

AD-A103 378

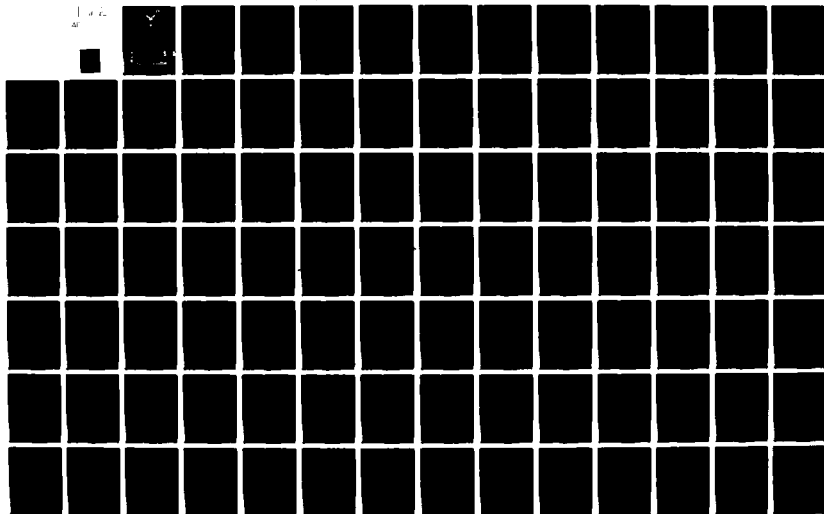
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/G 9/2
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS.(U)

DEC 80 M A SMART

UNCLASSIFIED

AFIT/GCS/EE/80-14

NL



AD A103378



LEVEL

BS

002
①



This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
ELECTE
AUG 27 1961

S D A

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

81 8 27 007

DTIC FILE COPY

AFIT/GCS/EE/80-14

AN EVALUATION OF DETAIL
IN
DYNAMIC VISUAL DISPLAYS.

THESIS

AFIT/GCS/EE/80-14

Mary A. Smart
2Lt USAF

DTIC
ELECTE
AUG 27 1981

A

Approved for public; distribution unlimited.

AN EVALUATION OF DETAIL
IN
DYNAMIC VISUAL DISPLAYS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Mary A. Smart, B.A.

2Lt

USAF

Graduate Electrical Engineering

December 1980

Approved for public release; distribution unlimited.

PREFACE

The Information, Presentation, and Controls Group (AAAT-1) of the Air Force Avionics Laboratory is interested in determining the human engineering problems that may be encountered by those using a display with a constantly changing background such as the Airborne Electronic Terrain Mapping System (AETMS). This research was conducted to enhance the AETMS by adding a symbol producing software overlay and to develop the basis for a symbol set for the AETMS.

I would like to extend my thanks to Dr. Matthew Kabrisky, my thesis advisor, Lt. David Rall for his software consultation, and Bruce for his understanding and support.

NAME	DATE
PROJECT	NO.
DESCRIPTION	
APPROVED	
SPECIAL	
A	

CONTENTS

	Page
Preface.....	ii
List of Figures.....	iv
List of Tables.....	v
Abstract.....	vi
I. Introduction.....	1
II. Software.....	4
III. Literature Review.....	19
IV. Symbol and Color Set.....	25
V. Methodology.....	28
VI. Conclusions.....	31
VII. Recommendations.....	32
Bibliography.....	33
Appendix A: Software Flowchart.....	34
Appendix B: Variable and Table Definitions.....	40
Appendix C: Equations.....	43
Appendix D: Symbol Set.....	44
Appendix E: Experiment Instructions and Data Forms....	45
Appendix F: User's Guide to Thesis Software.....	49
Appendix G: Software Listing.....	51
Vita.....	102

List of Figures

Figure		Page
1	Data Base Modeling Technique.....	5
2	AETMS Configuration.....	6
3	Hierarchy of SYMBOL with its sub-modules.....	8
4	Relationship of Common areas with SYMBOL.....	9
5	Rotated Display.....	9
6	CHECK2 Display.....	9
7	Rotated Display with Parameters.....	9
8	Flow Chart.....	35
9	Flow Chart.....	36
10	Flow Chart.....	37
11	Flow Chart.....	38
12	Flow Chart.....	39
13	Symbols.....	45

List of Tables

Table	Page
I. Arguments passed from PRSPCT to SYMBOL.....	11
II. Symbol Set.....	26
III. Color Set.....	27
IV. PRSPCT Variables.....	40
V. SYMBOL Variables.....	41
VI. SYMBOL Tables.....	41
VII. Common Areas.....	42

↓

ABSTRACT

This report has its basis in the Airborne Electronic Terrain Mapping System (AETMS), an aircraft mapping system based on an idea proposed by L.A. Tamburino and developed by the Air Force Avionics Laboratory.

A constantly changing background provides the basis for an interesting investigation on the hypothesis that a dynamic background will distract the operator's attention away from important details on the display.

The software developed provides an overlay of symbols onto the terrain map generated by the AETMS and is contained in this report. A symbol and color set is suggested for the AETMS.

An experiment to determine a master symbol and color set is suggested. The experiment suggests using both simple vector symbols and Fourier transformed symbols to help define the master set.

✓

An Evaluation of Detail in Dynamic Visual Displays

I. INTRODUCTION

This thesis has two main objectives. One objective is to provide a software package that will overlay symbols onto a terrain map generated by the Airborne Electronic Terrain Mapping System (AETMS). This system was developed by the Information, Presentation, and Controls Group (AAAT-1) of the Air Force Avionics Laboratory, based on an idea proposed by L.A. Tamburino (Ref 16). The software package will provide a video image processor the ability to display symbols as well as terrain. The second objective is to experiment with a small set of symbols to find shapes and colors that are easy to identify.

The hardware basis for this thesis is the AETMS. This system is designed to aid pilots in low altitude flying. "The AETMS is more than a horizonital or vertical situation map system. It is an integrated information system that will supplement current aircraft systems, giving the pilot the capability to negotiate low level, high speed profiles" (Ref 4:10). Designers have constructed software and hardware that converts a digitized representation of terrain into visual display. The software developed for this thesis provides an overlay of a symbol onto the terrain at the appropriate latitude and longitude.

There have been many studies of symbols and displays in the past. These studies were mainly concerned with displays

that had non-dynamic or static backgrounds, i.e., backgrounds that do not change. The AETMS display changes as does the vehicle's position. A constantly changing display presents some interesting questions:

1. Will the dynamic background distract the operator's attention away from important details, such as an approaching target?

2. If distraction may occur, then what is the best attention getting shape and/or color for a symbol?

3. What background color should be combined with a symbol's shape and color? These questions are the theoretical basis for the thesis.

Due to system limitations, it was impossible to integrate the thesis software with the AETMS. The reasons for this were twofold. First, the system allows only 28K of memory to be allocated to a task. The thesis software alone took over 28K and finally had to be tasked in three separate packages so that it could be tested by itself. Presently, the thesis software can generate symbols on a plain background. The thesis can be re-written to be overlaid onto itself. This method will allow the thesis software to be tasked with the AETMS. Secondly, the AETMS is a poorly documented system. The system exits upon encountering an error. Explanations as to why an exit occurred are not generated by the system. This makes trouble shooting rather difficult. This system is now being updated to include error statements.

These limitations made it impossible to complete the experiments associated with determining optimum symbol shape and color. Therefore, an explanation of the experiment is included, along with an analysis of symbols and color, a discussion of software, and a literature review.

II. SOFTWARE

The Airborne Electronic Terrain Mapping System (AETMS) can be simply described in the following manner: "The Digital recordings of the Defense Mapping Agency (DMA) world wide data base are preprocessed as depicted to obtain polynomial fits for the terrain elevations (Figure 1). These polynomials are stored as sequences of coefficients (i.e., compressed form). The on-board mass memory of the AETMS will be loaded from a world wide base. The aircraft navigation system would provide present position inputs to the on-board computer which uses them to access memory. The retrieve coefficients would be used to generate a terrain relief display for the pilot/navigator" (Ref 10:8).

The AETMS is currently being tested on a PDP 11/45. The information generated by the AETMS is displayed on a Ramtek display system via a device driver written by SRL, Inc. The original AETMS was written by the Systran Corporation of Dayton, Ohio. The AAAT-1 group is presently re-writting the AETMS software so that is easier to use and more responsive to the user. Figure 2 shows the system configuration.

The AETMS consists of two software packages, PRSPCT and CNTOUR. PRSPCT provides the user with a direct heads up view of the terrain as seen from the cockpit. CNTOUR provides a display much like that seen on a topographical map. Both PRSPCT and CNTOUR are interactive systems,

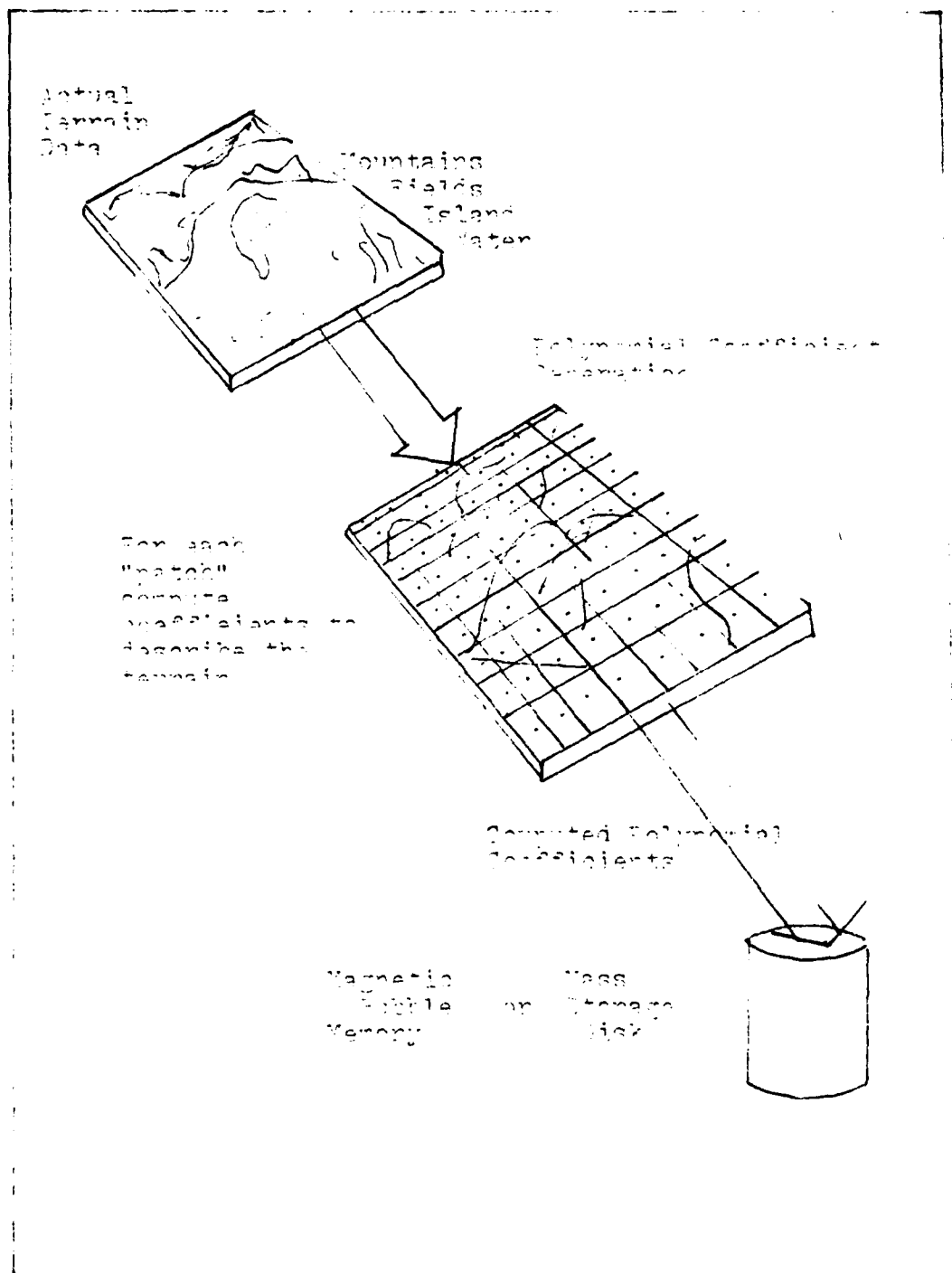


Fig. 1 Data Base Modeling Technique (540 1115)

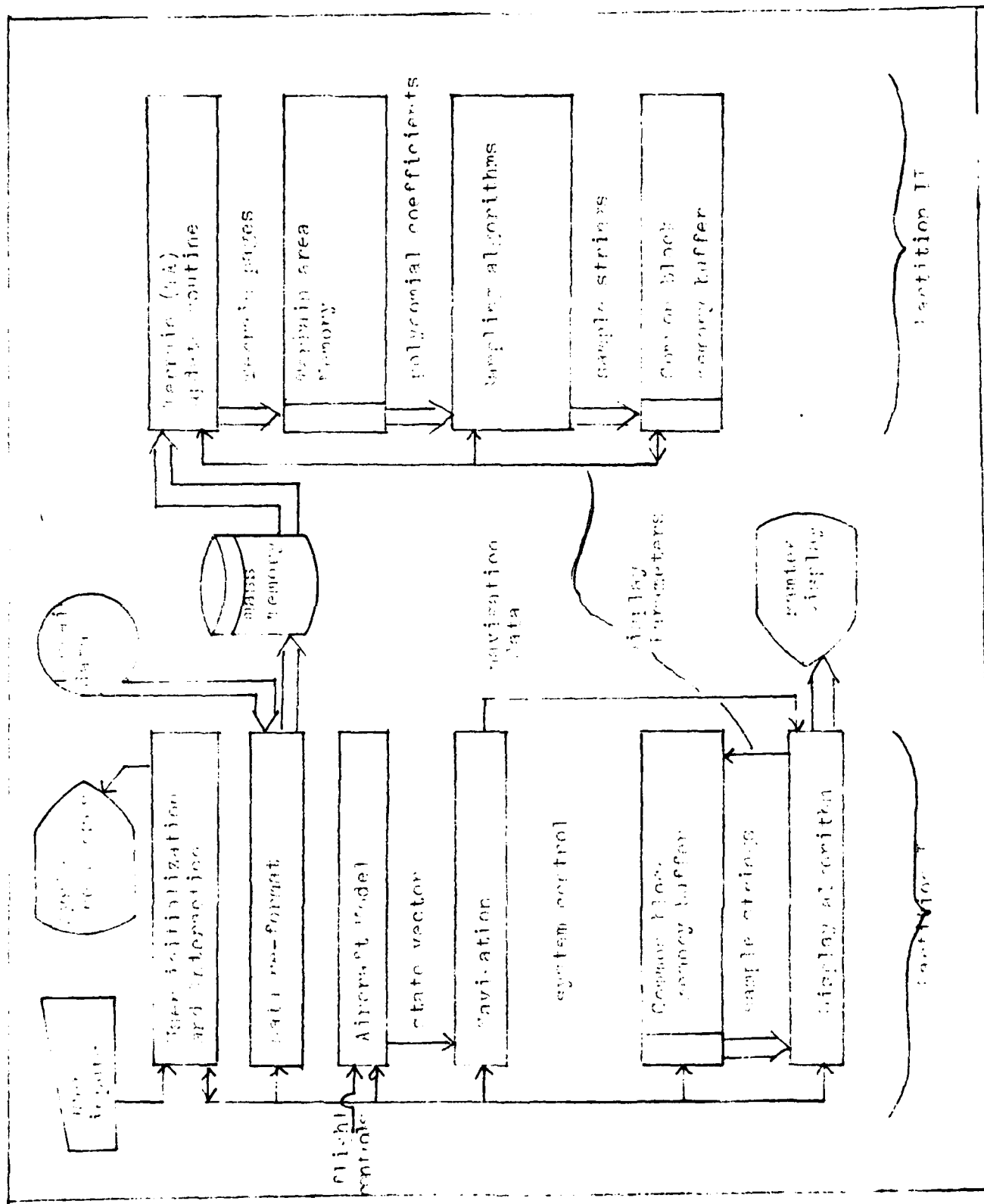


Fig. 2. Aircraft navigation (2 of 12)

accepting input from a joystick. The interested reader is referred to Systran documentation for a more detailed discussion of PRSPCT and CNTOUR (Ref 15).

AETMS designers use the following two terms in their definition of the AETMS structure. They are referenced in this description of software and so their definitions are included here. A patch is defined as follows: "Terrain is divided into a number of patches. A patch is a small area of terrain whose altitude at each point can be determined from a polynomial. The polynomial is:

$$Z = C1 + C2X + (C3 + C4X)Y$$

where X,Y are the coefficients of the point and C1, C2, C3, C4 are four coefficients that vary from patch to patch, and Z is the altitude of the terrain at X,Y (Ref 4:15).

Terrain data (coefficients) from the DMA data base is stored on a disk. These concepts are described in detail by Tamburino (Ref 16).

Another important concept is that of a BAU. BAU is an acronym for Binary Angular Unit and equals $\pi/128$. This is a constant used to convert azimuth into smaller units so that it is easier to access the data base.

The thesis software consists of modules that determine window bounds, check data against the bounds, and draw symbols over the terrain generated by the AETMS. The entire thesis module is titled SYMBOL. The PRSPCT executive routine call SYMBOL. Figure 3 gives a pictorial view of the

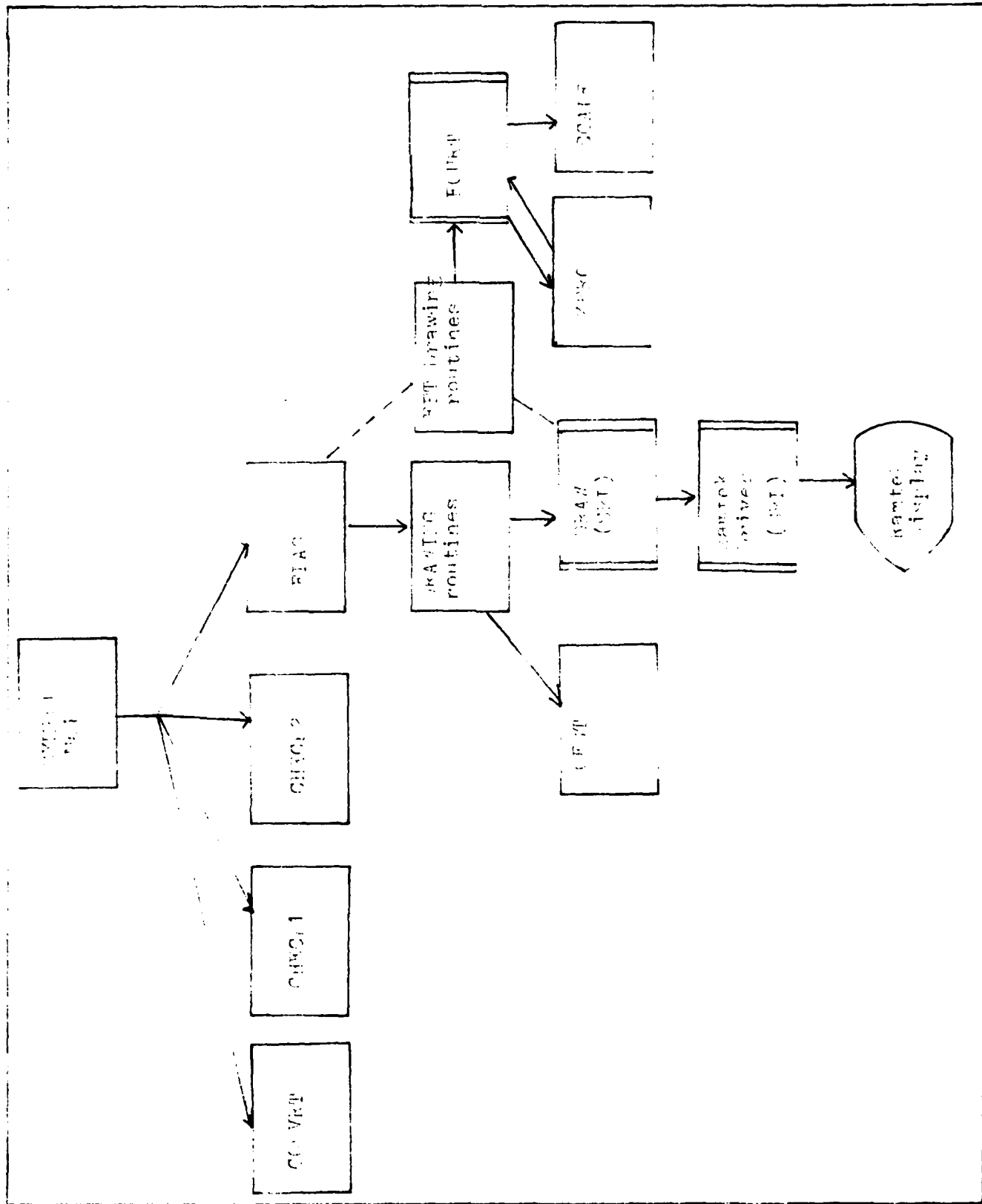


Fig. 3. Hierarchy of SWTC (with its sub-modules)

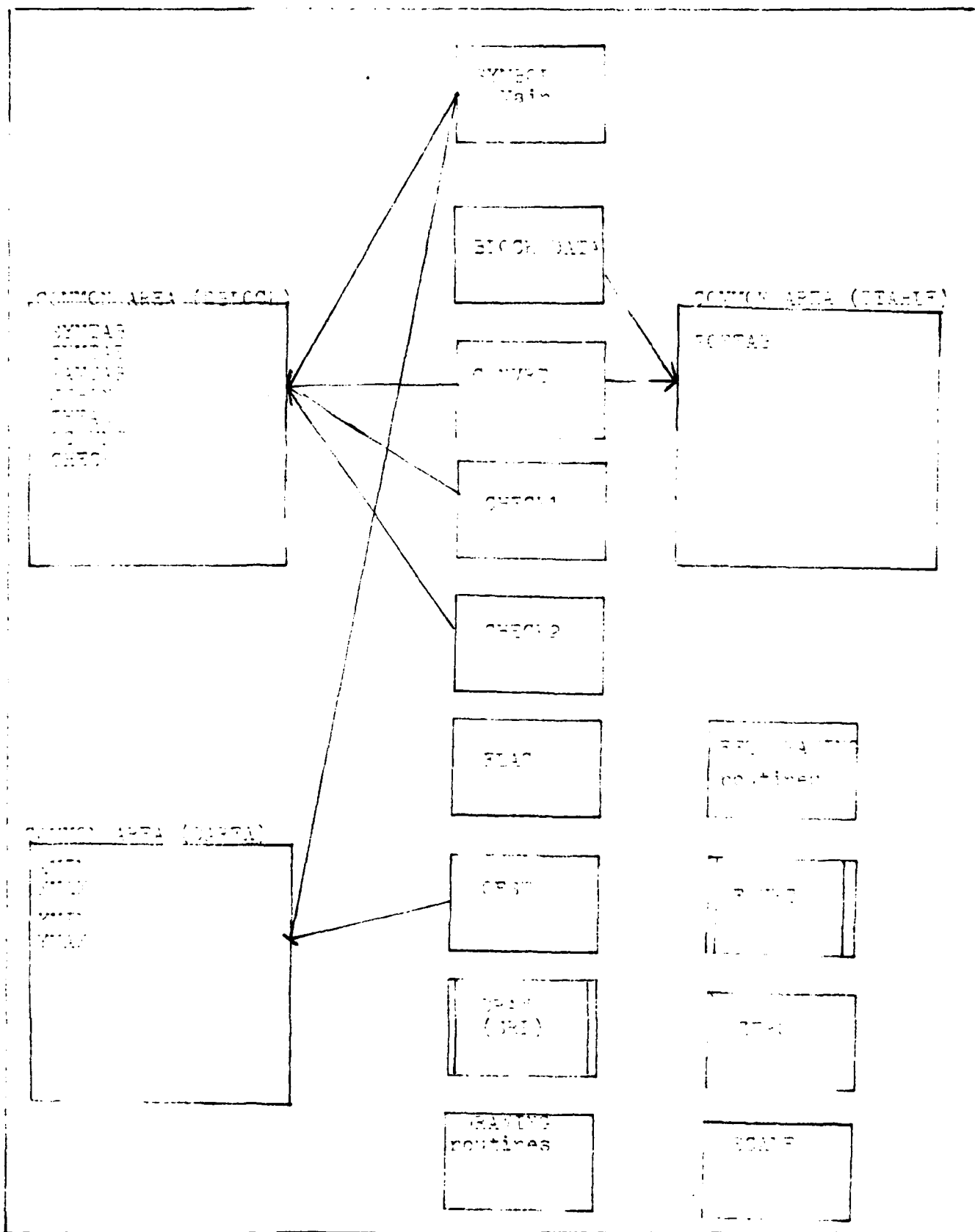


Fig. 3. Relationship of Common with subroutines.

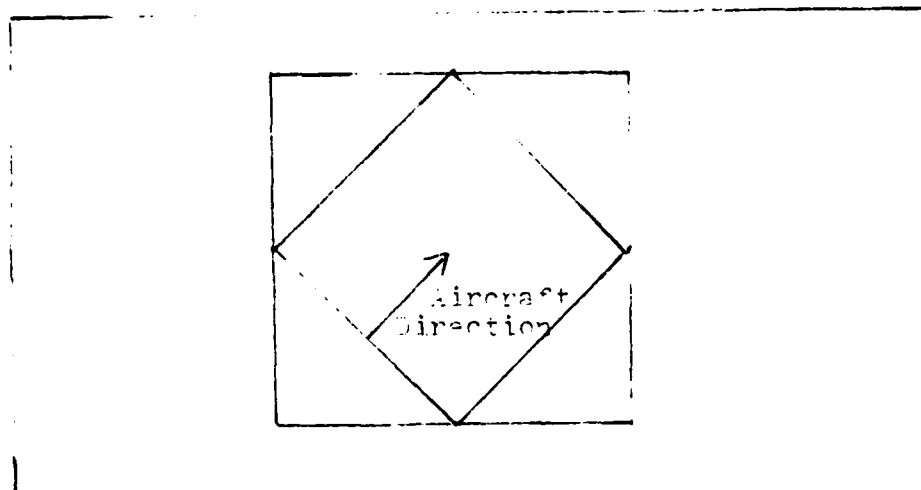


Fig. 5. Rotated display

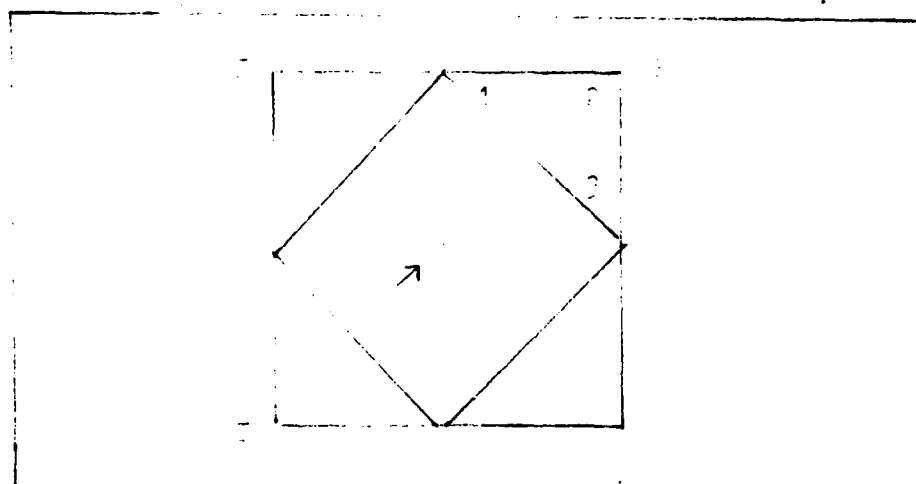


Fig. 6. CHEC-2 Display

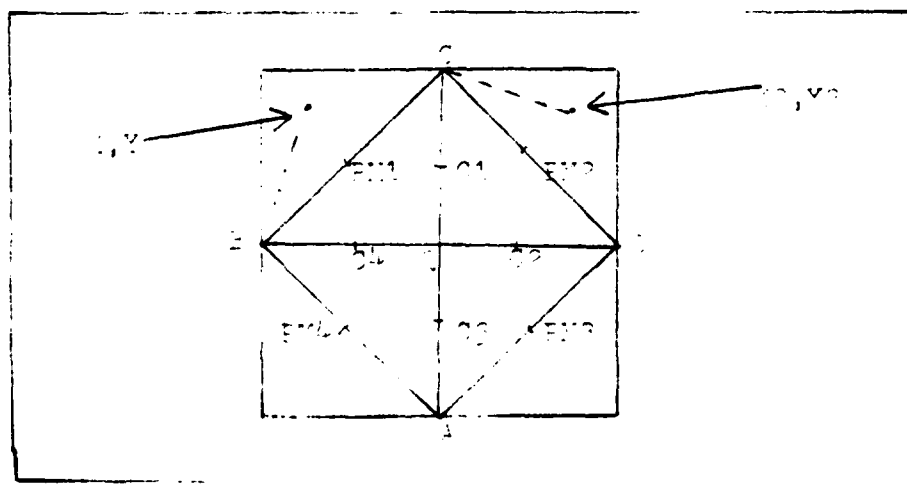


Fig. 7. Rotated display with parameters

hierarchy of the modules in SYMBOL. Figure 4 gives a representation of the relationship between sub-modules in SYMBOL and the defined common areas. Appendix A contains a flowchart for the module, and a listing of SYMBOL and test results are presented in Appendix G.

SYMBOL has five main functions:

1. Determine the values of the corners of the window, aircraft heading, and conversion factors.
2. Call CONVRT each time the aircraft's position changes.
3. Call CHECK1 with the bounds found (#1 above), if and only if aircraft heading is North (0 or 360), South (180), East (270), or West (90).
4. Call CHECK2 if and only if aircraft heading is anything other than North, South, East, or West.
5. Call FLAG if and only if there are symbols within the bounds of the window.

PRSPCT passes 11 arguments to SYMBOL. Table I presents the arguments and a short definition for each.

SYMBOL uses these arguments to determine the corner values (A,B,C,D), conversion factors (POSY,POSX), and aircraft heading (IHEAD). Appendix C contains the equations for generating the needed information. Appendix B contains the variable and table definitions for SYMBOL and PRSPCT. SYMBOL returns nothing to PRSPCT.

Subroutine CONVRT is the first subroutine called by the main body of SYMBOL. POSX and POSY, the conversion factors,

TABLE I

ARGUMENT	DEFINITION
IAY	Latitude (degrees)
FAY	Latitude (minutes)
IAX	Longitude (degrees)
FAX	Longitude (minutes)
DX/DY	Patch's angular width/length (degrees)
NX/NY	Number of Patches in memory
A	Aircraft azimuth in BAU's
D	Aircraft speed (knots)
DA	Aircraft change in azimuth in BAU's

are passed to CONVRT from SYMBOL. CONVRT uses BLOCK DATA via the common area, DTABLE. BLOCK DATA contains real world latitude and longitude in a table known as POSTAB. POSTAB also holds the flag of an object associated with a particular longitude and latitude. The dimension of POSTAB is (x,3), where the x indicates that POSTAB may contain as many symbols as may be encountered in a particular flight.

The main function of CONVRT is to convert POSTAB to display (patch) coordinates. This is accomplished by adding POSY to the latitude values, and POSX to the longitude values. The new values are then placed in SYMTAB, dimension (x,3), and returned to SYMBOL via the common area DBLOCK.

Subroutine CHECK1 is called if aircraft heading is

North (0 or 360), South (180), East (270), or West (90). The window coordinates are first sorted into minimum and maximum values and then passed to CHECK1. If the SYMTAB values fall within the bounds, then it is placed in RAMTAB. RAMTAB is later used in FLAG to tell the object drawing routines what and where to draw a symbol.

Subroutine CHECK2 is called if aircraft heading is anything other North, South, East, or West. If so, then it is known that the window is rotated as in figure 5. Checking for symbols is more complex in CHECK2 than in CHECK1. Figure 6 gives a pictorial description of the discussion to follow.

First an initial check is done against maximum and minimum values to see if the symbol is within the larger square, IJKL. Those values that are within these bounds are placed in a temporary table, TEMTAB. Next, the symbols that are within the triangles, (eg. 123), must be eliminated as they are not in the display. Midpoints and the slopes of the rotated display edges are used to do this checking. Figure 7 shows these points. Line slopes for lines AB, BC, CD, and AD are computed. The midpoints are necessary to determine which triangle the point is in. This is important since the next step involves a line slope comparison.

Suppose point x,y (figure 7) is a point that may or may not be in the display bounds. CHECK2 does the following:

1. Calculates which triangle x,y is in.
2. Calculates the slope of line Bx,y .

3. Slope of line BC has already been determined and is constant for the present aircraft heading.

4. Compares the slope of line Bx,y with that of line BC. If the slope of Bx,y is less than that of BC, then x,y is not transferred to RAMTAB. Consideration of another point, x2,y2, is the same except in line slope comparison. If the slope of Cx2,y2 is greater than that of line DC, then x2, 2 is not in the display. This is why determining which triangle the point is in is important.

Both CHECK1 and CHECK2 return a variable, LCOUNT, and a table, RAMTAB, to SYMBOL via the common area DBLOCK. LCOUNT is the number of symbols that are in RAMTAB and hence are within the display bounds. This is important in the execution of other subroutines.

Subroutine FLAG is the last routine called by SYMBOL. SYMBOL passes RAMTAB and LCOUNT to FLAG.

FLAG checks the value of the x,3 entry in RAMTAB. The value in x,3 determines which symbol must be drawn at a particular latitude and longitude.

Each object has a separate subroutine to output a representative of that object. FLAG does not return any values to SYMBOL. FLAG also sets a reset value, IFLAG, for use by the Ramtek display driver. Reset, or erasure, occurs only when a new table of RAMTAB values is passed to FLAG. After erasure, IFLAG is set to zero so that further erasures will be suppressed until FLAG is again called by SYMBOL.

The object drawing routines referred to above are

*Pages
13, 14, 15
Blank*

simple end point plotting routines. FLAG passes IFLAG, RAMTAB, and a RAMTAB position pointer, to these routines.

RAMTAB latitude and longitude must be scaled to fit onto the video screen. Therefore, it is necessary to first call subroutine OFST.

RAMTAB and J are passed to OFST. OFST establishes the symbols position on the screen by converting its position in the window to the same relative position on the screen. Screen size is 480 (x) by 640 (y), with the origin in the upper left hand corner. A re-scaled RAMTAB value is returned to the calling routine.

Final output of a symbol occurs when the object drawing routine calls subroutine DRAW. This subroutine is a slightly modified version of a vector drawing routine designed and implemented by Dale Rangeler of SRL, Inc.

DRAW utilizes the Ramtek graphic display driver to output vectors. Each drawing routine passes a 4X4 matrix and IFLAG to draw. DRAW simply writes a line between the beginning x and y coordinates and the ending x and y coordinates. The number of calls to DRAW by a particular object drawing routine is a function of the number of vectors that an object is composed of.

The program to output Fourier transforms of symbols is essentially the same except for the addition of three subroutines.

Subroutine FOURT is a program by Norman Brenner from the basic program by Charles Rader. It uses the Cooley-

Tukey Fast Fourier Transform and is written in USASI basic fortran (Ref 4). This subroutine is called by the drawing routines to determine the fast Fourier transform data points of the object in question.

Subroutine ZERO zeroes out data after the third harmonic of Fourier data. This is necessary to get a usable representation of the object from the data.

Subroutine SCALE reduces the data points to a size that is compatible to the Ramtek display dimensions.

Testing SYMBOL consists of checking values on the bounds and corners, as well as values in between the bounds. Appendix F contains a SYMBOL user guide. Appendix G contains a program listing and test values.

III. LITERATURE REVIEW

This chapter has a twofold purpose. The first objective is to present some ideas about contemporary map displays and information requirements, and to show that the AETMS is a versatile system that does infact, fulfill these information requirements.

The second objective is to put forth some ideas about research that has been done concerning symbols, their size and color, and to give a short synopsis of other problems that are associated with visual displays.

Human factors engineering evaluation has become an important pre-design consideration. This is necessary because a tool that is designed for human use reduces mistakes and frustration. In light of this, the AAAT-1 Group tasked the Aerospace Medical Research Laboratory (AMRL) to do an exhaustive search of literature on current map systems. AMRL found that there are six map systems in existence today. The following summary of map types describes each map and its major limitations, and is described in more detail in a report prepared for the AAAT-1 Group by AMRL (Ref 10).

The most primitive map is a handheld chart. It has the obvious disadvantage of being nearly impossible to use while flying at high speeds and low altitudes. Currently, no fighter system uses hand held charts as a map system.

Direct view maps are paper maps mounted between

rollers. The map moves as does the airplane. Movement is a result of computer or doppler input. The major disadvantage of this system is that it is difficult to input steering or heading changes into the system.

The projected map system uses a rear-project microfilmed transparency of the original map. It is limited because it is costly and cannot be changed for a specific flight.

A combined map/CRT is a system that displays a map on a CRT. Researchers have found that there are legibility problems with this system.

Electronic displays generate all information electronically. Currently, there are no systems with airborne capabilities for using an electronic system. Designers are hesitant to use this type of a system because it is easily detectable by enemy radar.

Terrain avoidance and terrain following (T/A, T/F) systems allow pilots the ability to hide their aircraft under conventional electronic detection systems by allowing them to hug the terrain. The AETMS is a system designed to be a T/F system. "The AETMS is more than just a horizontal or vertical situation map system. It is an integrated information system that will supplement current aircraft systems, giving the pilot the capability to negotiate low level, high speed profiles" (Ref 50:10).

Information parameters for future aircraft systems are demanding. They were described by Crawford (Ref 1) and are

listed below:

1. Altitude below 500 ft.
2. Speeds proportionate with pilot and aircraft capabilities.
3. Sensing equipment that is 'invisible', i.e., not seen by enemy radar.
4. Use of missiles which require accurate positioning.
5. Minimizing detection time by maximizing time 'under' radar.

The AETMS was designed to provide answers to all of these parameters. The pilot will be able to fly below 500 ft at his maximum speed. It is an internal system and hence, it does not leave a signature that can be read by enemy detection systems. It is a highly accurate system since it is an exact copy of digitized terrain data. The pilot can stay below radar until just before weapons delivery. This is because he will not need to manually search for targets. Target points will be displayed on the screen. The system has the advantage of providing a look ahead feature for the pilot. Look ahead gives the pilot time to plan at alternate routes or targets. It is obvious that this system can reduce pilot stress, and therefore, will help prevent errors from fatigue.

"The Joint Tactical Information Display System (JTIDS) is a digital, secure, jam-resistant, communication system for a real-time command and control of combat operations"

(Ref 1:2). JTIDS monitors have the responsibility for standardization of a symbol set for aircraft systems. This is necessary because contractors develop a set of symbols each time they bring a new aircraft system on-line. A Symbology Standardization Committee (SSC) was formed to develop the standards. The SSC made several recommendations, a few of which are important to this discussion. A more detailed explanation of the JTIDS, SSC, and its recommendations can be found in a report prepared by the SSC (Ref 2).

The important recommendations are:

1. Largest dimension of a symbol should not be greater than 17 minutes of visual angle.
2. The symbol height to width ratio should be 3:2 (Ref 2).

Past experiments have shown that 17 minutes of arc is optimum. The following equation is used to calculate the size of a character:

$$H=2DTAN(\theta/2*60)$$

D= viewing distance in inches from eye to surface

θ = minutes of visual angle (arc)

H= symbol height in inches (Ref 2)

A study by the McDonnell Aircraft Company (McAir) presents a definition of a test JTIDS symbol set in the F-15 system. They established the same basic principles on object size as did the SSC. In addition, they determined

that color is an excellent tool to use in preventing display clutter. Clutter is a result of trying to place too much information on a display. The operator becomes overloaded with information and therefore, may miss important points. McAir used the standard three color system: red indicates hostility or danger, green represents friendliness or safety, and yellow indicates an unknown situation (Ref 12).

Electronic display systems also present other complications in developing an optimum visual display. These problems include flash or flicker, excess brightness, ambient light, and the adaptation differences required for night vision.

Flash or flicker is a phenomenon typical of electronic displays. The image on the screen appears to flicker. This is especially noticeable in a dark environment. It increases fatigue and eye strain. Flicker can be minimized by reducing contrast and background brightness.

Excess brightness is another problem associated with electronic displays. Either a symbol or its background is too bright. Both of these possibilities increase the chance that other elements of the display will be hard to distinguish. This problem also causes fatigue and eye strain.

Ambient light is the light that surrounds the display. It changes some of the other parameters that must be considered in developing good displays. It may decrease contrast or foreground brightness depending on its

intensity. These changes may in turn increase the chance of eye strain or fatigue.

The adaptation requirement necessary for a change from day to night vision must be considered when designing a visual display system. It is more difficult to see at night. Looking out into the dark and then down into a lighted screen causes immediate adaptation problems. This is probably the hardest problem to solve or minimize, and can potentially be one of the most dangerous dilemmas that a pilot may face.

The preceding discussion was a summary of a few of the problems that others have observed and have tried to solve. This thesis does not consider any of these problems, but the experiment was designed with them in mind. A detailed description of these problems and their solutions can be found in a work by Ketchel and Jenney (Ref 8) and Soliday (Ref 14).

IV. SYMBOL AND COLOR SET

This chapter will give some insight as to why a set of symbols and colors were chosen for the experiment.

The object set for this experiment consists of eight symbols and their Fourier transforms (FFT). Appendix D contains a pictorial representation of the symbols as they are seen on the Ramtek display.

The objects in the symbol set represent cultural and linear objects, and come from three categories. These categories include obstructions, targets, and linear objects. Table II shows the categories and objects.

The obstruction set is mandatory since the AETMS is designed to be used in aircraft flying at low altitude and high speed. The pilot needs to be pre-warned of any upcoming obstructions in order to avoid them.

Landmarks and target categories are necessary for similar reasons. The pilot needs a look ahead capability in order to prepare for weapons delivery or a change in course.

Linear objects will be most difficult to manually identify, because of high speed and a perspective background. Hence, linear objects are included in the symbol set.

Kabrisky hypothesized that a two dimensional Fourier transform occurs on visual input, and that this data is used by the human visual system for processing (Ref 7). Therefore, the Fourier transforms are included in the set so

TABLE II

CATEGORY	OBJECT
Obstructions (Landmarks)	Bridge Tall Building Electrical Wires
Targets	Target Designator Pointer
Linear (Landmarks)	Landing Strip Highway

that it can be determined through testing whether a simple representation of a object or its FFT is easier to identify. Adding FFT's to this thesis is for testing purposes only, the final symbol set will not include FFT's.

Color is a tool that can help enhance the usefulness of any display. Color sparks curiosity. A color coded system can also help eliminate clutter. It is important to find the best combination of colors for the AETMS. Best can be described as any number of things, including appeal, functionality, and legibility.

Red, green, and yellow are traditional aircraft instrumentation colors. Red indicates hostility or danger, green shows friendliness or safety, and yellow symbolizes an unknown condition. Table III shows foreground and

TABLE III

<u>SYMBOL COLOR</u>	<u>BACKGROUND COLOR</u>	<u>SIGNIFICANCE</u>
Red	Grey	Hostile
Green	Grey	Friendly
Yellow	Grey	Unknown
Grey	Red	Hostile
Grey	Green	Friendly
Grey	Yellow	Unknown
White	Black	Friendly
Black	White	Hostile

background color combinations.

Determining a good combination of colors and symbols will help make the AETMS a versatile and functional aid to pilots.

V. METHODOLOGY

This chapter reviews how the experiment was to be conducted, and explains what video output is necessary for such an experiment. The AAAT-1 Group has the equipment necessary to make a video recording of output produced by the AETMS and SYMBOL. It is possible therefore, to produce a sequence of AETMS and symbol overlay outputs that can be used in an experiment.

The output should consist of three separate tasks: the learning task, a non-stress identification task, and a stress producing validation task.

Task 1, the learning task, is necessary because past results have shown that subjects involved in tests such as this, need to learn about the task before they can perform it accurately (Ref 3). If this segment is ignored, then the results may not be valid and conclusions should not be drawn. This phase consists of three events. First, the subject is shown what terrain output looks like. Secondly, the subject is shown simple symbols and their Fourier transforms on a plain background. Finally, the subject is presented the symbols overlaid onto the terrain.

Task 2, the non-stress producing identification, shows a simple sequence of simple objects, their Fourier transforms, and a sequence of colors to the subject. The subject views the symbol and indicates what he thinks it represents. The subject's identification is either

corrected or validated. The same interaction occurs in color identification.

Task 3, the stress producing validation segment, is done to validate that the symbols really do represent some identifiable object to the subject. Symbols are presented in a random order and the subject must identify it and indicate whether or not the object is a threat.

The experiment is conducted in the same order as the video product described above.

Phase 1 corresponds to Task 1. During this phase, the subject becomes oriented to what the terrain and symbols should look like. It is also during this phase that the subject and the one conducting the experiment develop a rapport. The subject must feel at ease during the experiment, and this phase allows this to happen.

Phase 2 is implemented by using Task 2. The subject and experimenter interact during this phase. Phase 2 events occur in the following order: First, a symbol appears on the screen; secondly the subject indicates what he thinks the symbol represents; finally, his answer is either corrected or validated. The same sequence of occurs for color identification.

Finally, Phase 3 is conducted using output from Task 3. Symbols are generated by the task. As soon as the subject notices the symbol, he must identify it, and indicate whether it is a threat, a weapons delivery point, or neither of these. This phase is used to validate the results of

previous phases.

The background data and instructions for the experiment are presented in Appendix E, along with the experiment questionnaire, experiment tally sheet, and post-experiment questionnaire.

VI. CONCLUSIONS

The AETMS will be a powerful tool once a few logistics problems are overcome. The database for the map is quite large and requires a great deal of memory. It has been suggested that bubble memory be used to solve the space and weight problems associated with other memory devices. In addition, bubble memory has a faster access time than other forms of memory.

Furthermore, additions of the symbol overlay and a threat overlay being developed by the University of Dayton, will make the entire package a versatile tool for aircraft systems.

VII. RECOMMENDATIONS

It is possible to task SYMBOL with PRSPCT and the data base by using the overlay task building system available on the PDP 11/45. The thesis software can be reduced to an overlay of itself. There are at least two ways to do this. One way is to overlay every subroutine onto the main body of SYMBOL. Another way to do it, is to break the software into logical groups, such as: SYMBOL (SYMBOL, CHECK1, CHECK2, CONVRT), FLAG (FLAG, DRAW, OFST, BLOCK DATA), and PICTURE (all of the symbol drawing routines). The Fourier transforms programs need to have one more module, FFT (FOURT,SCALE,ZERO). These groups can then be overlayed onto PRSPCT.

It becomes hard to make a large latitude and longitude table for symbols since memory is a problem. Therefore, a more compact table should be designed to alleviate the problem. One way to do this, is to divide BLOCK DATA into subsets that can be overlayed onto each other. There would be three sections of block data, one for a latitude table, a second for a longitude table, and a third for a flag table.

A subroutine needs to be added to check the aircraft's altitude. PRSPCT monitors altitude and so this parameter could be passed to SYMBOL. Symbol color can then be changed as the altitude increases or decreases.

The symbol set must be tested to determine which symbols should be included in a master set. More symbols

need to be added to the current table so that testing can be conducted to determine that master set.

Finally, the entire system should be documented so that those that do not have an intimate knowledge of the system can use it.

Bibliography

1. Carel, W.L., et al. "Design Criteria for Airborne Map Displays." Culver City, California: Hughes Aircraft Company, March 1974.
2. Cassella, J.R., et al. "Joint Tactical Information Distribution System (JTIDS), Standard Symbolology Phase I, Cockpit Display Presentation Formats and Flight Test Symbol Set." ASD-TR-77-31, Wright-Patterson Air Force Base, Ohio: Aeronautical Systems Division, June 1977.
3. Chapanis, Alphonse. Research Techniques in Human Engineering. Baltimore: Johns Hopkins Press, 1959.
4. Cooley, J.W. and J.W. Tukey. "An Algorithm for Machine calculation of Complex Fourier Series." *Mathematics of Computation*, 19: 297-301 (April 1965).
5. Crawford, N.W. "Low Level Attack of Armored Targets." Santa Monica, California: The Rand Corporation, August 1977.
6. Dreyfuss, Henry. Symbol Sourcebook. New York City: McGraw-Hill, 1972.
7. Kabrisky, Matthew. A Proposed Model for Visual Information Processing in the Human Brain. Urbana, Illinois: University of Illinois Press, 1966.
8. Ketchel, J.M. and L.L. Jenney. "Electronic and Optically Generated Aircraft Displays: A Study of Standardization Requirements." Washington, D.C.: Office of Naval Research, May 1968.
9. Kopala, C.J. "The Use of Color-Coded Symbols in a Highly Dense Situation Display." Wright-Patterson Air Force Base, Ohio: Air Force Flight Dynamics Laboratory, March, 1980.
10. Kuperman, G.G. and A.J. Defrances. "Airborne Electronic Terrain Mapping System: A Literature Review, AMRL-TR-79-92, Wright-Patterson Air Force Base, Ohio: Aerospace Medical Research Laboratory, October, 1979.
11. Kuperman, G.G., et al. "Information Requirements for Airborne Electronic Maps." *IEEE Transactions from NAECON* 1980, 1046-1051 (April 1980).
12. McDonnell Aircraft Company, "F-15 JTIDS Simulation." St. Louis, Mo: 1979.
13. Sander, D. E. Personal Communication.

14. Soliday, S.M. and J.R. Milligan. "Simulation of Low Altitude High Speed Mission Performance, Vol. II, Effectiveness of a HUD Display for Take-off and Landing in Fighter Aircraft." SEG-TR-66-67, Wright-Patterson Air Force Base, Ohio: Systems Engineering Group, February 1967.

15. Systran Corporation. "The Airborne Electronic Electronic Terrain Mapping System Documentation." Dayton, Ohio: Systran Corporation, 1978.

16. Tamburino, L.A. "Airborne Electronic Terrain Map Display System, AFAL-TR-77-232, Wright-Patterson Air Force Base, Ohio: Air Force Avionics Laboratory, December, 1977.

APPENDIX A

FLOWCHARTS

This appendix contains the flowcharts for SYMBOL and its sub-modules. Note the difference in figure 10 and figure's 11 and 12. Figure 10 represents a flow chart of data that has not been transformed to Fourier data. Figure's 11 and 12 represent the Fourier transform modules.

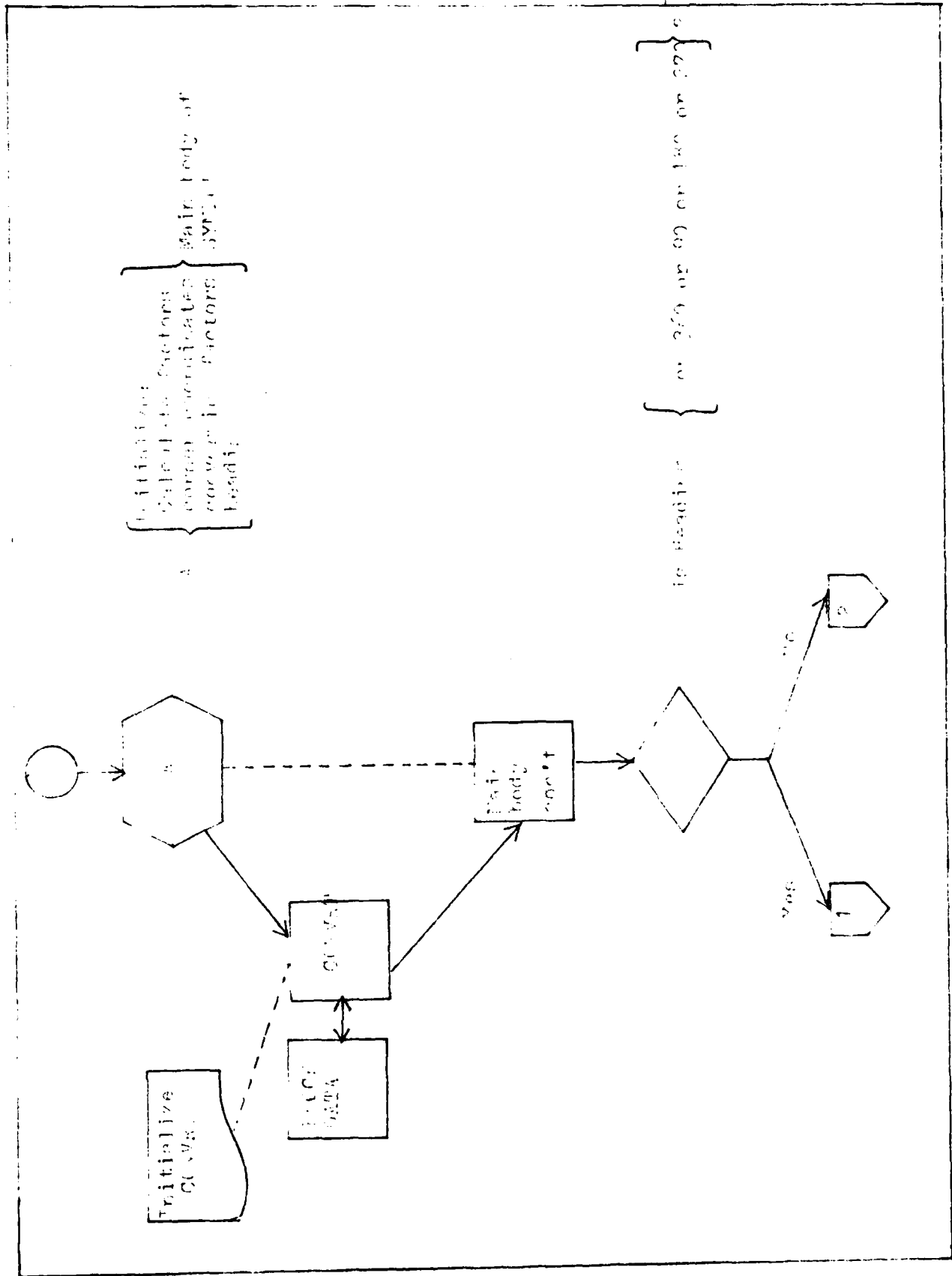
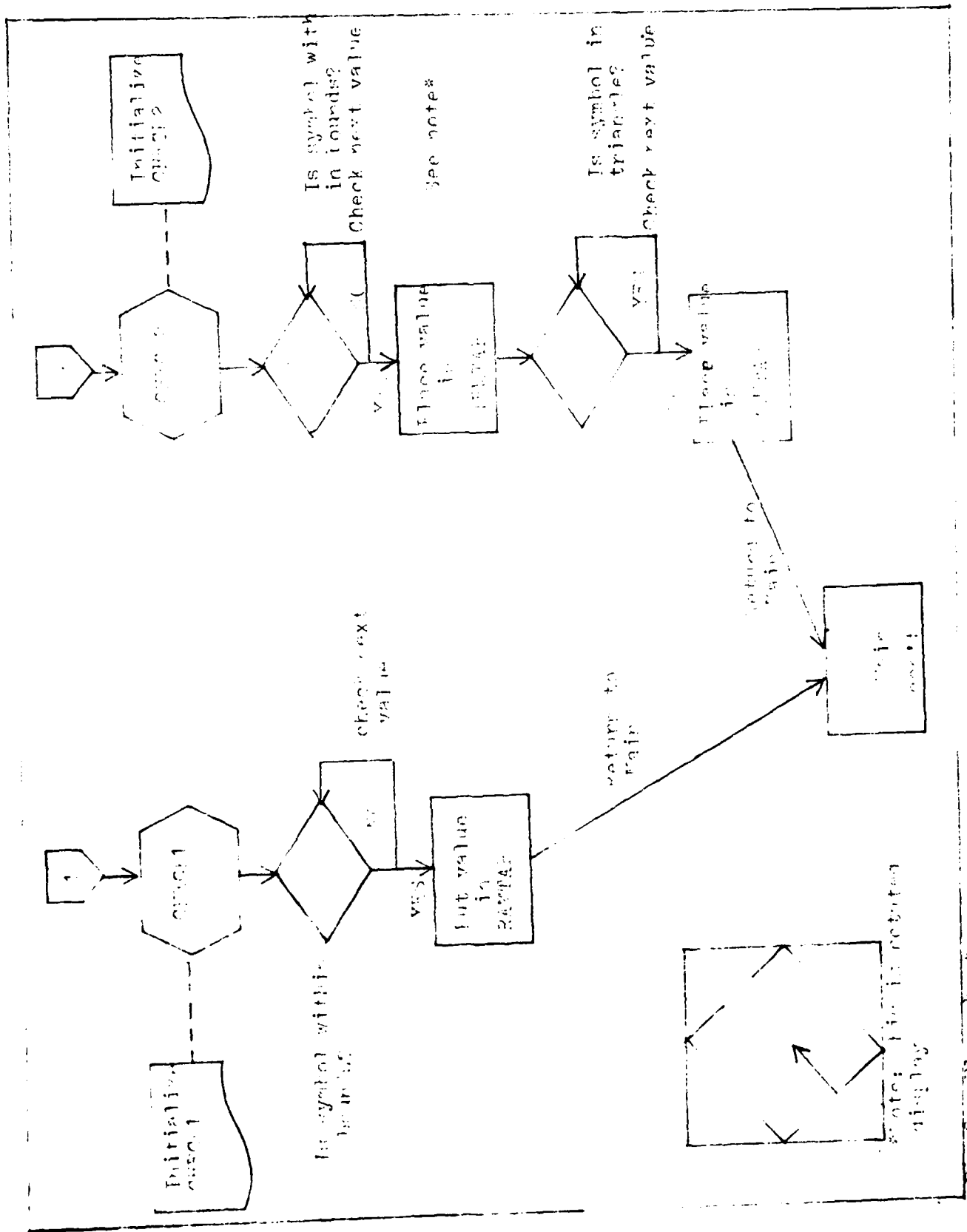
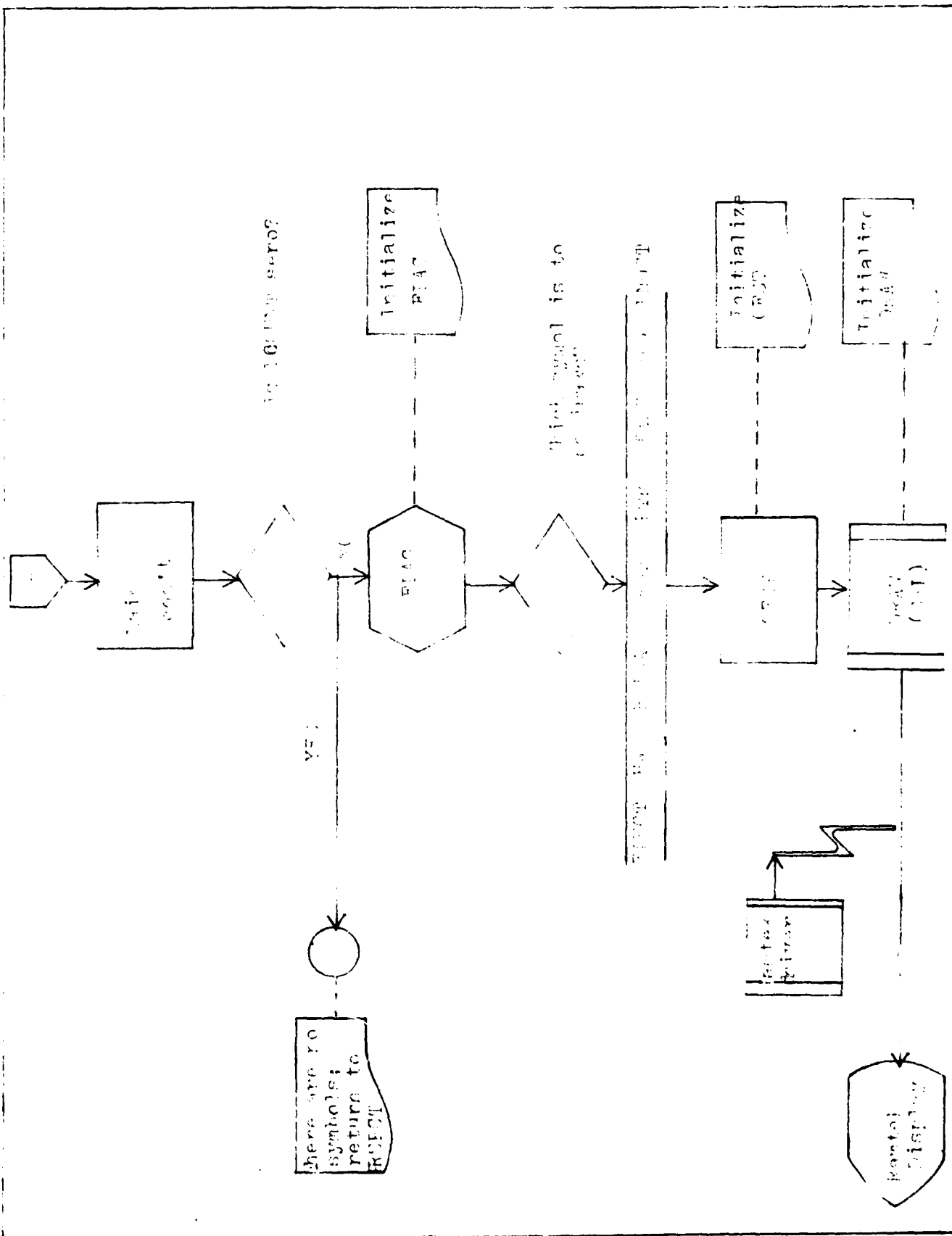


Fig. 2. Flow chart





Ch. 19. Coll. 1864

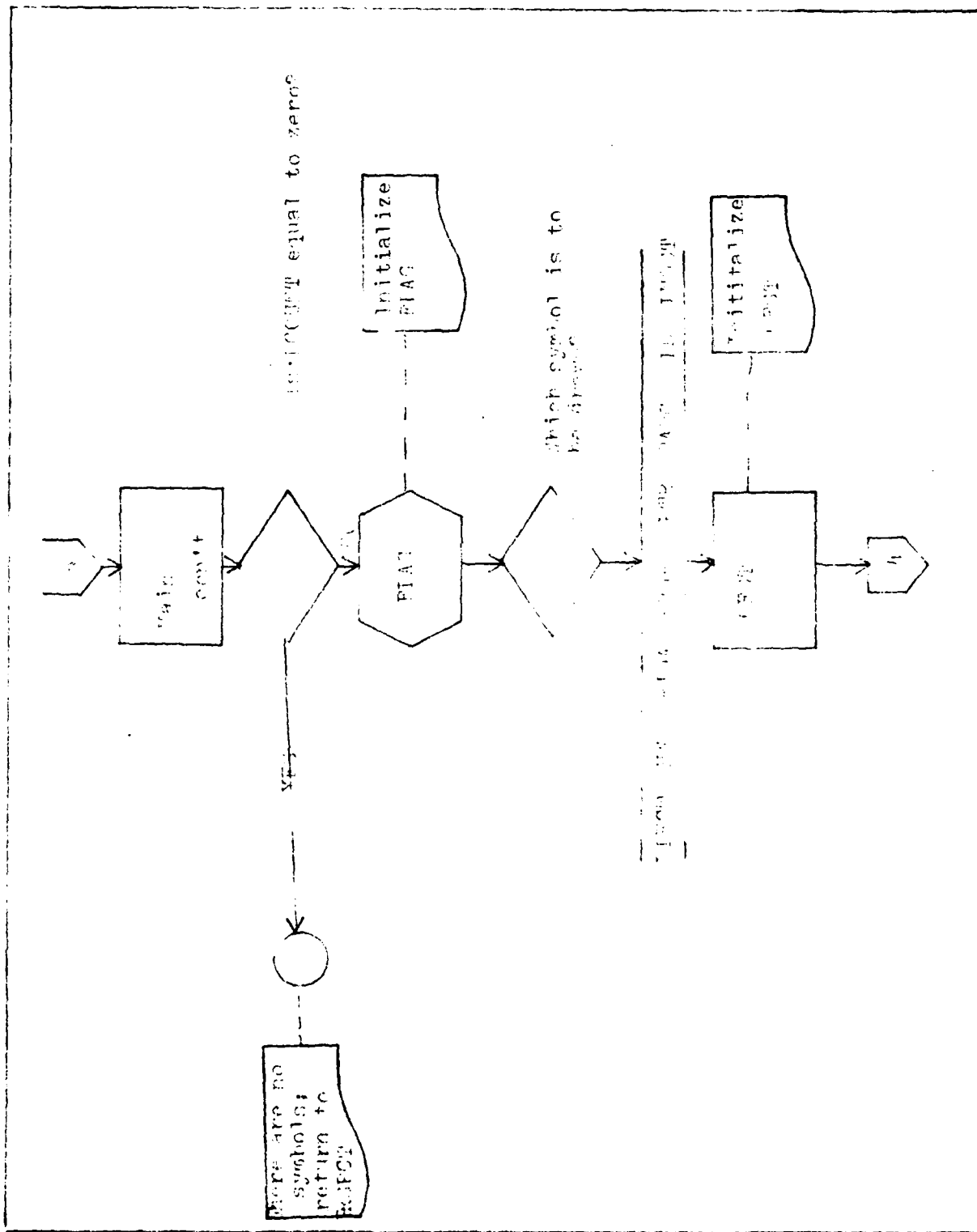


Fig. 11. Flow chart

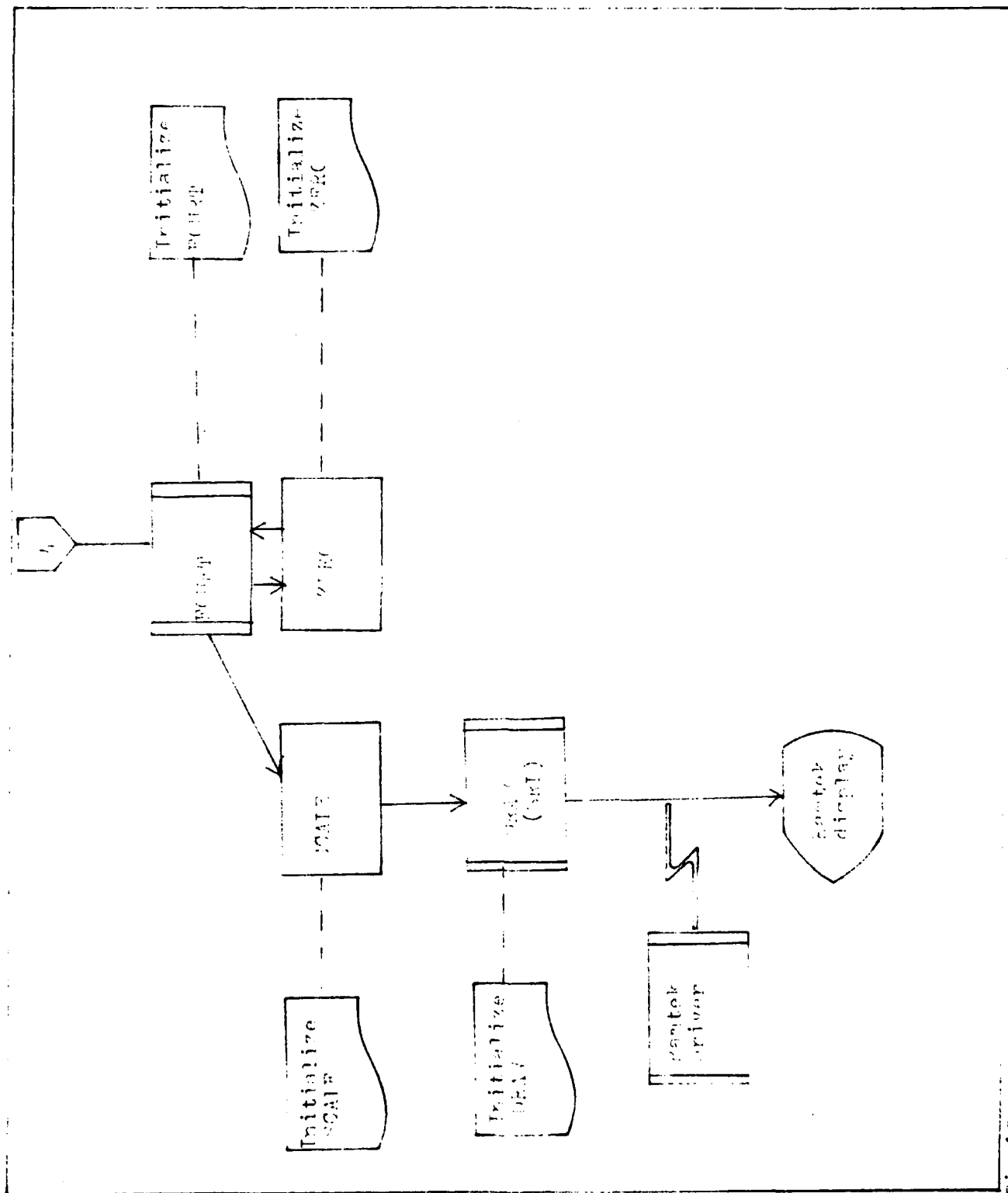


Fig 12. Flow Chart

APPENDIX B

VARIABLE AND TABLE DEFINITIONS

This appendix contains the variable and table definitions necessary to understand their use in SYMBOL and PRSPCT. Table IV describes the AETMS defined variables; Table V describes variables defined in SYMBOL; and Table VI describes the tables used in SYMBOL. Common areas are described in Table VII.

TABLE IV.

NAME	DESCRIPTION
BAU	$\pi/128$
B	1 BAU
A	aircraft azimuth
Patch	a digital description of a small area of terrain in memory
DH	display horizon (5.52)
DA	change in azimuth over one time slice
D	aircraft speed in knots
IWY	
IWX	
IX	
IY	
JX	
JY	
KX	
KY	Corner coordinates of the window

TABLE V.

NAME	DESCRIPTION
POSX POSY	Coordinate conversion factors
CONS CONC	sine and cosine factors
AX,AY BX,BY CX,CY DPX,DPY	Corner coordinates (window)
X;YMIN X;YMAX	boundary values
SAB SBC SAD SDC	Edge (window) slopes
LCOUNT	Number of symbols in the window

TABLE VI.

NAME	DESCRIPTION
RAMTAB	Table of values within the bounds
SYMTAB	Latitude and Longitude values in patch coordinates
TEMTAB	Temporary table
POSTAB	Latitude and Longitude values in real world coordinates

TABLE VII.

NAME	OWNERS
DTABLE	BLOCK DATA, CONVRT
DAREA	SYMBOL, OFST
DBLOCK	SYMBOL, CHECK1, CHECK2, CONVRT

APPENDIX C

EQUATIONS

This appendix contains the equations necessary to determine the conversion factors in symbol. All of these equations were developed by AETMS designers (Ref 15).

$$IAY, FAY = (IAY, FAY) + D(DA * (B/2) * SINA + COSA) / DY$$

$$IAX, FAX = (IAX, FAX) + D(SINA - DA * (B/2) * COSA) / DX$$

$$POSX = D(SINA - DA * (B/2) * COSA) / DX$$

$$POSY = D(DA * (B/2) * SINA + COSA) / DY$$

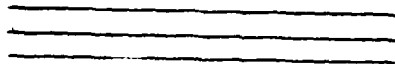
$$CONS = ((DH * SINA) / (2 * DX))$$

$$CONC = ((DH * COSA) / (2 * DY))$$

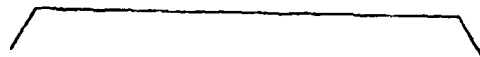
APPENDIX D

SYMBOL SET

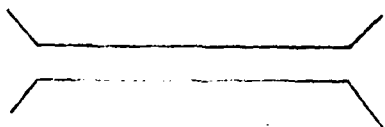
This appendix contains the eight symbols that can be displayed by the thesis software. The symbols are international symbols compiled by Dreyfuss (Ref 6).



HIGHWAY



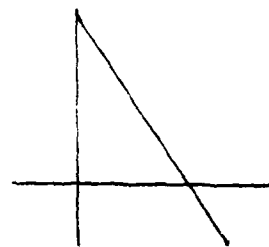
BRIDGE A



CUTTING



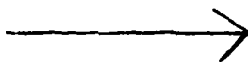
TUNNEL



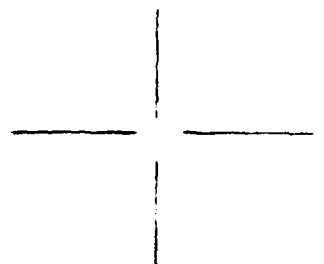
RAILROAD



POLYGONAL AREA



RIVER



RAILWAY STATION

APPENDIX E

EXPERIMENT INSTRUCTIONS AND DATA COLLECTION FORMS

This appendix contains the data collection forms necessary to compile dataB from the experiment. Included in this appendix are the background and instructions, an experiment questionnaire, an experiment tally sheet, and post experiment questionnaire.

Instructions and the background information are an important part of any experiment. If the subject is not interested in the experiment in which he will participate, then he may become bored or inattentive. Therefore, the subject is first given some background information about the AETMS and why the experiment is to be conducted. Furthermore, if the subject does not understand the instructions, he will not be able to complete his task and results may be invalid (Ref 3).

INSTRUCTIONS

This is a three part experiment. Part 1 is an orientation phase. Part 2 is an identification phase and Part 3 is a validation phase. I will give you some background on the AETMS and instructions before we begin. Are there any initial questions?

Part 1 is an orientation phase. You will be shown a sample terrain, symbols on a plain background, and symbols

overlayed on the terrain. Your task is to view the output so that you will have an idea of the visual output that will occur in the other two phases. Feel free to ask questions during this phase.

TASK 1 is presented to the subject.

Part 2 is an identification phase. We will interact in a question and answer format. You will be presented a symbol on the screen. Tell me what you think it is. I will validate your answer if you are correct, and correct your answer if it is incorrect. We go through the same sequence of events with different color combinations. Are there any questions?

TASK 2 is presented to the subject.

Part 3 is a validation phase. Symbols will be randomly presented. Once you see a symbol indicate what it is. Also indicate whether or not it is hostile or threatening. This is done by color coding. Colors and symbols seen in this phase are the same as seen in the previous phases.

TASK 3 is presented to the subject.

The experiment is now over. Please answer the questions on the questionnaire.

EXPERIMENT QUESTIONNAIRE

1. Task 2: What does the symbol represent?
2. Task 2: What does the color represent?
3. Any task: What comments do you have about this phase?

EXPERIMENT TALLY SHEET

TASK 2

OBJECT PLACE A 1 IF THE OBJECT/COLOR WAS CORRECTLY IDENTIFIED

Highway
Electrical lines
Bridge a
Bridge b
Pointer
Target
Landing strip
Building

COLOR

red
green
yellow
black
white

TASK 3

OBJECT COLOR COLOR/correct/SYMBOL (1 indicates correct)

Building green
Target red
Pointer red
Electrical lines yellow
Highway green
Landing strip green
Electrical lines red
Bridge A yellow
Target red
Bridge B yellow
Building black
Pointer white

POST EXPERIMENT QUESTIONNAIRE

1. Did you feel at ease during the experiment? If not, why not?

2. Were there too many distractions during the experiment?

3. Were the instructions clear and was there ample time to ask questions?

4. Rate the display with these factors. 1 indicates a high rating, 5 indicates a low rating.

Legibility 1 2 3 4 5

Understandibility 1 2 3 4 5

Functionality 1 2 3 4 5

Appeal 1 2 3 4 5

5. If you were in the position of accepting this display system of a SPO would you

a. Accept the system as you saw it in the experiment?

Yes No (if yes, goto 6)

b. Require a change to the terrain output? Yes No

c. Require a change to the symbols in the symbol set?

Yes No

What changes would you expect?

6. Imagine that you are a pilot tasked with destroying an enemy oil field. Your mission will take place at night and the weather leaves much to be desired. You therefore must depend on the display system in your aircraft. Will you feel at ease in trusting your life to this system?

Why?

7. Please feel free to make any other comments about the display or experiment.

APPENDIX F

USER'S GUIDE

This appendix contains a guide for those who would like to use the thesis software.

The following short program can be added to call SYMBOL.

```
Program Runner
Print*, 'input the information that will simulate AETMS inputs'
Read*, IAY, IAX, FAY, FAX, A, DX, DY, NX, NY, D, DA
Call SYMBOL(IAY, IAX, FAY, FAX, A, DX, DY, NX, NY, D, DA)
END
```

Figure 14 represents BLOCK DATA as it is in the thesis software. The first 21 entries represent IAX, FAX and the second 21 represent IAY, FAY. The last 21 entries represent flag values. The following equation will break each table entry to a minutes and degrees entry. Input into RUNNER must be in minutes and degrees.

Let the entry we are to consider be the first one. IAX, FAX= 15.5 Break down is done in the following manner: IAX=15 (integer); FAX=.5*60 (real) the IAY that corresponds to 15.5 is 10.75. Break down is similar to the IAX, FAX break down. IAY=10 (integer) and FAY=.75*60 (real). Input for these two values is: IAY IAX FAY FAX DX DY A NX NY D DA
10 15 30. 30. 10. 10. 0. 64 64 0. 1.

All these values are defined in APPENDIX B. A and DA must be input as BAU values. Conversion from degrees to BAU's is done with the following equation:

$$(\text{azimuth in degrees}) * (\text{PI}/128)$$

```

DATA POS1AB/15.5,25.5,35.5,45.5,55.5,65.5,75.5,
*85.50,86.79,87.00,88.12,88.59,89.10,89.9,83.51,84.26,
*85.59,85.22,86.13,86.61,85.0,10.75,20.75,30.75,40.75,50.75,
*60.76,70.75,80.75,35.22,35.39,35.59,35.45,35.16,34.21,34.0,
*34.26,34.59,34.16,34.76,34.61,35.0,20.0,30.0,40.0,50.0,
*60.0,70.0,80.0,90.0,90.0,30.0,20.0,20.0,20.0,20.0,20.0,
*20.0,20.0,20.0,20.0,20.0,20.0/

```

Figure 14. Block Data

Similar conversion for the other values in BLOCK DATA
will allow the user to the ability to retrieve all symbols.

APPENDIX G

SOFTWARE LISTING

This appendix contains a listing of all the modules in SYMBOL and the test data. Note that there are three subroutines named FLAG. This was necessary due to the fact that there is only 28K of usable memory in the PDP 11/45. The program could not be compiled as one unit. Therefore, there are three separate tasks. TASK I is the entire module with eight symbol drawing routines. TASK II uses the same SYMBOL, CHECKI, CHECKII, OFST, and CONVRT that TASK I uses. TASK II contains subroutines that output four of the symbols as Fourier data, while TASK III outputs the other four. FLAG is varied depending on which task it is a part of.

Also contained in this listing of the executive routine in the AETMS system, PRSPCT. SYMBOL is called by PRSPCT just before the end of the controlling loop.

```

SUBROUTINE SYMUL(CAY, IAX, EAY, FAX, A, DX, DY,
  ANX, NY, D, DA)
  COMMON/DHL/DLCK/SYNTAB(21,3),JINLN,RANTAB(21,3),
  ATENTAR(21,3),IHEAD,LCOUNT,CHECK
  COMMON/DAEA/XMIN,XMAX,YMIN,YMAX
  DIMENSION SORT(8,8)

```

THIS SUBROUTINE WAS WRITTEN BY LT. MARY A. SMART, AFIT(GCS-8WD) FOR A THESIS SPONSORED BY THE INFORMATION AND PRESENTATION GROUP OF THE AIR FORCE AVIONICS LAB, WRIGHT-PATTERSON AIR FORCE BASE, OHIO. IT IS AN OVERLAY ROUTINE, I.E., INFORMATION GENERATED BY THE ROUTINE IS WRITTEN ON TOP OF INFORMATION GENERATED BY THE MAIN PROGRAM. THE MAIN USE FOR THIS SUBROUTINE IS IN THE AIRBORNE ELECTRONIC TERRAIN MAPPING SYSTEM (AETMS) DEVELOPED BY THE AVIONICS LAB AND WRITTEN BY THE SYSTRAN CORPORATION. SYSTRAN DOCUMENTATION GIVES AN OVERVIEW OF THE AETMS SOFTWARE.

THE ENTIRE MODULE CONSISTS OF 15 SUBROUTINES IN THE FOLLOWING HIERARCHY:

```

LEVEL 0: SYMUL(MAIN)
LEVEL 1: CONVRT
          CHECK1
          CHECK2
          FLAG
LEVEL 2: IN
          ELECT
          BRIDA
          BRDB
          TARG
          HLD
          PTR
          LNDM
LEVEL 3: ORAM
          OFST

```

ALGORITHM FOR MAIN

- STEP 0: INITIALIZE ALL VARIABLES WITH ARGUMENTS RECEIVED FROM PRSPCT(AETMS).
- STEP 1: CONVERT REAL WORLD LATITUDE AND LONGITUDE TO LATITUDE AND LONGITUDE IN PATCH COORDINATES.
- STEP 2: DETERMINE THE HEADING AND CALL THE APPROPRIATE BOUNDARY CHECKING ROUTINE.
- STEP 3: DETERMINE WHICH SYMBOL BELONGS AT A PARTICULAR LATITUDE AND LONGITUDE.

[illegible]

```

00139      FCX=(CX-ICX)*60
00140      CY=(IAY+(2*CONC))+(FAY+(2*CONC))/60
00141      ICY=IFIX(CY)
00142      FCY=(CY-ICY)*60
00143      BX=(IOX-CONC)+(FOX-CONC)/60
00144      IBX=IFIX(BX)
00145      FBX=(BX-IBX)*60
00146      HY=(IOY+CONS)+(FOY+CONS)/60
00147      IBY=IFIX(HY)
00148      FBY=(HY-IBY)*60
00149      DPX=(IBX+(2*CONC))+(FOX+(2*CONC))/60
00150      IDPX=IFIX(DPX)
00151      FDPX=(DPX-IDPX)*60
00152      DPY=(IBY-(2*CONS))+(FOY-(2*CONS))/60
00153      IDPY=IFIX(DPY)
00154      FDPY=(DPY-IDPY)*60
00155      IMY=(IAY+(DIACUS(A))/(2*DY))/(2*DY)-(NY-2)/2
00156      IMX=(IAX+(DIASIN(A))/(2*DX))/(2*DX)-(NX-2)/2
00157      IX=IMX
00158      IY=IMY
00159      JX=IMX
00160      JY=IMY+(NY-2)
00161      KX=IMX+(NX-2)
00162      KY=IMY+(NY-2)
00163      PRINT*, 'CORNER COORDINATES'
00164      TYPE=AX,AY,BX,BY,CX,CY,DPX,DPY
00165      PRINT*, ' '
00166      DO 5 M=1, JTRLN
00167      DO 5 J=1, 3
00168      RANTAB(M,J)=0
00169      TETTAB(M,J)=0
00170      CONTINUE
00171      SORT(1,1)=AX
00172      SORT(1,2)=BX
00173      SORT(1,3)=CX
00174      SORT(1,4)=DPX
00175      SORT(2,1)=AY
00176      SORT(2,2)=BY
00177      SORT(2,3)=CY
00178      SORT(2,4)=DPY
00179      XMAX=0
00180      XMIN=999
00181      DO 17 M=1, 4

```

```

0082 IF(XMIN .LT. SURT(1,M))GOTO 27
0083 XMIN=SORT(1,M)
0084 YIPART=SORT(2,M)
0085 IF(XMAX .GT. SURT(1,M))GOTO 17
0086 XMAX=SORT(1,M)
0087 YAPART=SORT(2,M)
0088 CONTINUE
0089 YMAX=0
0090 YMIN=9999
0091 DO 18 M=1,4
0092 IF(YMIN .LT. SURT(2,M))GOTO 28
0093 YMIN=SORT(2,M)
0094 XIPART=SORT(1,M)
0095 IF(YMAX .GT. SURT(2,M))GOTO 18
0096 XMAX=SORT(2,M)
0097 XAPART=SORT(1,M)
0098 CONTINUE
0099 PRINT*,XMIN,XMAX,YMIN,YMAX,
0100 YIPART,XMIN,XMAX,YMIN,YMAX
0101 PRINT*,
0102 IF((IHEAD .EQ. 0) .OR. (IHEAD .EQ. 90) .OR.
* (IHEAD .EQ. 180) .OR. (IHEAD .EQ. 270) .OR.
* (IHEAD .EQ. 360)) CALL CHECK1(XMIN,XMAX,YMIN,YMAX)
0103 IF((IHEAD .NE. 0) .AND. (IHEAD .NE. 90) .AND.
* (IHEAD .NE. 180) .AND. (IHEAD .NE. 270) .AND.
* (IHEAD .NE. 360))CALL CHECK2(AX,AY,HX,BY,CX,CY,DPX,DPY,EX,
* AY,XMIN,XMAX,YMIN,YMAX)
0104 IF(LCOUNT .EQ. 0)GOTO 10
0105 PRINT*, ' THE RAMEK TABLE'
0106 DO 777 M=1,LCOUNT
0107 TYPE 778,(RAMTAB(M,J),J=1,5)
0108 FORMAT(3(2X,F9.0))
0109 CONTINUE
0110 CALL FLAG(LCOUNT,RAMTAB)
0111 RETURN
0112 END

```

```

C C C SUBROUTINE CHECK1(XMIN,XMAX,YMIN,YMAX)
0001 CONTINUE CHECK1(XMIN,XMAX,YMIN,YMAX)
0002 COMBIN=NBLOCK/SYNTAB(21,3),JTBLEN,RANTAB(21,3),
      *TEHTAB(21,3),LINEAD,COUNT,CHECK
C C C ALGORITHM(CHECK1)
C C C STEP 08 INITIALIZE VARIABLES WITH ARGUMENTS
C C C RECEIVED FROM MAIN.
C C C STEP 18 DO FOR ALL VALUES
C C C COMPARE TABLE VALUES AGAINST MAXIMUM
C C C AND MINIMUM VALUES, IF VALUE IS WITHIN
C C C THESE VALUES PLACE THE VALUE IN THE RANTEX
C C C TABLE.
C C C STEP 20 COUNT THESE VALUES.
C C C STEP 30 RETURN TO MAIN.
C C C LCOUNT=0
C K=0
C J=1
C DO 10 I=1,JTBLEN
0003 XVAL=SYNTAB(I,1)
0004 YVAL=SYNTAB(I,2)
0005 ZVAL=SYNTAB(I,3)
C C C IF ((YVAL .GT. YMAX) .OR. (XVAL .LT. YMIN)
      *.OR. (XVAL .GT. XMAX) .OR. (XVAL .LT. XMIN))
0006 *GOTO 10
C RANTAB(J,1)=XVAL
0007 RANTAB(J,2)=YVAL
0008 RANTAB(J,3)=ZVAL
C K=K+1
0009 J=J+1
C LCOUNT=LCOUNT+1
C CONTINUE
C IF K.EQ. 0,LCOUNT=0
0010 CHECK=1.0
C RETURN
C END
C
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
```

```

0001 SUBROUTINE CHECK2(AX,AY,BX,RY,CX,CY,DPX,DPY,BX,RY,XMIN,
0002 AXIAX,YMIN,YMAX)
      COMMON/DIM/CK/SYNTAB(21,3),JHLEN,KAMTAB(21,5),
      *LENTAB(21,3),IHEAD,LCOUNT,CHECK
      ALGORITHM(CHECK2)
      STEP 00 INITIALIZE VARIABLES WITH ARGUMENTS
      RECEIVED FROM MAIN.
      STEP 10 DO FOR ALL VALUES:
      DETERMINE IF VALUE IS IN WINDOW. IF VALUE IS
      WITHIN THESE BOUNDS, PLACE THE VALUE IN A TEMPORARY
      TABLE.
      STEP 20 COUNT THESE VALUES.
      STEP 30 DO FOR ALL VALUES IN TEMP:
      CHECK THESE VALUES AGAINST THE ROTATED BOUNDS.
      IF THE VALUE IS WITHIN THE BOUNDS, PLACE THE
      VALUE IN THE RANKEK TABLE.
      STEP 40 COUNT THESE VALUES.
      STEP 50 RETURN TO MAIN.
      IF (AY .NE. YMIN) GOTO 5
      SAB=(RY-AY)/(BX-AX)
      SBC=(CY-RY)/(CX-BX)
      SDC=(DPY-CY)/(DPX-CX)
      SAD=(AY-DPY)/(AX-DPX)
      SAXY=SAB
      SHXY=SBC
      SCXY=SDC
      SDXY=SAD
      IF (DPY .NE. YMIN) GOTO 10
      SAB=(DPY-AY)/(DPX-AX)
      SBC=(AY-RY)/(AX-BX)
      SDC=(RY-CY)/(BX-CX)
      SAD=(CY-DPY)/(CX-DPX)
      SAXY=SBC
      SBXY=SDC
      SCXY=SAD
      SDXY=SAB
      IF (CY .NE. YMIN) GOTO 15
      SAB=(CY-DPY)/(CX-DPX)
      SBC=(DPY-AY)/(DPX-AX)

```

```

0024      SDC=(AY-HY)/(AX-BX)
0025      SAD=(BY-CY)/(BX-CX)
C
0026      SAXY=SDC
0027      SHXY=SAD
0028      SLXY=SAB
0029      SDXY=SBC
C
0030      IF (BY .NE. YMIN) GO TO 35
0031      SAR=(BY-CY)/(BX-CX)
0032      SRC=(CY-DY)/(CX-DPX)
0033      SDC=(DPY-AY)/(DPX-AX)
0034      SAD=(BY-AY)/(BX-AX)
C
0035      SAXY=SAD
0036      SHXY=SAB
0037      SLXY=SBC
0038      SDXY=SDC
C
C
C
0039      PH1X=(XMIN+XAPART)/2
0040      PH1Y=(YIPART+YMAX)/2
C
0041      PH2X=(XAPART+XMAX)/2
0042      PH2Y=(YMAX+YAPART)/2
C
0043      PH3X=(XMAX+XIPART)/2
0044      PH3Y=(YAPART+YMIN)/2
C
0045      PH4X=(XIPART+XMIN)/2
0046      PH4Y=(YMIN+YIPART)/2
C
0047      C1X=(XAPART+OX)/2
0048      C1Y=(YMAX+OY)/2
C
0049      C2X=(XMAX+OX)/2
0050      C2Y=(YAPART+OY)/2
C
0051      C3X=(XIPART+OX)/2
0052      C3Y=(YMIN+OY)/2
C
0053      C4X=(XMIN+OX)/2
0054      C4Y=(YIPART+OY)/2
C
C
C
0055      LCOUNT=M
0056      CHECK=M*U
0057      K=0
0058      J=1
0059      DO 20 I=1, JTULN
0060      XVAL=SYNTAB(I,1)
0061      YVAL=SYNTAB(I,2)
0062      ZVAL=SYNTAB(I,3)
C

```

```

00063      IF((YVAL .GT. YMAX) .OR. (YVAL .LT. YMIN)
A      .OR. (XVAL .GT. XMAX) .OR. (XVAL .LT. XMIN))
A      GOTO 20

```

```

00064      TENTAB(J,1)=XVAL
00065      TENTAB(J,2)=YVAL
00066      TENTAB(J,3)=ZVAL
00067      K=K+1
00068      J=J+1
00069      CONTINUE

```

```

00070      IF(K .EQ. 0) GOTO 35
00071      J=1
00072      DO 30 I=1,K
00073          X=TENTAB(I,1)
00074          Y=TENTAB(I,2)
00075          Z=TENTAB(I,3)

```

```

00076      IF((X .EQ. AX) .OR. (X .EQ. BX) .OR. (X .EQ. CX)
A      .OR. (X .EQ. DPX)) .AND. ((Y .EQ. AY) .OR.
A      (Y .EQ. BY) .OR. (Y .EQ. CY) .OR. (Y .EQ. DPY)))
A      GOTO 25

```

```

00077      IF((PMIX .LE. X) .AND. (X .LE. CIX) .AND.
A      (Y .GE. CIY)) SAXY=(Y-YIPART)/(X-XMIN)

```

```

00078      IF((CIX .LE. X) .AND. (X .LE. PH2X) .AND.
A      (Y .GE. CIY)) SBXY=(Y-YMAX)/(X-XAPART)

```

```

00079      IF((C2Y .LE. Y) .AND. (Y .LE. PH2Y) .AND.
A      (X .GE. CIX)) SBXY=(Y-YMAX)/(X-XAPART)

```

```

00080      IF((PM3Y .LE. Y) .AND. (Y .LE. C2Y) .AND.
A      (X .GE. C2X)) SCXY=(Y-YAPART)/(X-XMAX)

```

```

00081      IF((C5X .LE. X) .AND. (X .LE. PM3X) .AND.
A      (Y .LE. C3Y)) SCXY=(Y-YAPART)/(X-XMAX)

```

```

00082      IF((PM4X .LE. X) .AND. (X .LE. C3X) .AND.
A      (Y .LE. C3Y)) SDXY=(Y-YMIN)/(X-XIPART)

```

```

00083      IF((PM4Y .LE. Y) .AND. (Y .LE. C4Y) .AND.
A      (X .LE. C4X)) SDXY=(Y-YMIN)/(X-XIPART)

```

```

00084      IF((C4Y .LE. Y) .AND. (Y .LE. PM1Y) .AND.
A      (X .LE. C4X)) SAXY=(Y-YIPART)/(X-XMIN)

```

```

00085      IF((SAXY .LT. SRC) .OR. (SBXY .GT. SRC) .OR.
A      (SCXY .GT. SAD) .OR. (SDXY .LT. SAR)) GOTO 30

```

```

00086      RANTAB(J,1)=X
00087      RANTAB(J,2)=Y

```

0088
0089
0090
0091
0092
0093

PARIA(J,5)Z
LLOUNT=CCOUI+1
J2J+1
CONTINUL
30
C 35 RETURN
END

0001
0002

0001
0002

0003

0003

0004

0004

0005

0005

0006

0007

0008

0009

0010

0011

0012

0013

0014

0015

0016

0017

0018

0019

0020

0021

0022

0023

0024

0025

0026

0027

0028

0029

0030

0031

0032

0033

0034

0035

0036

0037

0038

0039

0040

0041

0042

0043

0044

0045

0046

0047

0048

0049

0050

0051

0052

0053

0054

0055

0056

0057

0058

0059

0060

0061

0062

0063

0064

0065

0066

0067

0068

0069

0070

0071

0072

0073

0074

0075

0076

0077

0078

0079

0080

0081

0082

0083

0084

0085

0086

0087

0088

0089

0090

0091

0092

0093

0094

0095

0096

0097

0098

0099

0100

0101

0102

0103

0104

0105

0106

0107

0108

0109

0110

0111

0112

0113

0114

0115

0116

0117

0118

0119

0120

0121

0122

0123

0124

0125

0126

0127

0128

0129

0130

0131

0132

0133

0134

0135

0136

0137

0138

0139

0140

0141

0142

0143

0144

0145

0146

0147

0148

0149

0150

0151

0152

0153

0154

0155

0156

0157

0158

0159

0160

0161

0162

0163

0164

0165

0166

0167

0168

0169

0170

0171

0172

0173

0174

0175

0176

0177

0178

0179

0180

0181

0182

0183

0184

0185

0186

0187

0188

0189

0190

0191

0192

0193

0194

0195

0196

0197

0198

0199

0200

0201

0202

0203

0204

0205

0206

0207

0208

0209

0210

0211

0212

0213

0214

0215

0216

0217

0218

0219

0220

0221

0222

0223

0224

0225

0226

0227

0228

0229

0230

0231

0232

0233

0234

0235

0236

0237

0238

0239

0240

0241

0242

0243

0244

0245

0246

0247

0248

0249

0250

0251

0252

0253

0254

0255

0256

0257

0258

0259

0260

0261

0262

0263

0264

0265

0266

0267

0268

0269

0270

0271

0272

0273

0274

0275

0276

0277

0278

0279

0280

0281

0282

0283

0284

0285

0286

0287

0288

0289

0290

0291

0292

0293

0294

0295

0296

0297

0298

0299

0300

0301

0302

0303

0304

0305

0306

0307

0308

0309

0310

0311

0312

0313

0314

0315

0316

0317

0318

0319

0320

0321

0322

0323

0324

0325

0326

0327

0328

0329

0330

0331

[illegible]

0035		IFLAG=0
0036		PRINT*, 'THE LANDING STRIP HAS BEEN DRAWN'
0037		GOTO 30
	C	
0038	13	IF(RAMTAB(L,3) .NE. 80)GOTO 14
0039		CALL TARG(J, RAMTAB, IFLAG)
0040		IFLAG=0
0041		PRINT*, 'THE TARGET SYMBOL HAS BEEN DRAWN'
0042		GOTO 30
	C	
0043	14	IF(RAMTAB(L,3) .NE. 90)GOTO 15
0044		CALL RID(J, RAMTAB, IFLAG)
0045		IFLAG=0
0046		GOTO 30
	C	
0047	15	PRINT*, 'ERROR IN FLAG'
	C	
0048	30	L=L+1
0049		GOTO 4
0050		RETURN
0051		END

```

C      SUBROUTINE DRAM(MAT,IFLAG)
C
C      THIS SUBROUTINE WAS ADAPTED FROM A PROGRAM
C      WRITTEN BY DALE RAINGER OF SRL, INC.
C      PREVIOUS PROGRAM NAME: FINAL,V3.0,
C      9 SEPTEMBER 1980.
C
C      IMPLICIT INTEGER(A=7)
C      DIMENSION RANDAT(10),RAMPRM(6),IUSTAT(2),MA(5,5)
C      BYTE IOBYTE(2)
C      EQUIVALENCE (IOBYTE(1),IUSTAT(1))
C      MLB=#J00
C      HSLT=#2100
C      INOP=#3400
C      MSK=#1
C      FGD=#2
C      COP=#10000
C      MV=#7000
C      DF=#1
C      UF=#2
C
C      CALL ASHLUN(1,'XR',0)
C      RANDAT(1)=INOP
C      IF(IFLAG.EQ.1)RANDAT(1)=RSET
C      RANDAT(2)=(MV.OR.UF.OR.UF)
C      RANDAT(3)=(COP.OR.FGD.OR.MSK)
C      RANDAT(4)=#3400
C      RANDAT(5)=#2000
C      RANDAT(6)=MAT(1,1)
C      RANDAT(7)=MAT(1,2)
C      RANDAT(8)=4
C      RANDAT(9)=MAT(2,1)
C      RANDAT(10)=MAT(2,2)
C
C      CALL GETADR(RAMPRM,RANDAT)
C      RAMPRM(2)=20
C      RAMPRM(3)=0
C
C      CALL WTQIO(MLB,1,2,3,IUSTAT,RAMPRM,IER)
C
C      TYPE*, 'DIRECTIVE STATUS=', IER
C      TYPE*, 'IUSTAT =', IOBYTE(1)
C
C      RETURN
C      END

```

```

C C THE FOLLOWING ALGORITHM IS THE SAME FOR ALL SYMBOL
C C DRAWING SUBROUTINES.
C C
C C ALGORITHM
C C
C STEP 08 INITIALIZE VARIABLELS WITH ARGUMENTS
C FROM FLAG.
C
C STEP 18 SCALE THE STARTING POINT TO VIDEO SCREEN SIZE.
C
C STEP 28 DRAW THE PICTURE.
C
C STEP 38 RETURN TO FLAG.
C
SUBROUTINE HM(J, RANTAB, IFLAG)
IMPLICIT INTEGER(H)
DIMENSION RANTAB(21,3), IHM(5,5), RANTBL(21,3)
ICOUNT=3
L=J
CALL OFST(L,RANTAB,RANTBL)
IHM(1,1)=FIX(RANTBL(J,1)+0.5)
IHM(1,2)=FIX(RANTBL(J,2)+0.5)
IHM(2,1)=IHM(1,1)+43
IHM(2,2)=IHM(1,2)
CALL DRAW(IHM,IFLAG)
IFLAG=0
ICOUNT=ICOUNT+1
IF ICOUNT.EQ. 6 GO TO 250
RANTBL(J,2)=RANTBL(J,2)+14
GO TO 10
250 RETURN
END

```

0001	SUBROUTINE ELECT(J, RAMTAB, IFLAG)
0002	IMPLICIT INTEGER(I)
0003	DIMENSION RAMTAB(21, 5), LLEC(5, 5), RAMIHL(21, 3)
0004	ICOUNT=0
0005	CALL DFST(J, RAMTAB, RAMIHL)
0006	LLEC(1, 1)=1 IX(RAMTBL(J, 1)+0, 5)
0007	LLEC(1, 2)=RAMTBL(J, 2)
0008	LLEC(2, 1)=LLEC(1, 1)+23
0009	LLEC(2, 2)=LLEC(1, 2)
0010	RAMTBL(J, 1)=LLEC(2, 1)+36
0011	CALL DRAW(LLEC, IFLAG)
0012	IFLAG=0
0013	ICOUNT=ICOUNT+1
0014	IF (ICOUNT .GT. 5) GOTO 999
0015	GOTO 4
0016	RETURN
0017	END

4

C

C

999

0001	C	SUBROUTINE HRDA(J, RANTAB, IFLAG)
0002	C	IMPLICIT INTEGER(I)
0003	C	DIMENSION RANTAB(21,3), HRDA(5,5), HOLD(5), RANTHL(21,3)
0004	C	CALL UFST(J, RANTAB, RANTBL)
0005	C	HRDA(1,1)=FIX(RANTBL(J,1)+0.5)
0006		HRDA(1,2)=FIX(RANTBL(J,2)+0.5)
0007		HRDA(2,1)=HRDA(1,1)+23
0008		HOLD(1)=HRDA(2,1)
0009		HRDA(2,2)=HRDA(1,2)-23
0010		HOLD(2)=HRDA(2,2)
0011	C	CALL DRAW(HRDA, IFLAG)
0012	C	IFLAG=0
0013		HRDA(1,1)=HOLD(1)
0014		HRDA(1,2)=HOLD(2)
0015		HRDA(2,1)=HRDA(1,1)+35
0016		HOLD(1)=HRDA(2,1)
0017		HRDA(2,2)=HRDA(1,2)
0018		HOLD(2)=HRDA(2,2)
0019	C	CALL DRAW(HRDA, IFLAG)
0020	C	HRDA(1,1)=HOLD(1)
0021		HRDA(1,2)=HOLD(2)
0022		HRDA(2,1)=HRDA(1,1)+23
0023		HRDA(2,2)=HRDA(1,2)+23
0024	C	CALL DRAW(HRDA, IFLAG)
0025		RETURN
0026		END

```

C
C
C
0001 SUBROUTINE BRD(J,RAMTAB,IFLAG)
0002 IMPLICIT INTEGER(B)
0003 DIMENSION RAMTAB(21,5),BRD(5,5),HOLD(5),RAMTBL(21,5)
C
0004 CALL OFST(J,RAMTAB,RAMTBL)
C
0005 BRD(1,1)=IFIX(RAMTBL(J,1)+0.5)
0006 BRD(1,2)=IFIX(RAMTBL(J,2)+0.5)
0007 BRD(2,1)=BRD(1,1)+23
0008 HOLD(1)=BRD(2,1)
0009 BRD(2,2)=BRD(1,2)+23
0010 HOLD(2)=BRD(2,2)
C
0011 CALL DRAW(BRD,IFLAG)
0012 IFLAG=0
C
0013 BRD(1,1)=HOLD(1)
0014 BRD(1,2)=HOLD(2)
0015 BRD(2,1)=BRD(1,1)+35
0016 HOLD(1)=BRD(2,1)
0017 BRD(2,2)=BRD(1,2)
0018 HOLD(2)=BRD(2,2)
C
0019 CALL DRAW(BRD,IFLAG)
C
0020 BRD(1,1)=HOLD(1)
0021 BRD(1,2)=HOLD(2)
0022 BRD(2,1)=BRD(1,1)+23
0023 BRD(2,2)=BRD(1,2)-23
C
0024 CALL DRAW(BRD,IFLAG)
C
0025 BRD(1,1)=IFIX(RAMTBL(J,1)+0.5)
0026 BRD(1,2)=IFIX(RAMTBL(J,2)+0.5)+69
0027 BRD(2,1)=BRD(1,1)+23
0028 HOLD(1)=BRD(2,1)
0029 BRD(2,2)=BRD(1,2)-23
0030 HOLD(2)=BRD(2,2)
C
0031 CALL DRAW(BRD,IFLAG)
C
0032 BRD(1,1)=HOLD(1)
0033 BRD(1,2)=HOLD(2)
0034 BRD(2,1)=BRD(1,1)+35
0035 HOLD(1)=BRD(2,1)
0036 BRD(2,2)=BRD(1,2)
0037 HOLD(2)=BRD(2,2)
C
0038 CALL DRAW(BRD,IFLAG)
C
0039 BRD(1,1)=HOLD(1)
0040 BRD(1,2)=HOLD(2)

```

0041 BRD(2,1)=BRD(1,1)+23
0042 BRD(2,2)=BRD(1,2)+23
0043 CALL DRAW(BRD,IFLAG)
0044 RETURN
0045 END

C

0041
0042
0043
0044
0045

```

0001      SUBROUTINE PTR(J,NAHTAB,IFLAG)
0002      IMPLICIT INTEGER(P)
0003      DIMENSION NAHTAB(21,5),PTRM(5,5),HOLD(5),RAMIBL(21,3)
0004      CALL OFST(J,NAHTAB,RAMIBL)
0005      PTRM(1,1)=IFIX(RAMIBL(J,1)+0.5)
0006      PTRM(1,2)=IFIX(RAMIBL(J,2)+0.5)
0007      PTRM(2,1)=PTRM(1,1)+150
0008      HOLD(1)=PTRM(2,1)
0009      PTRM(2,2)=PTRM(1,2)
0010      HOLD(2)=PTRM(2,2)
0011      CALL DRAW(PTRM,IFLAG)
0012      IFLAG=0
0013      PTRM(1,1)=HOLD(1)
0014      PTRM(1,2)=HOLD(2)
0015      PTRM(2,1)=PTRM(1,1)-43
0016      HOLD(1)=PTRM(2,1)
0017      PTRM(2,2)=PTRM(1,2)-43
0018      HOLD(2)=PTRM(2,2)
0019      CALL DRAW(PTRM,IFLAG)
0020      PTRM(1,1)=HOLD(1)
0021      PTRM(1,2)=HOLD(2)+86
0022      PTRM(2,1)=PTRM(1,1)+43
0023      PTRM(2,2)=PTRM(1,2)+43
0024      CALL DRAW(PTRM,IFLAG)
0025      RETURN
0026      END

```

```

0001      SUBROUTINE LNDST(J,RAMTAB,IFLAG)
0002      DIMENSION RAMTAB(21,5),LNDM(5,5),HOLD(5),RAMTBL(21,3)
0003      CALL UFST(J,RAMTAB,RAMTBL)
0004      LNDM(1,1)=FIX(RAMTBL(J,1)+8.5)
0005      LNDM(1,2)=FIX(RAMTBL(J,2)+9.5)
0006      LNDM(2,1)=LNDM(1,1)+75
0007      HOLD(1)=LNDM(2,1)
0008      LNDM(2,2)=LNDM(1,2)
0009      HOLD(2)=LNDM(2,2)
0010      CALL DRAW(LNDM,IFLAG)
0011      IFLAG=0
0012      LNDM(1,1)=HOLD(1)-50
0013      LNDM(1,2)=HOLD(2)+25
0014      LNDM(2,1)=LNDM(1,1)
0015      HOLD(1)=LNDM(2,1)
0016      LNDM(2,2)=LNDM(1,2)-75
0017      HOLD(2)=LNDM(2,2)
0018      CALL DRAW(LNDM,IFLAG)
0019      LNDM(1,1)=HOLD(1)
0020      LNDM(1,2)=HOLD(2)
0021      LNDM(2,1)=LNDM(1,1)+25
0022      LNDM(2,2)=LNDM(1,2)+75
0023      CALL DRAW(LNDM,IFLAG)
0024      RETURN
0025      END

```

0001	C	SUBROUTINE TARG(J,RAMTAB,IFLAG)
0002	C	IMPLICIT INTEGER(A-C)
0003	C	DIMENSION RAMTAB(21,3), ATAR(5,5), CONST(5), RAMTBL(21,3)
0004	C	CALL OFST(J,RAMTAB,RAMTBL)
0005	C	CONST(1)=FIX(RAMTBL(J,1)+0.5)
0006	C	CONST(2)=FIX(RAMTBL(J,2)+0.5)
0007		ATAR(1,1)=CONST(1)+13
0008		ATAR(1,2)=CONST(2)
0009		ATAR(2,1)=ATAR(1,1)+30
0010		ATAR(2,2)=ATAR(1,2)
0011	C	CALL DRAW(ATAR,IFLAG)
0012	C	IFLAG=0
0013	C	ATAR(1,1)=CONST(1)
0014		ATAR(1,2)=CONST(2)+13
0015		ATAR(2,1)=ATAR(1,1)
0016		ATAR(2,2)=ATAR(1,2)+30
0017	C	CALL DRAW(ATAR,IFLAG)
0018		ATAR(1,1)=CONST(1)
0019		ATAR(1,2)=CONST(2)-13
0020		ATAR(2,1)=ATAR(1,1)
0021		ATAR(2,2)=ATAR(1,2)-30
0022	C	CALL DRAW(ATAR,IFLAG)
0023		ATAR(1,1)=CONST(1)-13
0024		ATAR(1,2)=CONST(2)
0025		ATAR(2,1)=ATAR(1,1)-30
0026		ATAR(2,2)=ATAR(1,2)
0027	C	CALL DRAW(ATAR,IFLAG)
0028	C	RETURN
0029		END

```

0001 SUBROUTINE HLD(J, RANTAB, IFLAG)
0002 IMPLICIT INTEGER(I,J)
0003 DIMENSION RANTAB(21,3), BLDH(5,5), IFMP(5), HOLD(5), RANTBL(21,3)
0004 CALL OFST(J, RANTAB, RANTBL)
0005 ICOUNT=0
0006 BLDH(1,1)=IFIX(RANTBL(J,1)+0.5)
0007 BLDH(1,2)=IFIX(RANTBL(J,2)+0.5)
0008 BLDH(2,1)=BLDH(1,1)
0009 HOLD(1)=BLDH(2,1)
0010 BLDH(2,2)=BLDH(1,2)+75
0011 HOLD(2)=BLDH(2,2)
0012 CALL DRAW(BLDH, IFLAG)
0013 IFLAG=0
0014 ICOUNT=ICOUNT+1
0015 BLDH(1,1)=HOLD(1)
0016 BLDH(1,2)=HOLD(2)
0017 BLDH(2,1)=BLDH(1,1)+23
0018 HOLD(1)=BLDH(2,1)
0019 BLDH(2,2)=BLDH(1,2)
0020 HOLD(2)=BLDH(2,2)
0021 CALL DRAW(BLDH, IFLAG)
0022 ICOUNT=ICOUNT+1
0023 IF(ICOUNT, 5) GO TO 6
0024 BLDH(1,1)=HOLD(1)
0025 BLDH(1,2)=HOLD(2)
0026 BLDH(2,1)=BLDH(1,1)
0027 BLDH(2,2)=BLDH(1,2)+75
0028 CALL DRAW(BLDH, IFLAG)
0029 ICOUNT=ICOUNT+1
0030 TEMP(1)=IFIX(RANTBL(J,1)+0.5)
0031 TEMP(2)=IFIX(RANTBL(J,2)+0.5)
0032 RANTBL(J,1)=TEMP(1)+23
0033 RANTBL(J,2)=TEMP(2)
0034 GO TO 5
0035 BLDH(1,1)=IFIX(RANTBL(J,1)+0.5)+23
0036 BLDH(1,2)=IFIX(RANTBL(J,2)+0.5)+75
0037 BLDH(2,1)=BLDH(1,1)
0038 HOLD(1)=BLDH(2,1)
0039 BLDH(2,2)=BLDH(1,2)+45
0040 HOLD(2)=BLDH(2,2)
0041 CALL DRAW(BLDH, IFLAG)

```

0042	C	HOLD(1,1)=HOLD(1)
0043		HOLD(1,2)=HOLD(2)
0044		HOLD(2,1)=HOLD(1,1)+55
0045		HOLD(1)=HOLD(2,1)
0046		HOLD(2,2)=HOLD(1,2)
0047		HOLD(2)=HOLD(2,2)
0048	C	CALL DRAW(HOLD,IFLAG)
0049	C	HOLD(1,1)=HOLD(1)
0050		HOLD(1,2)=HOLD(2)
0051		HOLD(2,1)=HOLD(1,1)
0052		HOLD(1)=HOLD(2,1)
0053		HOLD(2,2)=HOLD(1,2)+30
0054		HOLD(2)=HOLD(2,2)
0055	C	CALL DRAW(HOLD,IFLAG)
0056	C	HOLD(1,1)=TEMP(1)
0057		HOLD(1,2)=TEMP(2)
0058		HOLD(2,1)=HOLD(1)
0059		HOLD(2,2)=HOLD(2)
0060	C	CALL DRAW(HOLD,IFLAG)
0061	C	RETURN
0062		END

SUBROUTINE UFST(L,RAMTAB,RAMTHL)
COMMON/DARTA/XMIN,XMAX,YMIN,YMAX
OTC=USION RAMTAB(21,3),RAMIDI(21,3)

5444
5445
5446

ALGURITIM(UFST)

STEP 4: INITIALIZE VARIABLES WITH ARGUMENTS FROM THE SYMBOL DRAWING ROUTINE.

STEP 1: ADD OFFSET VALUE,

STEP 28 RETURN TO THE APPROPRIATE ROUTINE.

```

R=AMTBL(L,1)=Q99,*(C(AMTAB(L,1)-XMIN)/(XMAX-XMIN))
R=AMTHL(L,2)=Q10,*(C(AMTAB(L,2)-YMIN)/(YMAX-YMIN))

```

RETURN
END

5444
5445

1947
1948

SUBROUTINE FLAG(LCOUNT,RAMTAB)
 DIMENSION RAMTAB(21,3)

THIS ROUTINE HAS THE SAME ALGORITHM AS THE
 FIRST FLAG.

L=1
 IF(L.GT. LCOUNT)GOTO 999
 J=L
 IFLAG=0
 IF(L.EQ. 1)IFLAG=1

IF(RAMTAB(L,3).NE. 95)GOTO 16
 CALL FFTORD(J,RAMTAB,IFLAG)
 IFLAG=0
 GOTO 30

IF(RAMTAB(L,3).NE. 105)GOTO 17
 CALL FFTELE(J,RAMTAB,IFLAG)
 IFLAG=0
 GOTO 30

IF(RAMTAB(L,3).NE. 115)GOTO 18
 CALL FFTPTR(J,RAMTAB,IFLAG)
 IFLAG=0
 GOTO 30

IF(RAMTAB(L,3).NE. 125)GOTO 19
 CALL FFTARG(J,RAMTAB,IFLAG)
 IFLAG=0
 GOTO 30

PRINTA,'ERROR IN FLAG'
 L=L+1

GOTO 4
 RETURN
 END

0001
 0002
 C
 C
 C
 C
 C
 4
 0003
 0004
 0005
 0006
 0007
 C
 0008
 0009
 0010
 0011
 C
 0012
 0013
 0014
 0015
 C
 0016
 0017
 0018
 0019
 C
 0020
 0021
 0022
 0023
 C
 0024
 0025
 C
 C
 999
 0026
 0027
 0028

```

C      SUBROUTINE FLAG(LCOUNT,RAMTAB)
C      DIMENSION RAMTAB(21,3)
C
C      DO TO ADDRESS OVERFLOW IN THE SYSTEM, IT WAS
C      NECESSARY TO BREAK THE FFT GENERATING
C      PROGRAM INTO TWO SEPARATE PROGRAMS. PROGRAM 1
C      IS EXACTLY THE SAME AS PROGRAM 2 EXCEPT THAT
C      THEY GENERATE FOUR DIFFERENT SYMBOLS AS
C      FOLLOWS: PROGRAM 1 GENERATES
C      FFTH,FFTBRH,FFTLND,AND FFTRLD; PROGRAM 2
C      GENERATES FFTPTR,FFTTR,FFTIR,AND FFTLE.
C
C      ALGORITHM(FLAG)
C
C      STEP 01 INITIALIZE VARIABLES WITH ARGUMENTS
C          RECEIVED FROM MAIN.
C
C      STEP 10 DO FOR ALL VALUES
C          CHECK THE FLAG (RAMTAB(L,3)) AND CALL
C          THE APPROPRIATE SYMBOL DRAWING ROUTINE.
C
C      STEP 20 RETURN TO MAIN AFTER ALL SYMBOLS HAVE BEEN
C          DRAWN.
C
C      L=1
C      IF(L .GT. LCOUNT)GOTO 999
C      J=L
C      IFLAG=0
C      IF(L .EQ. 1)IFLAG=1
C
C      IF(RAMTAB(L,3) .NE. 95)GOTO 16
C      CALL FFTDRB(J,RAMTAB,IFLAG)
C      IFLAG=0
C      GOTO 30
C
C      IF(RAMTAB(L,3) .NE. 105)GOTO 17
C      CALL FFTHND(J,RAMTAB,IFLAG)
C      IFLAG=0
C      GOTO 30
C
C      IF(RAMTAB(L,3) .NE. 115)GOTO 18
C      CALL FFTLND(J,RAMTAB,IFLAG)
C      IFLAG=0
C      GOTO 30
C
C      IF(RAMTAB(L,3) .NE. 125)GOTO 19
C      CALL FFTBLD(J,RAMTAB,IFLAG)
C      IFLAG=0
C      GOTO 30
C
C      PRINT*, 'ERROR IN FLAG'
C      L=L+1
C
C      STOP

```

0026
0027
0028

C 999

GO TO 4
RE FWHN
END

```

0001 SUBROUTINE SCAL(MAT,IMAT)
0002 DIMENSION MAT(128,2),IMAT(128,2)
0003 XMIN=MAT(1,1)
0004 XMAX=XMIN
0005 YMIN=MAT(1,2)
0006 YMAX=YMIN
0007 DO 10 I=1,128
0008   IF(MAT(I,1) .LT. XMIN)XMIN=MAT(I,1)
0009   IF(MAT(I,1) .GT. XMAX)XMAX=MAT(I,1)
0010   IF(MAT(I,2) .LT. YMIN)YMIN=MAT(I,2)
0011   IF(MAT(I,2) .GT. YMAX)YMAX=MAT(I,2)
0012 CONTINUE
0013 DO 20 I=1,128
0014   IMAT(I,1)=IFIX((MAT(I,1)-XMIN)/(XMAX-XMIN))*4096.5)
0015   IMAT(I,2)=IFIX((MAT(I,2)-YMIN)/(YMAX-YMIN))*6450.5)
0016 CONTINUE
0017 RETURN
0018 END

```

14

20

```

0001 SUBROUTINE FFTW(J,RAMTAR,IFLAG)
0002 DIMENSION IMAT(128,2),NM(2),WORK(56),RMAT(128,2)
0003 NIMHSION IMAT(128,2),X(128),Y(128),IHM(5,5),MAMIBL(21,3)
0004 COMPLEX DATA(128,2)
0005 DATA IH/128,2/
0006 IPTS=128
0007 ILEG=128/3
0008 RNTVL=0,9/(IPTS-1)
0009
0010 L=J
0011 CALL OFST(L,RAMTAR,RAMTBL)
0012 Y(I)=RAMTH(J,2)
0013 X(I)=RAMTHL(J,2)
0014 DO 10 I=2,ILEG
0015   X(I)=X(I-1)+RNTVL
0016   Y(I)=Y(I-1)
0017 CONTINUE
0018 KK=IPTS-ILEG-1
0019 K=ILEG+1
0020 X(K)=X(I)
0021 Y(K)=Y(ILEG)+RNTVL
0022 DO 20 I=K,KK
0023   X(I)=X(I-1)+RNTVL
0024   Y(I)=Y(I-1)
0025 CONTINUE
0026 K=KK+1
0027 Y(K)=Y(KK)+RNTVL
0028 X(K)=X(I)
0029 DO 30 I=K,IPTS
0030   X(I)=X(I-1)+RNTVL
0031   Y(I)=Y(I-1)
0032 CONTINUE
0033 DO 40 J=1,128
0034   DATA(J,1)=CMPLX(X(J),0,0)
0035   DATA(J,2)=CMPLX(Y(J),0,0)
0036 CONTINUE
0037 CALL FOURT(DATA,NM,2,1,0,WORK)
0038 CALL ZERO(DATA)
0039 CALL FOURT(DATA,NM,2,-1,1,WORK)
0040 DO 50 I=1,128
0041   RMAT(I,1)=REAL(DATA(I,1))
0042   RMAT(I,2)=REAL(DATA(I,2))
0043 CONTINUE
0044 CALL SCALE(RMAT,IMAT)
0045 DO 60 I=2,128
0046   IHM(I,1)=IMAT(I-1,1)
0047   IHM(I,2)=IMAT(I-1,2)
0048   IHM(2,1)=IMAT(1,1)
0049   IHM(2,2)=IMAT(1,2)
0050 CALL DRAW(IHM,IFLAG)
0051 IFLAG=0
0052 CONTINUE
0053 RETURN
0054 END

```

```

0001 SUBROUTINE FTILND(J,RAMTAB,IFLAG)
0002 DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
0003 DIMENSION RMAT(128,2),IMAT(128,2),ILND(5,5),NH(2),WORK(50)

0004 COMPLEX DATA(128,2)
0005 DATA NH/128,2/
0006 L=J
0007 IPTS=128
0008 ILG=128/3
0009 RNTVL=9/(IPTS-1)

0010 CALL UFST(L,RAMTAB,RAMTBL)
0011 X(1)=RAMTBL(J,1)
0012 Y(1)=RAMTBL(J,2)
0013 DO 30 I=2,ILG
0014 X(I)=X(I-1)+RNTVL
0015 Y(I)=Y(I-1)
0016 CONTINUE

0017 K=ILEG+1
0018 KN=IPTS-ILEG-1
0019 Y(K)=Y(1)-RNTVL
0020 X(K)=X(1)+RNTVL

0021 K=K+1
0022 DO 40 I=K,KN
0023 X(I)=X(I-1)-(0.5*RNTVL)
0024 Y(I)=Y(I-1)+RNTVL
0025 CONTINUE

0026 K=K+1
0027 DO 50 I=K,IPTS
0028 X(I)=X(I-1)+RNTVL
0029 Y(I)=Y(I-1)-RNTVL
0030 CONTINUE

0031 DO 70 I=1,128
0032 DATA(1,1)=CMPLX(X(1),0,0)
0033 DATA(1,2)=CMPLX(Y(1),0,0)
0034 CONTINUE
0035 CALL FOUR1(DATA,NH,2,1,0,WORK)
0036 CALL ZERU(DATA)
0037 CALL FOUR1(DATA,NH,2,-1,1,WORK)

0038 DO 80 I=1,128
0039 RMAT(I,1)=REAL(DATA(1,1))
0040 RMAT(I,2)=REAL(DATA(1,2))
0041 CONTINUE
0042 CALL SCALE(RMAT,IMAT)
0043 DO 90 I=2,128
0044 ILND(1,1)=IMAT(I-1,1)
0045 ILND(1,2)=IMAT(I-1,2)
0046 ILND(2,1)=IMAT(1,1)
0047 ILND(2,2)=IMAT(1,2)
0048 CALL DRAM(ILND,IFLAG)
0049 IFLAG=0

```

CONTINUE
RETURN
END

90

0050
0051
0052

0001	C	SYNCHOUTIME FTTHRB(J,RAMTAB,IFLAG)
0002		SIMLNSTOU RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
0003		DIMENSION RHAT(128,2),IMAT(128,2),NH(2),MURK(SH),IBRD(5,5)
0004		COMPLX DATA(128,2)
0005		DATA IM/128,2/
0006	C	I=J
0007		IPTS=128
0008		ILEG=128/6
0009	C	RNIVI=9/(IPTS-1)
0010	C	CALL OFST(L,RAMTAB,RAMTBL)
0011	C	X(1)=RAMTAB(J,1)
0012	C	Y(1)=RAMTAB(J,2)
0013	C	DO 10 I=2,ILEG
0014		X(I)=X(I-1)+HNTVL
0015		Y(I)=Y(I-1)+HNTVL
0016	10	CONTINUE
0017	C	K=ILEG+1
0018	C	KK=ILEG+(128/6)-1
0019	C	DO 20 I=K,KK
0020		X(I)=X(I-1)+HNTVL
0021		Y(I)=Y(I-1)
0022	20	CONTINUE
0023	C	K=KK+1
0024	C	KK=KK+(128/6)-1
0025	C	DO 30 I=K,KK
0026		X(I)=X(I-1)+HNTVL
0027		Y(I)=Y(I-1)-HNTVL
0028	30	CONTINUE
0029	C	K=KK+1
0030		KK=KK+(128/6)-1
0031		X(K)=X(1)
0032	C	Y(K)=Y(K)+(128/HNTVL)
0033	C	K=KK+1
0034		DO 40 I=K,KK
0035		X(I)=X(I-1)+HNTVL
0036		Y(I)=Y(I-1)-HNTVL
0037	40	CONTINUE
0038	C	K=KK+1
0039	C	KK=KK+(128/6)-1
0040	C	DO 50 I=K,KK
0041		X(I)=X(I-1)+HNTVL
0042		Y(I)=Y(I-1)
0043	50	CONTINUE

```

0044      C
0045      KERN+1
0046      DO 60 I=K,IPTS
0047      X(I)=X(I-1)+RMIVL
0048      Y(I)=Y(I-1)+RMIVL
0049      CONTINUE
0050
0051      DO 70 I=1,128
0052      DATA(I,1)=CMPLX(X(I),0,0)
0053      DATA(I,2)=CMPLX(Y(I),0,0)
0054      CONTINUE
0055      CALL FOURT(DATA,NN,2,1,0,WORK)
0056      CALL ZER0(DATA)
0057      CALL FOURT(DATA,NN,2,-1,1,WORK)
0058
0059      DO 80 I=1,128
0060      RHAT(I,1)=REAL(DATA(I,1))
0061      RHAT(I,2)=REAL(DATA(I,2))
0062      CONTINUE
0063      CALL SCALE(RHAT,IMAT)
0064
0065      DO 90 I=2,128
0066      IHRD(I,1)=IMAT(I-1,1)
0067      IHRD(I,2)=IMAT(I-1,2)
0068      IHRD(I,1)=IMAT(I,1)
0069      IHRD(I,2)=IMAT(I,2)
0070      CALL DRAW(IHRD,IFLAG)
0071      IFLAG=0
0072      CONTINUE
0073      RETURN
0074      END

```

```

0001 SUBROUTINE FFTHLD(J,KAMTAB,IFLAG)
0002 DIMENSION KAMTAB(20,5),RANTUL(20,3),X(128),Y(128)
0003 DIMENSION RHAT(128,2),IHAT(128,2),HH(2),WORK(50),IHL(5,5)
0004 COMPLEX DATA(128,2)
0005 DATA HH/128,2/
0006
0006 I=J
0007 IPTS=128
0008 ILEG=128/7
0009 RNTVL=.9/(IPTS-1)
0010
0010 CALL OFST(L,KAMTAB,KAMTHL)
0011
0011 X(1)=RANTUL(J,1)
0012 Y(1)=RANTUL(J,2)
0013
0013 DO 10 I=2,ILEG
0014 X(I)=X(I-1)
0015 Y(I)=Y(I-1)+RNTVL
0016 CONTINUE
0017
0017 K=ILEG+1
0018 KK=ILEG+(128/7)
0019
0019 DO 20 I=K,KK
0020 X(I)=X(I-1)+RNTVL
0021 Y(I)=Y(I-1)
0022 CONTINUE
0023
0023 VALX=X(KK)
0024 VALY=Y(KK)
0025 K=KK+1
0026 KK=KK+(128/7)
0027
0027 X(K)=VALX-(8*RNTVL)
0028 Y(K)=VALY
0029 K=K+1
0030
0030 DO 30 I=K,KK
0031 X(I)=X(I-1)
0032 Y(I)=Y(I-1)+RNTVL
0033 CONTINUE
0034
0034 K=KK+1
0035 KK=KK+(128/7)
0036
0036 X(K)=VALX
0037 Y(K)=VALY
0038 K=K+1
0039
0039 DO 40 I=K,KK
0040 X(I)=X(I-1)+(0.5*RNTVL)
0041 Y(I)=Y(I-1)
0042 CONTINUE

```

```

0043      K=KK+1
0044      KP=KK+(128/7)
0045      DO 50 I=K,KK
0046      X(I)=X(I-1)+(0.5*RNITVL)
0047      Y(I)=Y(I-1)
0048      CONTINUE
0049      C
0050      K=KK+1
0051      KK=KK+(128/7)
0052      DO 60 I=K,KK
0053      X(I)=X(I-1)
0054      Y(I)=Y(I-1)-(0.5*RNITVL)
0055      CONTINUE
0056      C
0057      K=KK+1
0058      DO 70 I=K,IPTS
0059      X(I)=X(I-1)-RNITVL
0060      Y(I)=Y(I-1)
0061      CONTINUE
0062      C
0063      DO 80 I=1,128
0064      DATA(1,1)=CMPLX(X(I),0,0)
0065      DATA(1,2)=CMPLX(Y(I),0,0)
0066      CONTINUE
0067      CALL FOURT(DATA,NN,2,1,0,WORK)
0068      CALL ZERO(DATA)
0069      CALL FOURT(DATA,NN,2,-1,1,WORK)
0070      CONTINUE
0071      CALL SCALE(RMAT,IMAT)
0072      C
0073      DO 100 I=2,128
0074      IBLD(1,1)=IMAT(I-1,1)
0075      IBLD(1,2)=IMAT(I-1,2)
0076      IBLD(2,1)=IMAT(I,1)
0077      IBLD(2,2)=IMAT(I,2)
0078      CALL DRAW(IBLD,IFLAG)
0079      IFLAG=0
0080      CONTINUE
0081      RETURN
0082      END

```

AD-A103 378

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/6 9/2
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS.(U)

DEC 80 M A SMART

AFIT/GCS/EE/80-14

UNCLASSIFIED

NL

0-1-2
40



END
DATE
FILMED
081
DTIC

```

0001 SUBROUTINE FFTLEQJ,RANTAU,IFLAG)
0002 DIMENSION RANTAU(21,3),RANTBL(21,3),X(128),Y(128)
0003 DIMENSION RMAT(128,2),IMAT(128,2),ILE(5,5),INC(2),WORKK(50)
0004 COMPLEX DATA(128,2)
0005 DATA NN/128,2/
0006 L=J
0007 IPTS=128
0008 ILEG=128/3
0009 RNTVL=9/(IPTS-1)
0010 CALL OFST(L,RANTAU,RANTBL)
0011
0012 X(1)=RANTBL(J,1)
0013 Y(1)=RANTBL(J,2)
0014
0015 DO 10 I=2,ILEG
0016 X(I)=X(I-1)+RNTVL
0017 Y(I)=Y(I-1)
0018 CONTINUE
0019 K=ILEG+1
0020 KK=IPTS-ILEG-1
0021
0022 X(K)=X(ILEG)+(0.5*RNTVL)
0023 Y(K)=Y(1)
0024 K=K+1
0025
0026 DO 20 I=K,KK
0027 X(I)=X(I-1)+RNTVL
0028 Y(I)=Y(I-1)
0029 CONTINUE
0030 K=KK+1
0031 X(K)=X(KK)+(0.5*RNTVL)
0032 Y(K)=Y(1)
0033 K=K+1
0034
0035 DO 30 I=K,IPTS
0036 X(I)=X(I-1)+RNTVL
0037 Y(I)=Y(I-1)
0038 CONTINUE
0039
0040 DO 40 I=1,128
0041 DATA(I,1)=CMPLX(X(I),0,0)
0042 DATA(I,2)=CMPLX(Y(I),0,0)
0043 CONTINUE
0044
0045 CALL FOURT(DATA,NN,2,1,0,WORKK)
0046 CALL ZERO(DATA)
0047 CALL FOURI(DATA,NN,2,-1,1,WORKK)
0048
0049 DO 50 I=1,128
0050 RMAT(I,1)=REAL(DATA(I,1))
0051 IMAT(I,2)=REAL(DATA(I,2))
0052 CONTINUE
0053
0054 CALL SCALE(RMAT,IMAT)

```

```

      PD 48 12,128
      ILE(1,1)=IMAT(1,1)
      ILE(1,2)=IMAT(1,2)
      ILE(2,1)=IMAT(1,1)
      ILE(2,2)=IMAT(1,2)
      CALL DRAW(IMAT,IFLAG)
      IFLAG=0
      CONTINUE
      RETURN
      END

```

C

64

```

      0046
      0047
      0048
      0049
      0050
      0051
      0052
      0053
      0054
      0055

```

```

C
C
0001 SUBROUTINE FFTPTH(J,RAMTAB,IFLAG)
0002 DIMENSION RAMTAB(21,3),RAMTBL(21,3),IPTR(5,5),X(128),Y(128)
0003 DIMENSION RIAT(128,2),IMAT(128,2),NM(2),WORK(50)
0004 COMPLEX DATA(128,2)
0005 DATA NM/128,2/
C
0006 L=J
0007 IPTS=128
0008 ILEG=128/3
0009 RNTVL=.9/(IPTS-1)
C
0010 CALL OFST(L,RAMTAB,RAMTBL)
0011 X(1)=RAMTBL(J,1)
0012 Y(1)=RAMTBL(J,2)
0013 DO 10 I=2,ILEG
0014 X(I)=X(I-1)+RNTVL
0015 Y(I)=Y(I-1)
0016 CONTINUE
10
C
0017 J=ILEG
0018 K=ILEG+1
0019 KK=IPTS-ILEG-1
C
0020 DO 20 I=K,KK
0021 X(I)=X(I-1)-(0.5*RNTVL)
0022 Y(I)=Y(I-1)+(0.5*RNTVL)
0023 CONTINUE
20
C
0024 K=KK+1
0025 Y(K)=Y(1)
0026 X(K)=X(J)
C
0027 K=K+1
0028 DO 30 I=K,IPTS
0029 X(I)=X(I-1)-(0.5*RNTVL)
0030 Y(I)=Y(I-1)+(0.5*RNTVL)
0031 CONTINUE
30
C
0032 DO 40 I=1,128
0033 DATA(I,1)=CMPLX(X(I),0,0)
0034 DATA(I,2)=CMPLX(Y(I),0,0)
0035 CONTINUE
40
C
0036 CALL FOURT(DATA,NM,2,1,0,WORK)
0037 CALL ZERO(DATA)
0038 CALL FOURT(DATA,NM,2,-1,1,WORK)
C
0039 DO 50 I=1,128
0040 RHAT(I,1)=REAL(DATA(I,1))
0041 RHAT(I,2)=REAL(DATA(I,2))
0042 CONTINUE
50
C
0043 CALL SCALE(RHAT,IMAT)
0044 DO 60 I=2,128

```

```

IPTR(1,1)=IMAT(1-1,1)
IPTR(1,2)=IMAT(1-1,2)
IPTR(2,1)=IMAT(1,1)
IPTR(2,2)=IMAT(1,2)
CALL DRAW(IPTR,IFLAG)
IFLAG=0
CONTINUE
RETURN
END

```

611

```

0045
0046
0047
0048
0049
0050
0051
0052
0053

```

```

C
0001      SUMROUTINE: FTARG(J, RANTAB, IFLAG)
0002      DIMENSION RANTAB(21,3), RANTBL(21,3), X(128), Y(128)
0003      DIMENSION RMT(128,2), IMAT(128,2), NN(2), WORK(50), IARG(5,5)

C
0004      COMPLEX DATA(128,2)
0005      DATA NN/128,2/

C
0006      L=J
0007      IPTS=128
0008      ILEG=128/4
0009      RNTVL=.9/(IPTS-1)

C
0010      CALL GFST(L, RANTAB, RANTBL)
0011      X(1)=RANTBL(J,1)+RNTVL
0012      Y(1)=RANTBL(J,2)

C
0013      DO 10 I=2, ILEG
0014      X(I)=X(I-1)+RNTVL
0015      Y(I)=Y(I-1)
0016      CONTINUE

C
0017      K=KK+1
0018      KK=1PTS-(ILEG+2)-1
0019      X(K)=X(1)-(2*RNTVL)
0020      Y(K)=Y(K-1)
0021      K=K+1

C
0022      DO 20 I=K, KK
0023      X(I)=X(I-1)+RNTVL
0024      Y(I)=Y(I-1)
0025      CONTINUE

C
0026      K=KK+1
0027      KK=1PTS-ILEG-1
0028      Y(K)=Y(1)+RNTVL
0029      X(K)=X(1)
0030      K=K+1

C
0031      DO 30 I=K, KK
0032      X(I)=X(I-1)
0033      Y(I)=Y(I-1)+RNTVL
0034      CONTINUE

C
0035      K=KK+1
0036      X(K)=X(1)
0037      Y(K)=Y(1)-RNTVL
0038      K=K+1
0039      DO 60 I=K, IPTS
0040      X(I)=X(I-1)
0041      Y(I)=Y(I-1)-RNTVL
0042      CONTINUE

```

0043	C	DO 70 I=1,128
0044		DATA(1,I)=CMPLX(X(I),0,0)
0045		DATA(1,2)=CMPLX(Y(I),0,0)
0046	70	CONTINUE
	C	
	C	
0047		CALL FOUNT(DATA,NM,2,1,0,MURK)
0048		CALL ZEND(DATA)
0049		CALL FOUNT(DATA,NM,2,-1,1,MURK)
0050		DO 80 I=1,128
0051		RHAT(I,1)=REAL(DATA(1,I))
0052		RHAT(I,2)=REAL(DATA(1,2))
0053	80	CONTINUE
0054		CALL SCALE(RHAT,IMAT)
	C	
	C	
0055		DO 90 I=2,128
0056		ITARG(I,1)=IMAT(I-1,1)
0057		ITARG(I,2)=IMAT(I-1,2)
0058		ITARG(I,1)=IMAT(I,1)
0059		ITARG(I,2)=IMAT(I,2)
0060		CALL DRAM(ITARG,IFLAG)
0061		IFLAG=0
0062		CONTINUE
0063	90	RETURN
0064		END

```

0001 SUBROUTINE FETHRD(J,RANTAB,IFLAG)
0002 DIMENSION RANTAB(21,3),NN(2),WORK(50),RMAT(128,2),X(128)
0003 DIMENSION Y(1,8),IBRID(5,5),IMAT(128,2),RANTHL(21,3)
0004 COMPLEX DATA(128,2)
0005 DATA NN/128,2/
0006 IPTS=128
0007 ILEG=IPTS/3
0008 RNTVL=(.9/(IPTS-1))
0009 I=J
0010 CALL OFST(L,RANTAB,RANTHL)
0011 X(1)=RANTHL(J,1)
0012 Y(1)=RANTHL(J,2)
0013 DO 14 I=2,ILEG
0014 X(I)=X(I-1)+RNTVL
0015 Y(I)=Y(I-1)
0016 CONTINUE
0017 KK=IPTS-ILEG-1
0018 K=ILEG+1
0019 DO 20 I=K,KK
0020 X(I)=X(I-1)+RNTVL
0021 Y(I)=Y(I-1)
0022 CONTINUE
0023 K=KK+1
0024 DO 30 I=K,IPTS
0025 X(I)=X(I-1)+RNTVL
0026 Y(I)=Y(I-1)
0027 CONTINUE
0028 DO 50 J=1,128
0029 DATA(J,1)=CMPLX(X(J),0.0)
0030 DATA(J,2)=CMPLX(Y(J),0.0)
0031 CONTINUE
0032 CALL FOURT(DATA,NN,2,1,0,WORK)
0033 CALL ZERO(DATA)
0034 CALL FOURT(DATA,NN,2,-1,1,WORK)
0035 DO 60 I=1,128
0036 RMAT(I,1)=REAL(DATA(I,1))
0037 RMAT(I,2)=REAL(DATA(I,2))
0038 CONTINUE
0039 CALL SCALE(RMAT,IMAT)
0040 DO 70 I=2,128
0041 IBRID(I,1)=IMAT(I-1,1)
0042 IBRID(I,2)=IMAT(I-1,2)
0043 IBRID(2,1)=IMAT(I,1)
0044 IBRID(2,2)=IMAT(I,2)
0045 CALL DRAW(IBRID,IFLAG)
0046 IFLAG=0
0047 CONTINUE
0048 RETURN
0049 END

```

```

SYMBOL:
GC:
MC:
EFN:
PRSPCT:

INITL PRSPCT
GLOBAL INIT
GLOBAL INWIND
GLOBAL NAV
GLOBAL OPVRIN
GLOBAL OPVRIN
GLOBAL CLVRIN
GLOBAL CLVRIN
GLOBAL WINDOW
GLOBAL FRAME
GLOBAL BUFFER
MCAL EXITSS,SP+NS
SYMBOL /SYMBOL/
WORD 2
WORD 3
WORD 4
JSR R5,INTER
JSR R5,INIT
JSR R5,OPVRIN
JSR R5,OPVRIN
JSR R5,INWIND
MOV #0,INIRAH+10
MOV #0,INIRAH+12
MOV #639,INIRAH+14
MOV #479,INIRAH+16
MOV #INIRAH,BUFFER
JSR R5,RAMTEK
MOV #PLANE,BUFFER
JSR R5,RAMTEK
MOV #0,BUF3
MOV #17,BUF3+100
MOV #368,BUF3+200
MOV #7768,BUF3+776
MOV #COLOR,BUFFER
JSR R5,RAMTEK
MOV #0,AGL+16
JSR R5,JOYSTK
JSR R5,NAV
JSR R5,WINDOW
JSR R5,FRAME
SPHNS #SYMBOL,#GC,#MC,#EFN
TST BDR
BLT ENR
LOOP
JSR R5,CLVRIN
JSR R5,CLVRIN
EXITSS
END

INITIALIZATION FOR INTERACTIVE MAP
INITIALIZATION OF WINDOW
NAVIGATION ROUTINE
OPEN VERTICAL DATA FILE FOR INPUT
OPEN HORIZONTAL DATA FILE FOR INPUT
CLOSE VERTICAL FILE FROM INPUT
CLOSE HORIZONTAL FILE FROM INPUT
UPDATE THE TUBOIDAL MEMORY WINDOW
GENERATE THE PERSPECTIVE VIEW
FRAMTK BUFFER NUMBER PARAMTER

INITIALIZE PARAMETERS FOR INTERACTIVE MAP
OPEN THE VERTICAL FILE FOR INPUT
OPEN THE HORIZONTAL FILE FOR INPUT
INITIALIZE THE TUBOIDAL MEMORY WINDOW
THESE ARE DEFAULT VALUES - THEY CAN
BE REMOVED FROM PRCON

SET PARAMETER TO INITIALIZE THE RAMTEK
INITIALIZE THE RAMTEK
DRAW THE AIRPLANE SYMBOL

RESTORE THE LOWER LEFT HAND CORNER POINTER
NAVIGATE TO NEW POSITION
UPDATE THE TUBOIDAL MEMORY WINDOW
CREATE THE PERSPECTIVE VIEW

LOOP FOREVER
CLOSE THE VERTICAL FILE
CLOSE THE HORIZONTAL FILE
BYE-BYE!
PRSPCT

```

TEST RESULTS

The data that follows is the result of input to the thesis software. The input is data that is simulated to be like information that will be received from PRSPCT. The output statement, DIRECTIVE STATUS 1 and IOSTAT 1 indicates that the Ramtek driver successfully output information to the video display.

The first nine sets of data show that all symbols can be drawn. The second set of data shows that the program does not fail at the boundaries.

The corner coordinates are the coordinates of A, B, C, and D of the window. XMIN, XMAX, YMIN, and YMAX show the minimum and maximum boundary values. The other data is self-explanatory.

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 10.50 INITIAL LONGITUDE: 15.50

CORNER COORDINATES
15.50000 10.50000 15.21940 10.70060 15.50000 11.
15.78060 10.78060

XMIN,XMAX,YMIN,YMAX
15.21940 15.78060 10.50000 11.06120

THE RANTEK TABLE
15.5000 10.7517 20.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
HIGHWAY HAS BEEN DRAWN

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 30.50 INITIAL LONGITUDE: 35.50

CORNER COORDINATES
35.50000 30.50000 35.21940 30.78060 35.50000 31.
35.78060 30.78060

XMIN,XMAX,YMIN,YMAX
35.21940 35.78060 30.50000 31.06120

THE RANTEK TABLE
35.5000 30.7517 40.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
BRIDGE A HAS BEEN DRAWN

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 20.50 INITIAL LONGITUDE: 25.50

CORNER COORDINATES
25.50000 20.50000 25.21940 20.78060 25.50000 21.
25.78060 20.78060

XMIN,XMAX,YMIN,YMAX
25.21940 25.78060 20.50000 21.06120

THE RANTEK TABLE
25.5000 20.7517 30.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
ELECTRICAL LINES HAVE BEEN DRAWN

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

CORNER COORDINATES

85.72060 80.72060

XMIN,XMAX,YMIN,YMAX

85.21940	85.70050	80.50000	81.05120
----------	----------	----------	----------

THE RAMTEK TABLE

85.5630 80.7517 80.0380

[illegible]

**AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS**

CORNER COORDINATES

45.78060 40.78060

XMIN,XMAX,YMIN,YMAX

45.21940	45.72360	40.50000	41.03120
----------	----------	----------	----------

THE RANTEK TABLE

45.5030	40.7517	50.0000
---------	---------	---------

DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
BRIDGE 8 HAS BEEN DRAWN
>

37,35,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 30.50 INITIAL LONGITUDE: 35.50

CORNER COORDINATES
35.50000 30.50000 35.21940 30.78060 35.50000 31.
35.78060 30.78060

XMIN,XMAX,YMIN,YMAX
35.21940 35.78060 30.50000 31.66120

THE RANTEK TABLE
35.5000 30.7517 40.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
BRIDGE A HAS BEEN DRAWN
)

60,65,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 60.50 INITIAL LONGITUDE: 65.50

CORNER COORDINATES
65.50000 60.50000 65.21940 60.78060 65.50000 61.
65.78060 60.78060

XMIN,XMAX,YMIN,YMAX
65.21940 65.78060 60.50000 61.66120

THE RANTEK TABLE
65.5000 60.7517 70.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
THE LANDING STRIP HAS BEEN DRAWN
)

50,55,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 50.50 INITIAL LONGITUDE: 55.50

CORNER COORDINATES
55.50000 50.50000 55.21940 50.78060 55.50000 51.
55.78060 50.78060

XMIN,XMAX,YMIN,YMAX
55.21940 55.78060 50.50000 51.66120

THE RANTEK TABLE
55.5000 50.7517 60.0000
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
DIRECTIVE STATUS- 1
IOSTAT - 1
THE POINTER HAS BEEN DRAWN
)

70,75,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 70.50 INITIAL LONGITUDE: 75.50

CORNER COORDINATES

75.50000 70.50000 75.21940 70.70060 75.50000 71.

75.78060 70.78060

XMIN,XMAX,YMIN,YMAX

75.21940 75.70060 70.50000 71.05120

THE RAMTEK TABLE

75.5000 70.7517 80.0000

DIRECTIVE STATUS- 1

IOSTAT - 1

DIRECTIVE STATUS- 1

IOSTAT - 1

DIRECTIVE STATUS- 1

IOSTAT - 1

DIRECTIVE STATUS- 1

IOSTAT - 1

THE TARGET SYMBOL HAS BEEN DRAWN

>

85,35,0.,0.,1.104,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
45 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 35.12430 85.37685 35.50116 85.
35.37686 84.87569

XMIN,XMAX,YMIN,YMAX

35.00000 35.50116 84.87569 85.37685

85,35,0.,0.,2.209,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
90 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 35.39254 85.05820 35.45074 84.
35.05820 84.60745

XMIN,XMAX,YMIN,YMAX

35.00000 35.45074 84.60745 85.05820

85,35,0.,0.,3.313,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
135 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 35.22852 84.67565 34.80428 84.
34.67565 84.77133

XMIN,XMAX,YMIN,YMAX

34.67565 35.22852 84.44703 85.00000

85,35,0.,0.,4.414,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
180 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 34.81429 84.64931 34.46360 84.
34.64931 85.18572

XMIN,XMAX,YMIN,YMAX

34.46360 35.00000 84.64931 85.18572

85,35,0.,0.,5.515,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
225 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 34.60323 85.00084 34.61006 85.
35.00683 85.39677

XMIN,XMAX,YMIN,YMAX

34.60323 35.00683 85.00084 85.40359

85.5\5\35.0..0.,6.616.10..10.,64.64.0.,1.

AIRCRAFT HEADING AND SPEED
270 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 34.82647 85.35687 35.10335 85.

35.35687 85.17353

XMIN,XMAX,YMIN,YMAX

34.82647 35.35687 85.00000 85.53041

85.35.0.,0.,7.720.10..10.,64.64.0.,1.

AIRCRAFT HEADING AND SPEED
315 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 35.24060 85.31557 35.55617 85.

35.31557 84.75941

XMIN,XMAX,YMIN,YMAX

35.00000 35.55617 84.75941 85.31557

85.35.0.,0.,8.924.10..10.,64.64.0.,1.

AIRCRAFT HEADING AND SPEED
350 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES

35.00000 85.00000 35.39009 84.92716 35.31724 84.

34.92716 84.00000

XMIN,XMAX,YMIN,YMAX

34.92716 35.39009 84.53706 85.00000

75.70.0.,0.,1.104.10..10.,64.64.2.209,1.

AIRCRAFT HEADING AND SPEED
135 DEGREES. 1. KNOTS

INITIAL LATITUDE: 75.00 INITIAL LONGITUDE: 70.00

CORNER COORDINATES

70.00000 75.00000 70.22863 74.67565 69.90427 74.

69.67566 74.77133

XMIN,XMAX,YMIN,YMAX

69.67566 70.22863 74.44703 75.00000

VITA

Mary A. Smart was born 24 February, 1957 in Newark, Ohio. She graduated from Newark Catholic High School in Newark, Ohio in June, 1975. She attended Ohio Dominican College, Columbus, Ohio where she graduated in May, 1979 with a Bachelor of Arts degree in Mathematics and Business Administration. While attending Ohio Dominican College, where participated in the Reserve Officers Training Program at Capital University, Columbus, Ohio. She was commissioned as a second Lieutenant in the United States Air Force in May, 1979. In June, 1979 she was assigned to the Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Permanent address: 38 Madison Avenue
Newark, Ohio 43055

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/EE/SOD-14	2. GOVT ACCESSION NO. AD-A143 378	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Evaluation of Detail in Dynamic Visual Displays		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) Mary A. Smart 2Lt USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-ENA) Wright-Patterson AFB, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Information, Presentation, and Controls Group, Air Force Avionics Lab (AFWAL-AAAT-1) Wright-Patterson, AFB, OHIO 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 1 December 1980
		13. NUMBER OF PAGES 113
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; Distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
21 AUG 1981 18. SUPPLEMENTARY NOTES <i>Frederic C. Lynch</i> Frederic Lynch Major USAF Director of Public Affairs Distribution unlimited IAW-AFR 190-17 20 JUL 1981		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Airborne Electronic Terrain Mapping System (AETMS) symbolology basic symbol set visual display Fast Fourier Transform (FFT)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Airborne Electronic Terrain Mapping System (AETMS) is a mapping system designed to be used in aircraft. The report explains what software is necessary to provide an overlay of symbols onto the terrain map. The report also suggests a symbol and color set for the AETMS as well as a format to test the symbol set. Fast Fourier Transforms (FFT's) are a sub-set of the test symbol set.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)