 4, 4 0, 0, 0, 0  (6,7), (7,8), (5,1), (4,1) (2,7), (7,5), (7,4), (2,2) (4,0), (0,1), (2,3), (2,1)	0	13	13
VI. TEST CASE - 2 PIXELS SHIFTED A. (1) TREE LENGTHS (2) $10 \left  \text{RATIO} \frac{IA}{VI A(1)} - 1 \right $ B. (1) END POINTS (2) REF. END POINTS (3) $ VI B(1) - VI B(2) $	4, 4, 4, 4 0, 0, 0, 0  (2,7), (7,5), (7,4), (1,2) (2,7), (7,5), (7,4), (2,2) (0,0), (0,0), (0,0), (1,0)	0	1	1

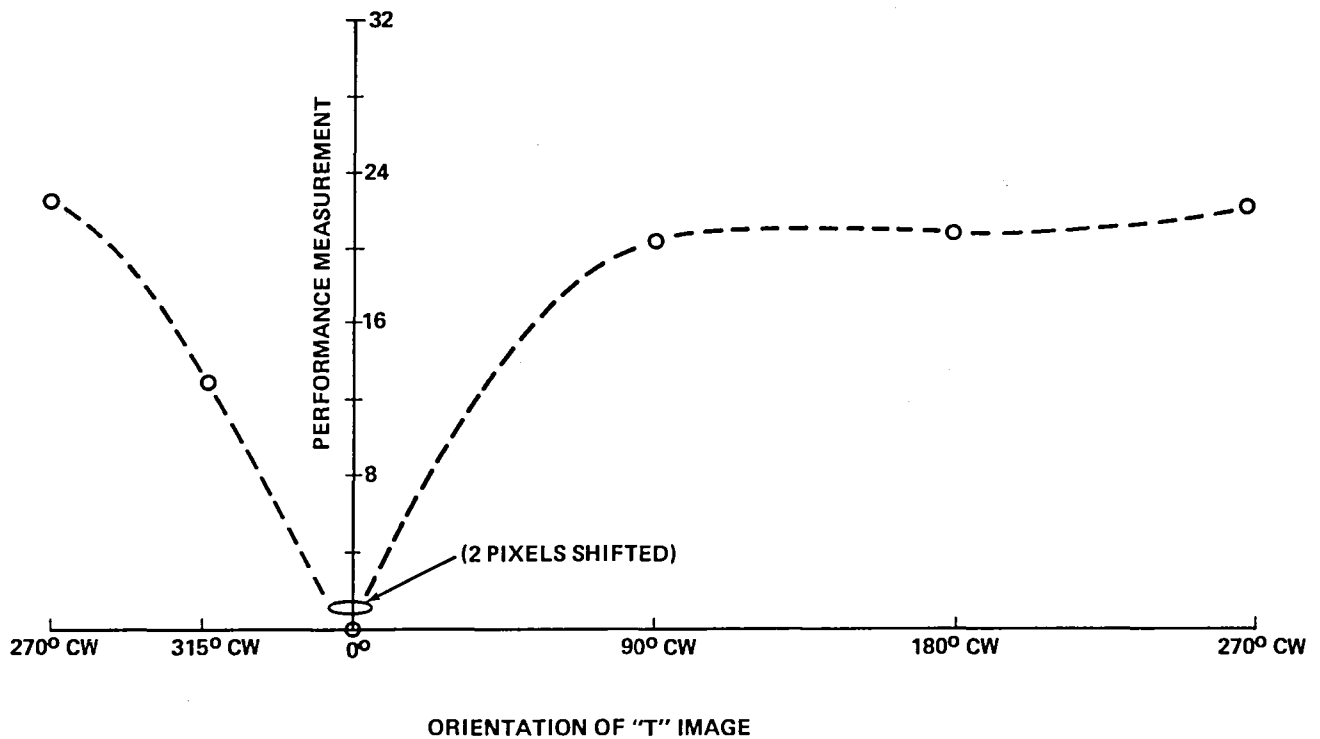


Figure 12. Performance measurement for chaser roll search of "T" pattern.

"AI BOSS" was applied to an irregular pattern for further testing. This irregular pattern consisted of 16 pixels also on an 8 x 8 scene matrix. The pattern was examined with the following orientations:

Figure 13 — Reference orientation of irregular pattern.

Figure 14 — Irregular pattern rotated 90 deg CW.

Figure 15 — Irregular pattern rotated 180 deg CW.

Figure 16 — Irregular pattern rotated 270 deg CW.

Figure 17 — Irregular pattern rotated 315 deg CW.

As in the case of the "T" pattern investigations, a performance measurement was calculated for each orientation. The results are included in Table 3. The characteristics of this performance measurement for chaser roll search control appears to be a monotonically increasing function as shown in Figure 18. However, there is no guarantee that this will occur.



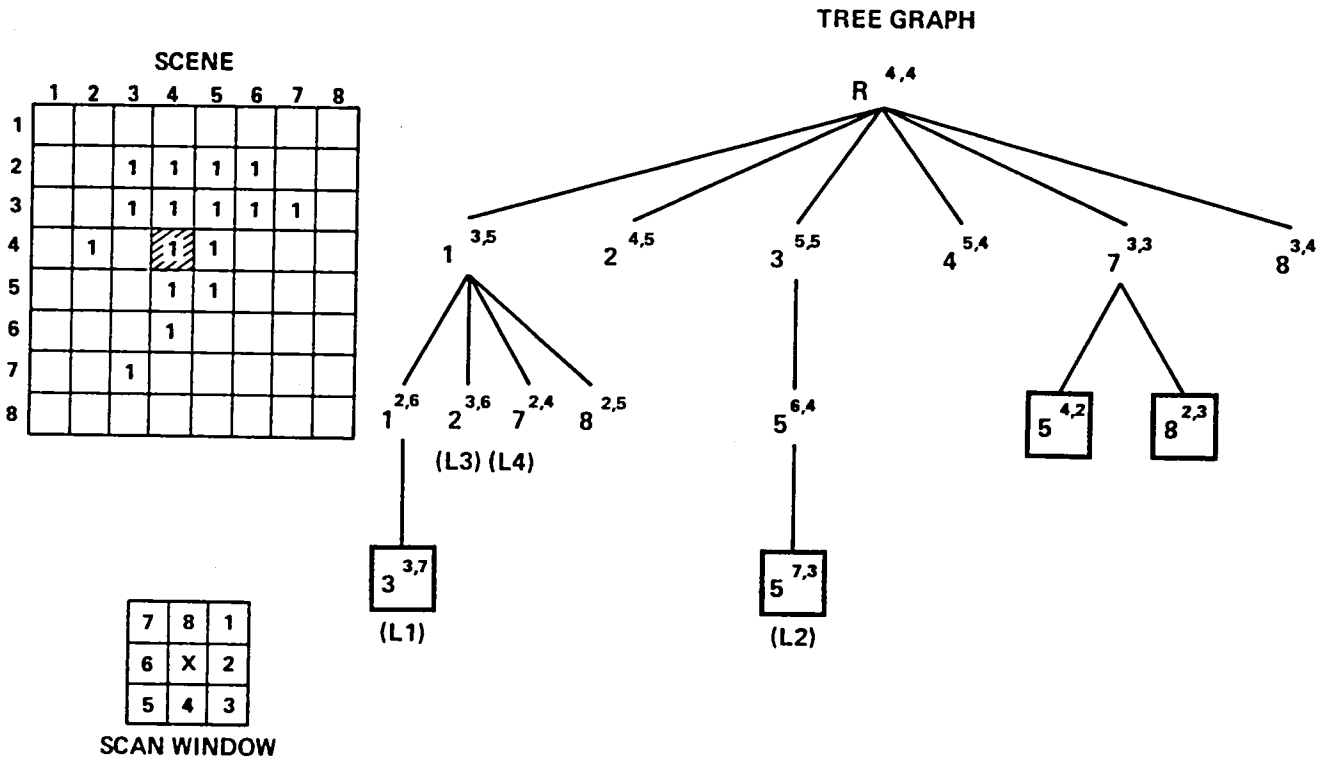


Figure 13. Tree graph of irregular pattern.

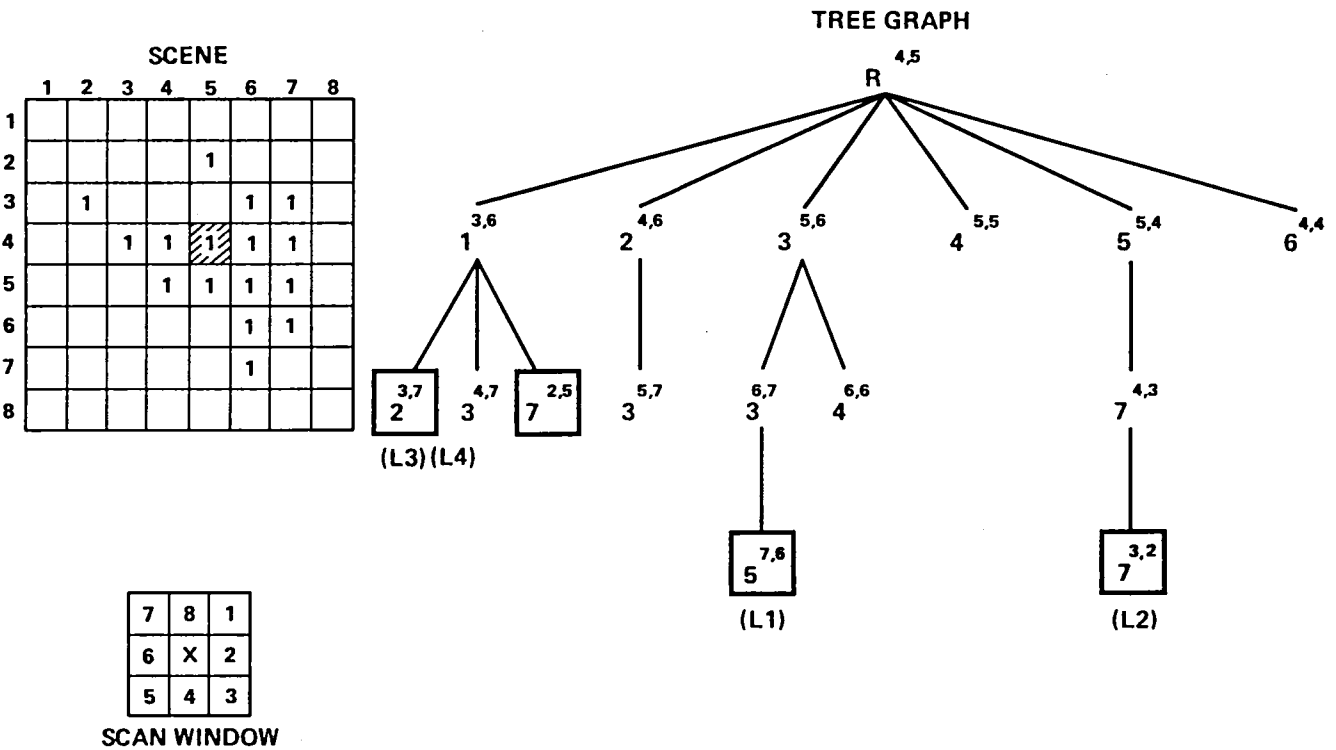


Figure 14. Tree graph of irregular pattern rotated 90 deg CW.

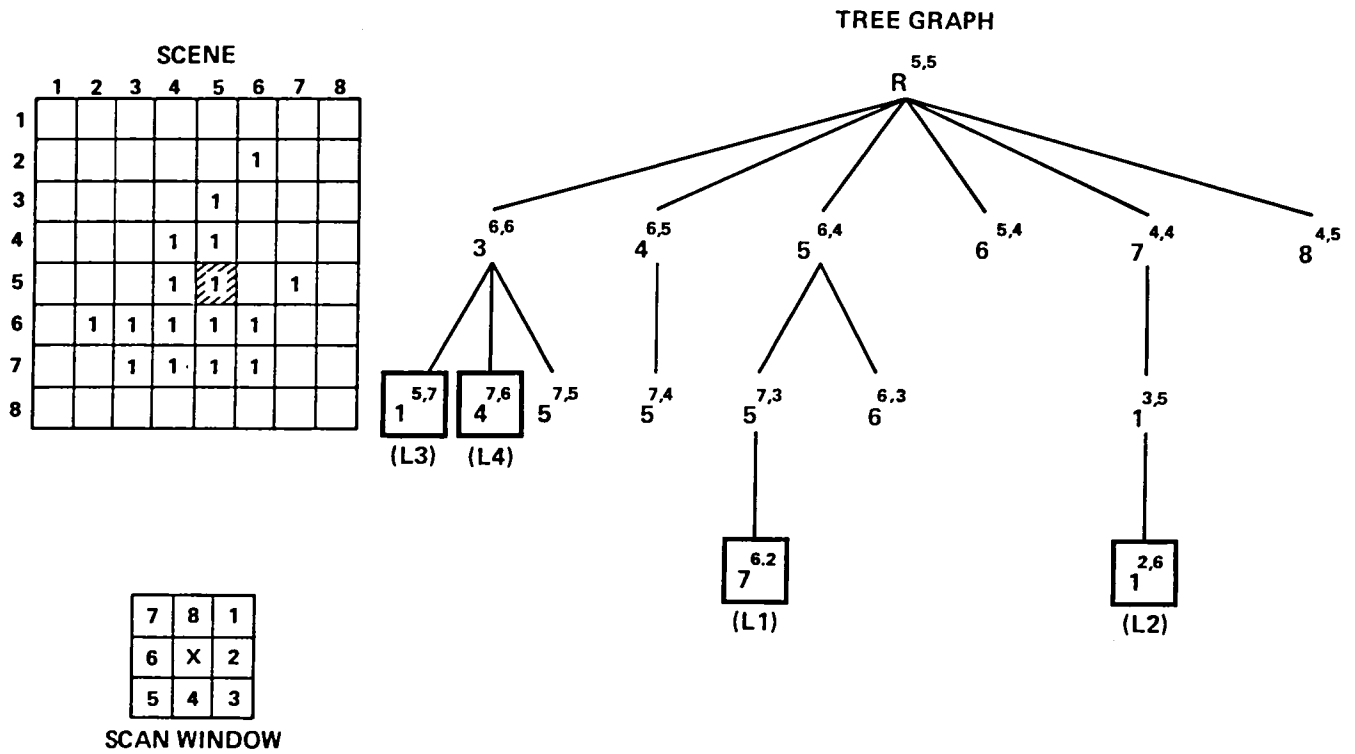


Figure 15. Tree graph of irregular pattern rotated 180 deg CW.

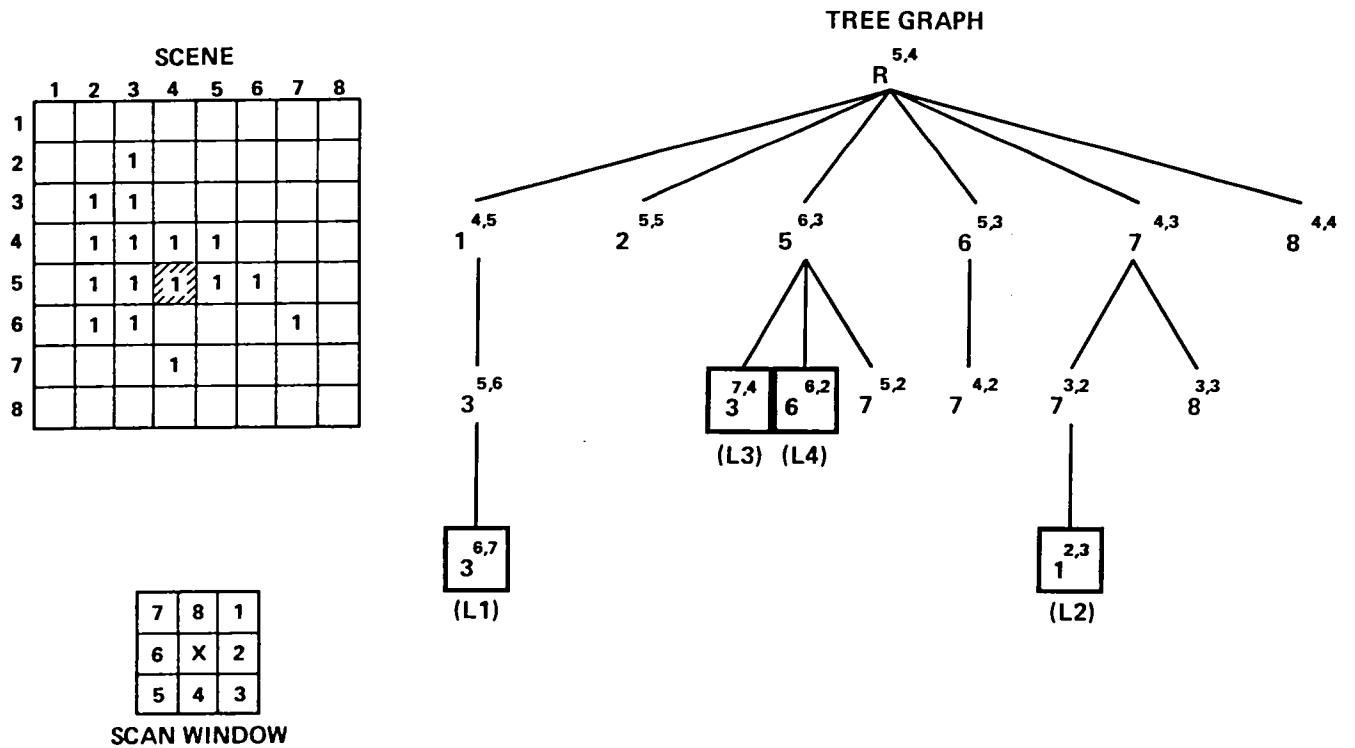


Figure 16. Tree graph of irregular pattern rotated 270 deg CW.

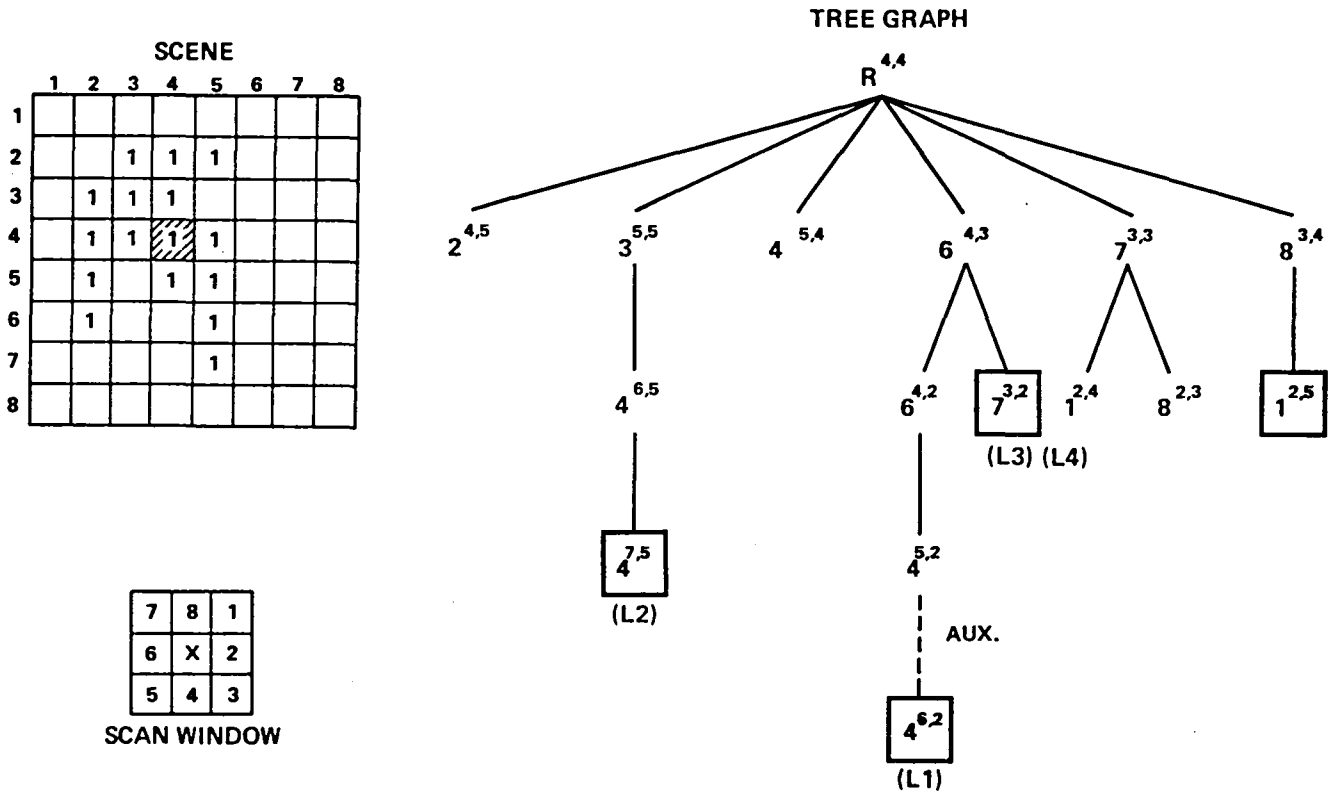


Figure 17. Tree graph of irregular pattern rotated 315 deg CW.

TABLE 3. PERFORMANCE MEASUREMENTS OF IRREGULAR PATTERNS

	DATA & CALCULATIONS	ΣA. (2)	ΣB. (3)	ΣA (2) + ΣB (3) PERFORMANCE MEASUREMENT
I. REFERENCE IMAGE - IRREGULAR PATTERN - A. (1) TREE LENGTHS L1, L2, L3, L4 B. END POINTS	4, 4, 3, 3.... (3,7), (7,3), (3,6), (2,4)	-	-	0
II. TEST CASE - 90° CW A. (1) TREE LENGTHS (2) 10  RATIO $\frac{IA}{IIA(1)}$ -1  B. (1) END POINTS (2) REF END POINTS (3)  II B (1) - II B (2)	4, 4, 3, 3.... 0, 0, 0, 0 (7,6), (3,2), (3,7), (4,7) (3,7), (7,3), (3,6), (2,4) (4,1), (4,1), (0,1), (2,3)	0	16	16
III. TEST CASE - 180° CW A. (1) TREE LENGTHS (2) 10  RATIO $\frac{IA}{IIIA}$ -1  B. (1) END POINTS (2) REF END POINTS (3)  III B (1) - III B (2)	4, 4, 3, 3.... 0, 0, 0, 0 (6,2), (2,6), (5,7), (7,6) (3,7), (7,3), (3,6), (2,4) (3,5), (5,3), (2,1), (5,2)	0	26	26
IV. TEST CASE - 270° CW A. (1) TREE LENGTHS (2) 10  RATIO $\frac{IA}{IIIA}$ -1  B. (1) END POINTS (2) REF. END POINTS (3)  IV B (1) - IV B (2)	4, 4, 3, 3.... 0, 0, 0, 0 (6,7), (2,3), (7,4), (6,2) (3,7), (7,3), (3,6), (2,4) (3,0), (5,0), (4,2), (4,2)	0	20	20
V. TEST CASE - 315° CW A. (1) TREE LENGTHS (2) 10  RATIO $\frac{IA}{IIIA}$ -1  B. (1) END POINTS (2) REF. END POINTS (3)  V B (1) - V B (2)	5, 4, 3, 3.... 2,5, 0, 0, 0 (6,2), (7,5), (3,2), (2,4) (3,7), (7,3), (3,5), (2,4) (3,5), (0,2), (0,4), (0,0)	2.5	14	16.5

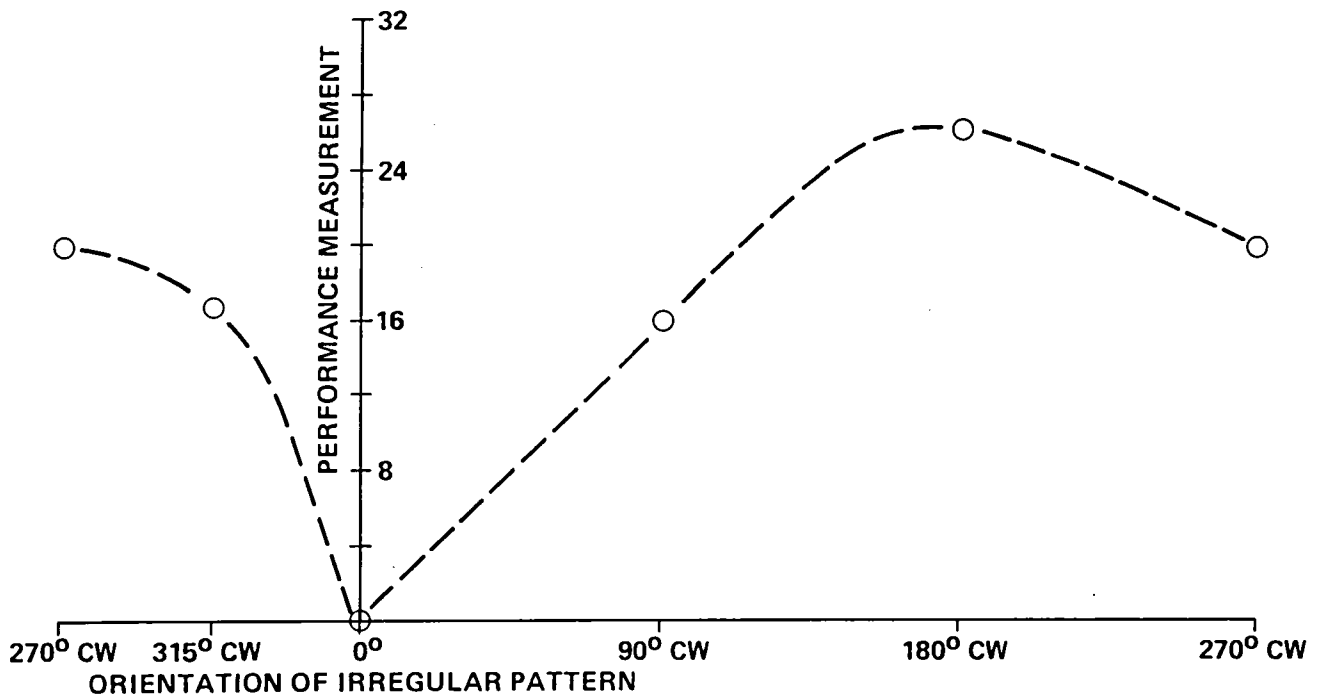


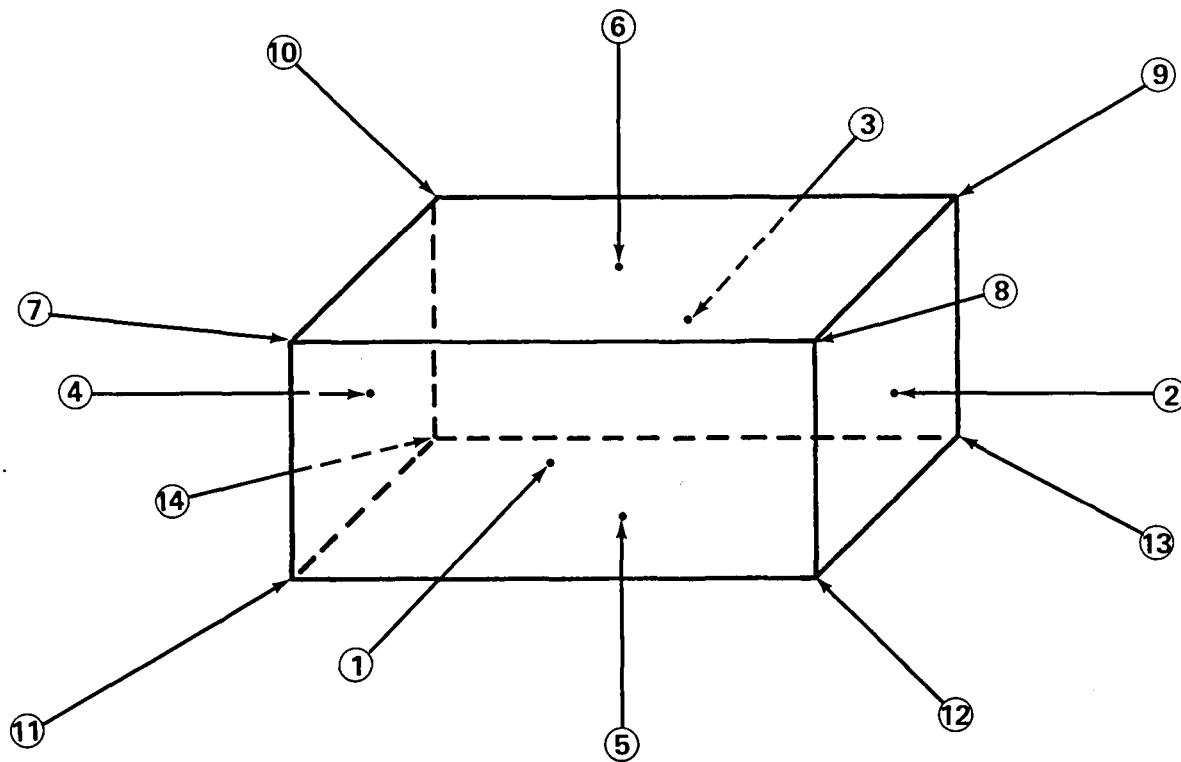
Figure 18. Performance measurement for chaser roll search of irregular pattern.

#### APPLICATION OF "AI BOSS" TO REPRESENTATIVE TARGET VEHICLE

The box scan and tree graph procedure of "AI BOSS" was applied for analysis of an image more detailed than the elementary figures initially described. Typical reference views of target vehicle are shown in Figure 19. The image selected was that of the Space Telescope shown in Figure 20. The binary image resulting from a threshold slicing is illustrated in Figure 21. Figure 22 is a 32 x 32 pixel representation of the binary image. The box scan layers are next included in Figure 23. The tree graph constructed with the box scanning is shown in Figure 24. The selection of end points for image representation and correlation is shown in the tree graph in Figure 25. These end points are also shown on the image outline in Figure 26 where it can be seen that they provide representative indicators of the extremities of the Space Telescope image.

Algorithms were developed for the selection of these end points on the tree graph. First, the number of end points was selected as a function of the number of pixels in the scene. Based on general experience this was made equal to 1/2 the number of pixels on a side. In other words, an 8 x 8 pixel scene would use four end points whereas a 32 x 32 pixel scene would have 16 end points. The determination of the initial end point/points was that having the longest branch/branches. Selection proceeded laterally to shorter branches until 16 were selected for the case of the 32 x 32 pixel representation of the Space Telescope. The x,y values of the tree graph elements were used for calculations to select all but the initial end points. A branch extremity was designated an end point if the following relation was met:

$$|x_{i-1} - x_i| + |y_{i-1} - y_i| \cong 4$$



MIRROR IMAGES OF SHAPE

- VIEWS ① & ③
- VIEWS ② & ④
- VIEWS ⑤ & ⑥
- VIEWS ⑦ & ⑬
- VIEWS ⑧ & ⑭
- VIEWS ⑨ & ⑪
- VIEWS ⑩ & ⑫

6 NORMAL VIEWS: (3 UNIQUE VIEWS OF SHAPE)  
 8 CORNER VIEWS: (4 UNIQUE VIEWS OF SHAPE)  
 TOTAL: 14 VIEWS → (7 UNIQUE VIEWS OF SHAPE)

Figure 19. Typical reference views of target vehicle.

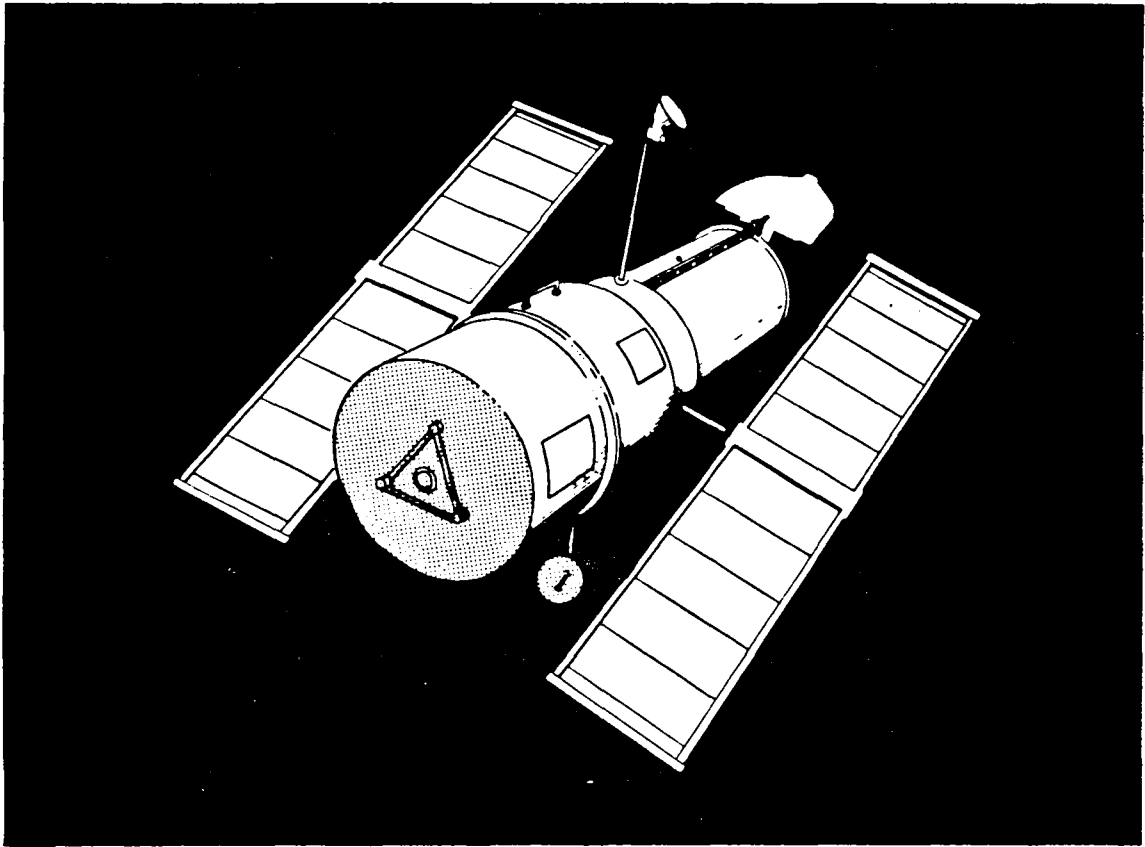


Figure 20. Space Telescope image.

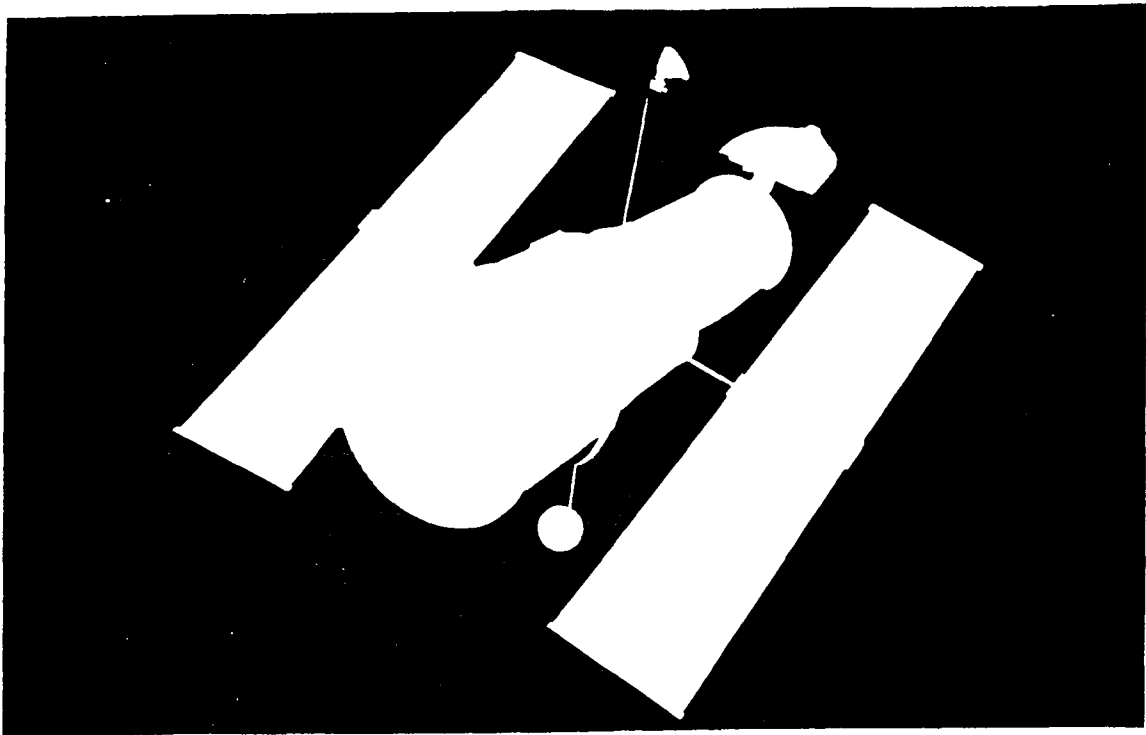


Figure 21. Binary image of Space Telescope.

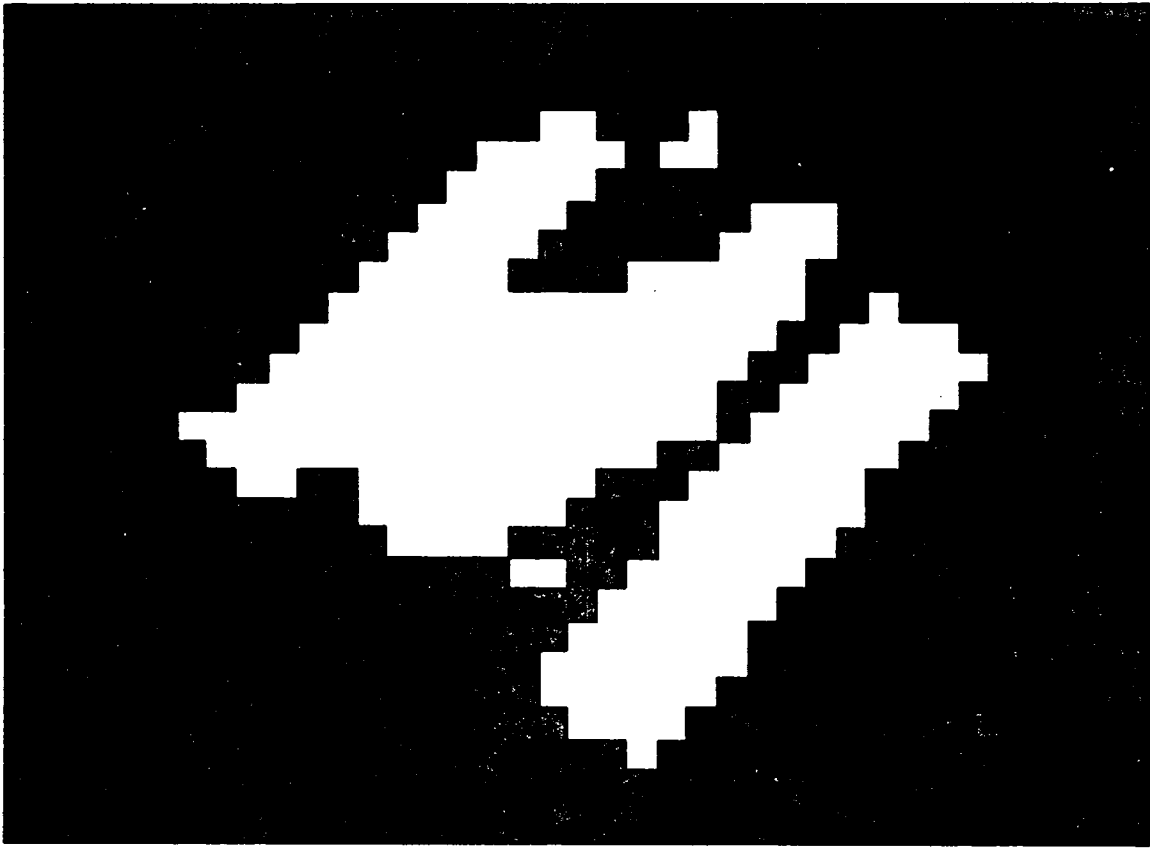


Figure 22. 32 x 32 pixel representation of the Space Telescope.

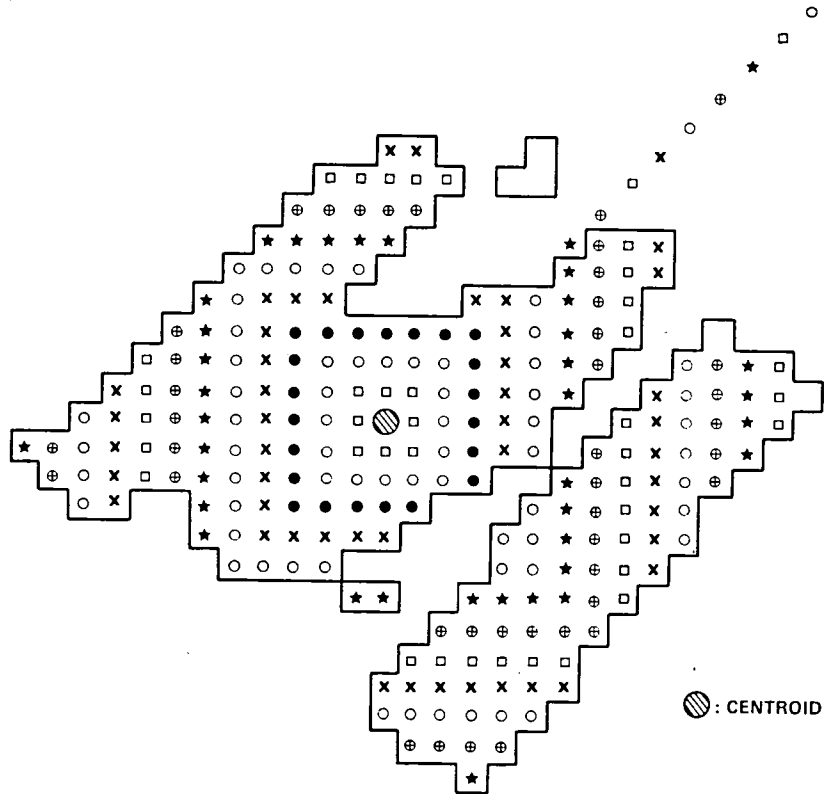


Figure 23. Sequential box scan.

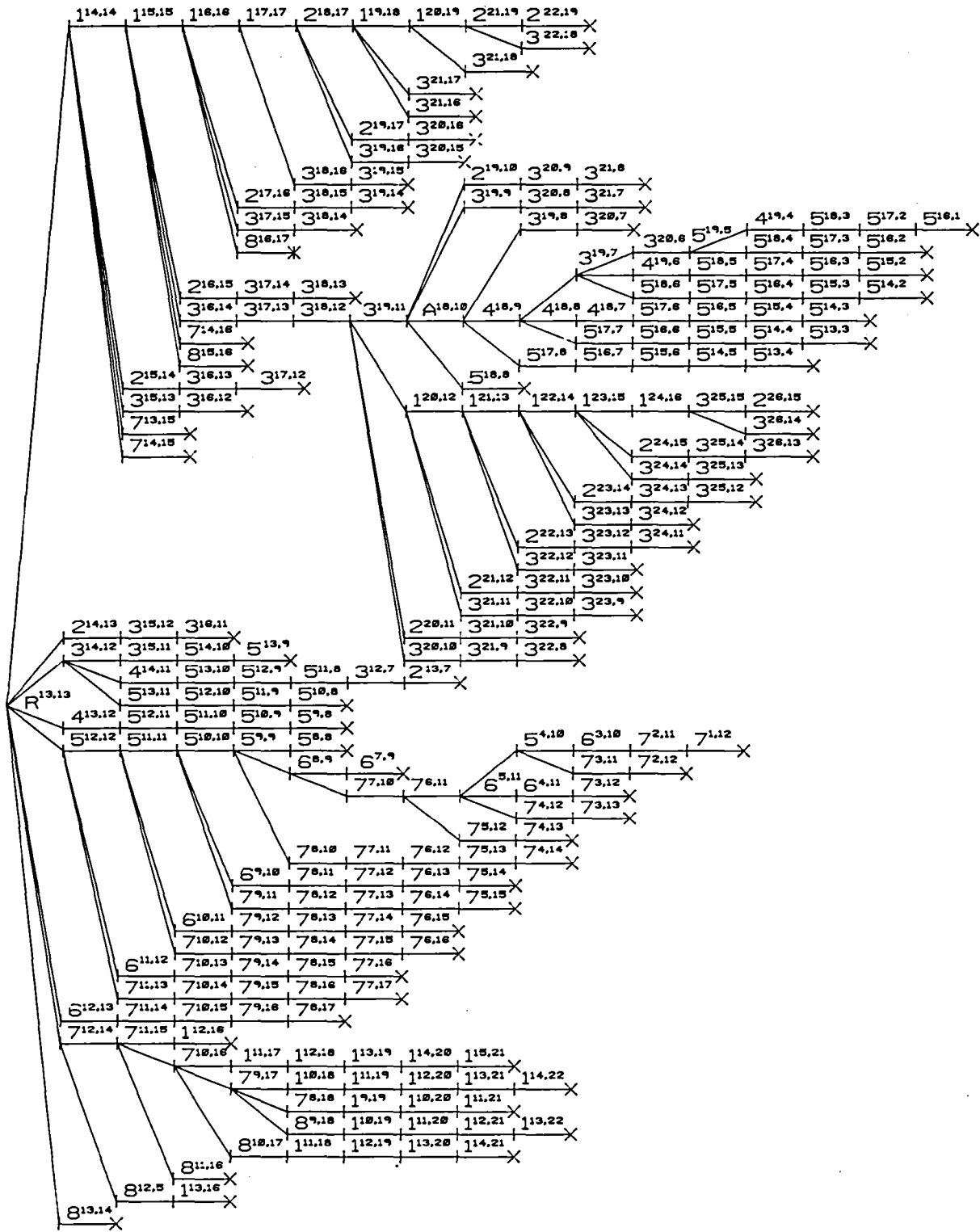


Figure 24. Tree graph of Space Telescope.



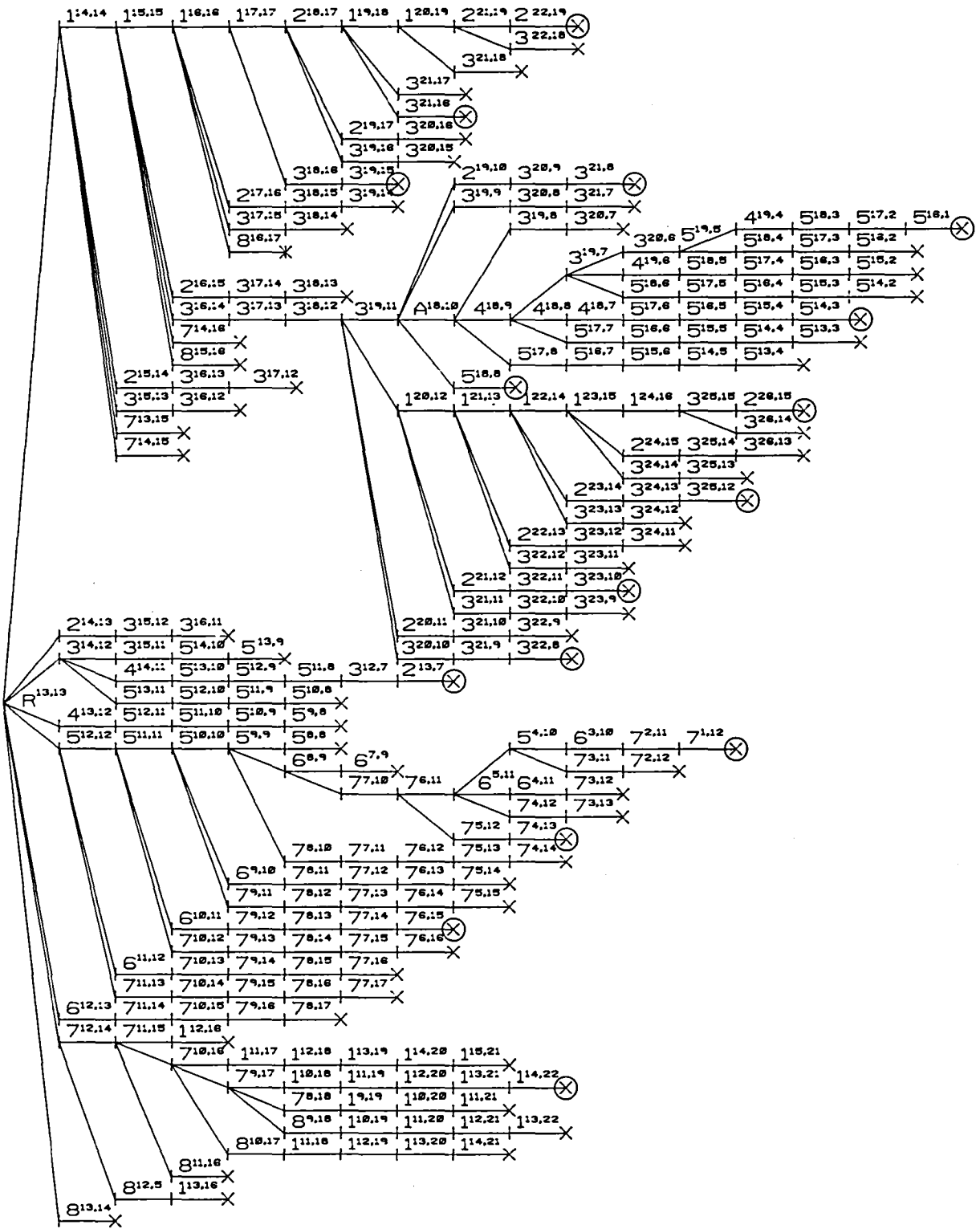


Figure 25. Tree graph end point selection.

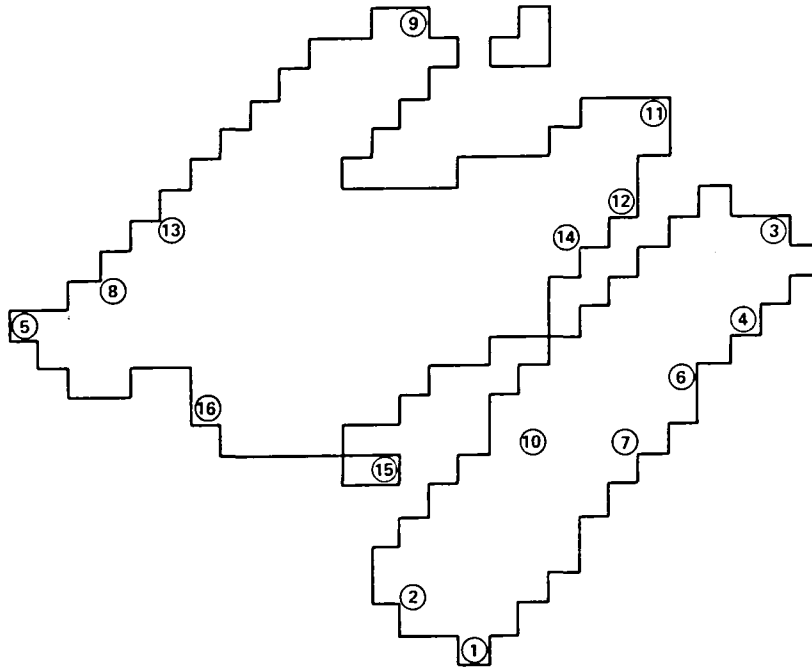


Figure 26. Tree graph end points.

In this manner the following end points were designated as indicators:

- 1 — 5<sup>16,1</sup> (Initial Selection)
- 2 — 5<sup>14,3</sup>
- 3 — 2<sup>26,15</sup>
- 4 — 3<sup>25,12</sup>
- 5 — 7<sup>1,12</sup>
- 6 — 3<sup>23,10</sup>
- 7 — 3<sup>21,8</sup>
- 8 — 7<sup>4,13</sup>
- 9 — 1<sup>14,22</sup>
- 10 — 5<sup>18,8</sup>
- 11 — 2<sup>22,19</sup>
- 12 — 3<sup>21,16</sup>
- 13 — 7<sup>6,15</sup>
- 14 — 3<sup>19,15</sup>
- 15 — 2<sup>13,7</sup>
- 16 — 6<sup>7,9</sup>

## SCENARIO OF AUTONOMOUS DOCKING MISSION

Implementation of the "AI BOSS" machine vision technique on the OMV has the potential of providing a viable autonomous docking capability. This proposed system will be represented on the MSFC orbital docking facility, shown previously in Figure 4, and evaluated for various initial conditions of target vehicle orientation. The minimum equipment required to provide this OMV enhancement appears to consist of a computer compatible TV camera having 256 x 256 pixels, computer memory of 64 K for storage of a single video frame in real time (either 1/60 or 1/30 sec), additional computer memory for image processing, computer processing speed fast enough to complete the "AI BOSS" computations in less than 1 sec, and associated control laws for docking such as a profile of range versus range rate. All other requirements are OMV baseline requirements and are not unique for autonomous docking.

A scenario for autonomous docking by means of "AI BOSS" is depicted in Figure 27 and includes the following sequential procedures:

- 1) Information on the orbits of the target and chaser vehicles will allow an automatic approach to less than one mile.
- 2) As the target vehicle is visually acquired on the TV camera field of view, the chaser vehicle controls its pitch and yaw body rotations so as to keep the target centered.
- 3) The approach to the target continues until the target image fills up an appreciable part of the TV field of view. Pitch and yaw commands must continue to maintain the target in the center of the view. Calculations of the image centroid by real time storage of video frames and at a speed of at least once per second will be required.
- 4) The chaser will maintain a station keeping distance from the target by analysis of the video data. A border of perhaps 10 to 20 pixel widths on the outer portions of the video frame will be used to maintain the station keeping distance. The extremities of the target image must be maintained within this border yet they will not be allowed to shrink more than an additional 20 pixel widths inside the border.
- 5) The chaser computer memory will have reference tree graphs of the target vehicle as viewed from representative directions. Figure 19 shows typical reference views and indicates that seven may be sufficient for any orientation of the target. The "AI BOSS" scan technique also must be employed for generating the reference tree graphs.
- 6) When the station keeping distance is obtained, the tree graphs will be obtained for the target in question. By comparison with the length ratios of the major tree branches, the system will determine which reference view it is most likely near.
- 7) A chaser vehicle roll search will be made to provide a better match to the selected reference tree graph. The search must also command chaser vehicle horizontal and vertical motions also in order to improve the performance measurement as previously shown in Tables 2 and 3. While this search is in progress the station keeping distance as well as pitch and yaw centering must be maintained.

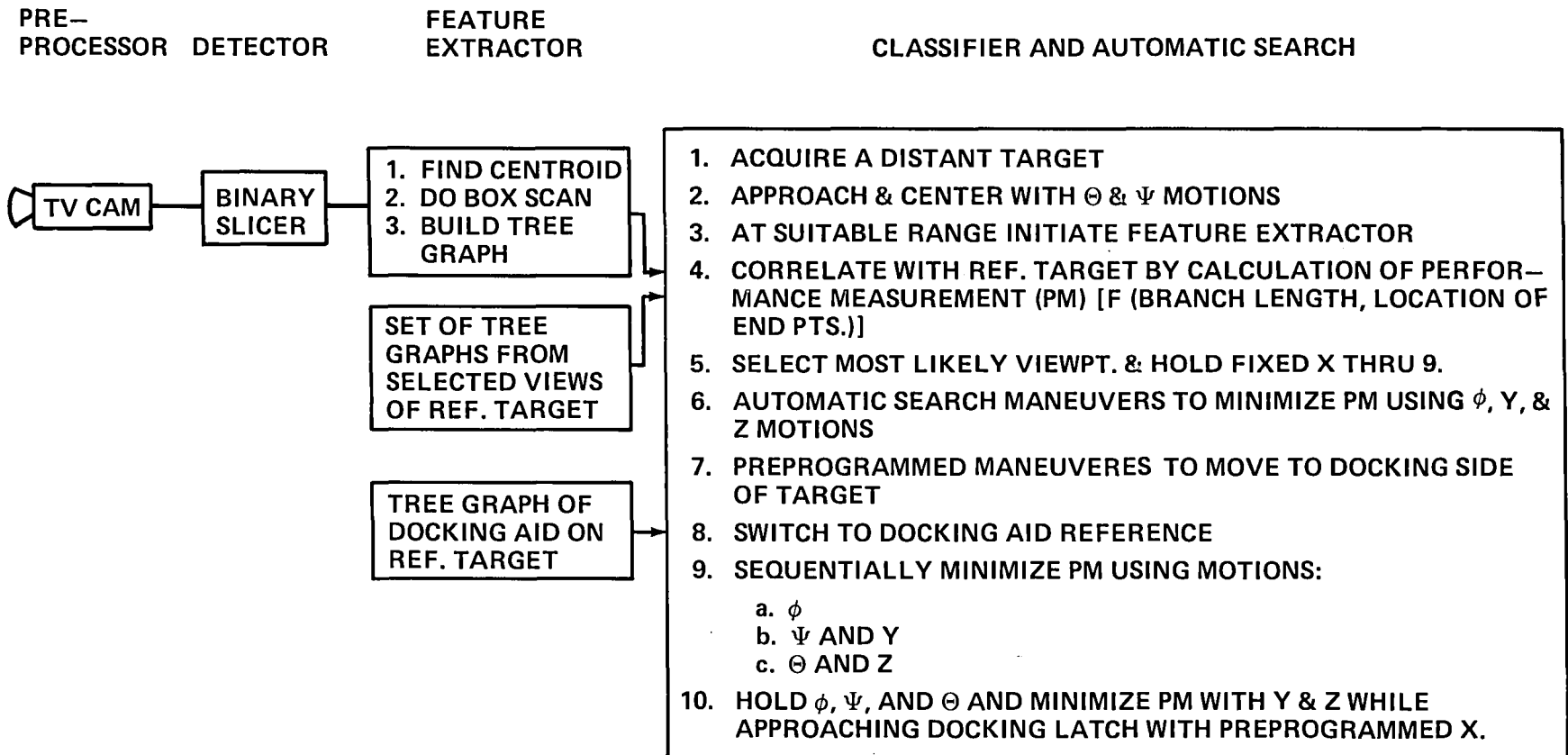


Figure 27. Implementation of "AI BOSS" for autonomous docking.

8) The search must continue until a very good correlation with one of the reference tree graphs results.

9) By recognizing which view has been matched, the chaser can then implement a stored maneuver to allow it to become aligned with the docking mechanism side of the target vehicle. The station keeping distance must be maintained until completion of this step. In case the target vehicle is not attitude stabilized, the chaser must command itself so as to track these uncontrolled motions in order to maintain this reference position.

10) The reference image data is then switched to a different format corresponding to image information on the docking mechanism for docking alignment device. The approach to a docked position is then accomplished with a pre-programmed closure maneuver by again using relative orientation to the chaser vehicle, range control, and utilizing "AI BOSS" image analysis of the docking mechanism features.

### CONCLUSION

The application of syntactic pattern matching for autonomous operation of an orbital servicing vehicle has the potential to dramatically reduce onboard computer requirements. Machine vision for automatic orbital docking is a complex and computation-intensive task using classical signal processing techniques such as 2D FFT's. The "AI BOSS" technique described embodies powerful heuristic pattern recognition capability by identifying such image shapes as elongation, compactness, symmetry, relative lengths of appendages, etc. Yet "AI BOSS" requires much less computer memory and computation than classical signal processing techniques. Further investigation of this technique using an MSFC orbital docking simulator will expand and quantify the merits of this unique approach.

## REFERENCES

1. Cohen, Paul R. and Feigenbaum, Edward A.: The Handbook of Artificial Intelligence, Volume III. Heuristech Press, Stanford, California, 1982.
2. Paupoulis, Anthanasios: Signal Analysis. McGraw-Hill Book Company, New York, 1980.
3. Girffiths, J. W. R., Stocklin, P. L., and Van Schooneveld, C.: Signal Processing. Academic Press, London, 1973.
4. Gonzalez, Rafael E. and Thomason, Michael G.: Syntactic Pattern Recognition. Addison-Wesley, Reading, Massachusetts, 1978.

## APPENDIX

### DESIGN DESCRIPTION OF REAL TIME VIDEO INTERFACE

The basic function of the interface is to transfer data from a video analog to digital (A/D) converter to the FPS-5205 array processor. The video converter works real time on 256 x 256 pixels at 8-bit resolution. This translates to a word rate in excess of 6 MHz, considering that the camera runs at 60 frames per second. The maximum asynchronous data input rate for the FPS-5205 is 3 MHz, hence the interface problem.

This problem is overcome by the fact that the FPS-5205 has 38 data input/output lines to the GPIOP, plus various control lines required for timing input/output. Therefore, the FPS-5205 can input four 8-bit words in parallel in this instance yielding an effective data input rate (maximum) of 12 MHz. For further details on data input to the FPS-5205, consult the FPS-5205 GPIOP Reference Manual D.4, 860-7437-003A.

The operation of the interface is as follows: the video converter generates the 8-bit representation of the analog video signal at better than 6 MHz. As each conversion is finished, it is clocked into one of eight 8-bit data latches. (Recall that the data is input to the FPS-5205 at 4 words per cycle.) When four latches have valid updated data, the data is clocked into the output buffers and the timing of the control (hand-shaking) lines is begun. While the data is being output to (input by) the FPS-5205, four data words are clocked into the remaining four data latches. By the time this is done, the FPS-5205 has input the data in the output buffers, thus freeing the original four data latches discussed. The output cycle now repeats with the data from the latter four data latches being clocked into the output buffers and handshake timing is started again. Four new words are now clocked into the original four data latches, and so on.

On a timing diagram, the control signals OE (0-3) and OE (4 - 7) are output enables for the two sets of four data latches. They are asserted low, enabling the data outputs to be buffered onto the output lines. The control line OE (DRIVERS) is also asserted low, driving the outputs of the data buffers to the state of the inputs. IBLOAD is the data strobe to notify the FPS-5205 that data is waiting to be read, and it is also asserted low. IBLOAD ENABLE insures that valid data is ready before allowing IBLOAD to be asserted. This signal only influences startup on a new scan line. C0 through C7 are clock signals which direct the digital data output of the video converter to one of the eight data latches. Q0 through Q2 are three outputs of a recycling binary counter used to sequence C0 through C7. DATA READY serves as the timing signal for the counter and the decoder, Q0 - Q2 clocked on the rising edge of DATA READY and C0 - C7 clocked on the falling edge of DATA READY. DATA READY is a timing strobe generated by the video converter. S1 is a 157 nanosecond clock generated by the Hamamatsu camera controller, and HUNBL and HSYNC are input from the controller to the interface, being generated by the video sync generator. HSYNC is used to reset several control signals. FPSINP is a control line from the FPS-5205 used to enable conversions during horizontal retrace so that all valid data may be read.

To understand the need for the FPSINP control input to the interface, one must understand how a video frame is generated. In our case, the screen consists of 256 lines (non-interfaced). The screen is completely generated 60 times per second. For each of the 256 lines we are using 256 pixels, giving a screen

dimension of 256 x 256. The horizontal scanning period (the time it takes to write one line) is 60.5 microseconds. One-third of this time is used for beam retrace, getting the beam in position to write the next line. This leaves 40.33 microseconds as the active horizontal period. (HUNBL is asserted [low] during the beam retrace.) For 256 pixels per line, this means one pixel is written every 157.54 nanoseconds. Hence the cycle time of the clock signal X1.

Since HUNBL is asserted during horizontal retrace, FSPINP must go high, e.g., enable more conversions, so that the last four conversions previously performed and stored in data latches may be clocked out to the FPS-5205. Recall that four data words are being output while four new data words are being converted and stored.

HSYNC will reset IBLOAD, IBLOAD ENABLE, and the binary counter (lines Q0-Q2). When HUNBL is raised high (the scan is active), sequencer line C6 is asserted. On the first falling edge of DATA READY, a byte of invalid data will be stored in data latch 7. (This is because the video converter outputs the previous conversion, and in this case it was performed during horizontal retrace.) On the next falling edge of DATA READY, the data word representing the first valid conversion will be clocked into data latch 1. The conversion and output sequence previously described is now in motion.



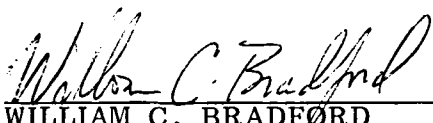
APPROVAL

COMPUTER VISION FOR REAL-TIME ORBITAL OPERATIONS

Center Directors Discretionary Fund Final Report (Project No. 82-27)

By Frank L. Vinz, Linda L. Brewster, and L. Dale Thomas

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.



---

WILLIAM C. BRADFORD

Director, Information and Electronic  
Systems Laboratory

**End of Document**