<table>
<thead>
<tr>
<th>UNCLASSIFIED</th>
<th>AFIT/GCS/EE/80-14</th>
<th>NL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS.

THESIS

APIT/GCS/EE/80-14

Mary A. Smart
2LT USAF

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.
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.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>II. Software</td>
<td>4</td>
</tr>
<tr>
<td>III. Literature Review</td>
<td>19</td>
</tr>
<tr>
<td>IV. Symbol and Color Set</td>
<td>25</td>
</tr>
<tr>
<td>V. Methodology</td>
<td>28</td>
</tr>
<tr>
<td>VI. Conclusions</td>
<td>31</td>
</tr>
<tr>
<td>VII. Recommendations</td>
<td>32</td>
</tr>
<tr>
<td>Bibliography</td>
<td>33</td>
</tr>
<tr>
<td>Appendix A: Software Flowchart</td>
<td>34</td>
</tr>
<tr>
<td>Appendix B: Variable and Table Definitions</td>
<td>40</td>
</tr>
<tr>
<td>Appendix C: Equations</td>
<td>43</td>
</tr>
<tr>
<td>Appendix D: Symbol Set</td>
<td>44</td>
</tr>
<tr>
<td>Appendix E: Experiment Instructions and Data Forms</td>
<td>45</td>
</tr>
<tr>
<td>Appendix G: Software Listing</td>
<td>51</td>
</tr>
<tr>
<td>Vita</td>
<td>102</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Data Base Modeling Technique</td>
<td>5</td>
</tr>
<tr>
<td>2 AETMS Configuration</td>
<td>6</td>
</tr>
<tr>
<td>3 Hierarchy of SYMBOL with its sub-modules</td>
<td>8</td>
</tr>
<tr>
<td>4 Relationship of Common areas with SYMBOL</td>
<td>9</td>
</tr>
<tr>
<td>5 Rotated Display</td>
<td>9</td>
</tr>
<tr>
<td>6 CHECK2 Display</td>
<td>9</td>
</tr>
<tr>
<td>7 Rotated Display with Parameters</td>
<td>9</td>
</tr>
<tr>
<td>8 Flow Chart</td>
<td>35</td>
</tr>
<tr>
<td>9 Flow Chart</td>
<td>36</td>
</tr>
<tr>
<td>10 Flow Chart</td>
<td>37</td>
</tr>
<tr>
<td>11 Flow Chart</td>
<td>38</td>
</tr>
<tr>
<td>12 Flow Chart</td>
<td>39</td>
</tr>
<tr>
<td>13 Symbols</td>
<td>45</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Arguments passed from PRSPCT to SYMBOL</td>
<td>11</td>
</tr>
<tr>
<td>II. Symbol Set</td>
<td>26</td>
</tr>
<tr>
<td>III. Color Set</td>
<td>27</td>
</tr>
<tr>
<td>IV. PRSPCT Variables</td>
<td>40</td>
</tr>
<tr>
<td>V. SYMBOL Variables</td>
<td>41</td>
</tr>
<tr>
<td>VI. SYMBOL Tables</td>
<td>41</td>
</tr>
<tr>
<td>VII. Common Areas</td>
<td>42</td>
</tr>
</tbody>
</table>
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 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 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 overlayed 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 troubleshooting 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) worldwide 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 worldwide 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-writing 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, PRSPCCT and CNTOUR. PRSPCCT 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 PRSPCCT and CNTOUR are interactive systems,
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. 8. Rotated display

Fig. 9. CHECK 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

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

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, (e.g. 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 is 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, y2 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
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.
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 in fact, 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 imput 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"
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:

\[ H = 2D \tan \left( \frac{\theta}{2} \right) \]

\( D \) = viewing distance in inches from eye to surface
\( \theta \) = 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
that it can be determined through testing whether a simple representation of an 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

<table>
<thead>
<tr>
<th>SYMBOL COLOR</th>
<th>BACKGROUND COLOR</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>Grey</td>
<td>Hostile</td>
</tr>
<tr>
<td>Green</td>
<td>Grey</td>
<td>Friendly</td>
</tr>
<tr>
<td>Yellow</td>
<td>Grey</td>
<td>Unknown</td>
</tr>
<tr>
<td>Grey</td>
<td>Red</td>
<td>Hostile</td>
</tr>
<tr>
<td>Grey</td>
<td>Green</td>
<td>Friendly</td>
</tr>
<tr>
<td>Grey</td>
<td>Yellow</td>
<td>Unknown</td>
</tr>
<tr>
<td>White</td>
<td>Black</td>
<td>Friendly</td>
</tr>
<tr>
<td>Black</td>
<td>White</td>
<td>Hostile</td>
</tr>
</tbody>
</table>

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


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.
In Figure 3, the chart shows a process flow with the following steps:

1. **Initialize Chart**
2. **Input Data**
3. **Chart**
4. **Flowchart**
5. **In Figure**
   - **In Figure**
     - **or 79 or 99 or 180 or 270**

The flowchart includes decision points and branches for further processing.
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.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAU</td>
<td>i/128</td>
</tr>
<tr>
<td>B</td>
<td>1 BAU</td>
</tr>
<tr>
<td>A</td>
<td>aircraft azimuth</td>
</tr>
<tr>
<td>Patch</td>
<td>a digital description of a small area of terrain in memory</td>
</tr>
<tr>
<td>DH</td>
<td>display horizon (5.52)</td>
</tr>
<tr>
<td>DA</td>
<td>change in azimuth over one time slice</td>
</tr>
<tr>
<td>D</td>
<td>aircraft speed in knots</td>
</tr>
<tr>
<td>IWY</td>
<td></td>
</tr>
<tr>
<td>IWX</td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td></td>
</tr>
<tr>
<td>IY</td>
<td></td>
</tr>
<tr>
<td>JX</td>
<td></td>
</tr>
<tr>
<td>JY</td>
<td></td>
</tr>
<tr>
<td>KX</td>
<td>Corner coordinates of the window</td>
</tr>
<tr>
<td>KY</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE V.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSX</td>
<td>Coordinate conversion factors</td>
</tr>
<tr>
<td>POSY</td>
<td></td>
</tr>
<tr>
<td>CONS</td>
<td>sine and cosine factors</td>
</tr>
<tr>
<td>CONC</td>
<td></td>
</tr>
<tr>
<td>AX,AY</td>
<td>Corner coordinate's (window)</td>
</tr>
<tr>
<td>BX,BY</td>
<td></td>
</tr>
<tr>
<td>CX,CY</td>
<td></td>
</tr>
<tr>
<td>DPX,DPY</td>
<td></td>
</tr>
<tr>
<td>X;YMIN</td>
<td>boundary values</td>
</tr>
<tr>
<td>X;YMAX</td>
<td></td>
</tr>
<tr>
<td>SAB</td>
<td></td>
</tr>
<tr>
<td>SBC</td>
<td></td>
</tr>
<tr>
<td>SAD</td>
<td>Edge (window) slopes</td>
</tr>
<tr>
<td>SDC</td>
<td></td>
</tr>
<tr>
<td>LCOUNT</td>
<td>Number of symbols in the window</td>
</tr>
</tbody>
</table>

### TABLE VI.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAMTAB</td>
<td>Table of values within the bounds</td>
</tr>
<tr>
<td>SYMTAB</td>
<td>Latitude and Longitude values in patch coordinates</td>
</tr>
<tr>
<td>TEMTAB</td>
<td>Temporary table</td>
</tr>
<tr>
<td>POSTAB</td>
<td>Latitude and Longitude values in real world coordinates</td>
</tr>
<tr>
<td>NAME</td>
<td>OWNERS</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>DTABLE</td>
<td>BLOCK DATA, CONVRT</td>
</tr>
<tr>
<td>DAREA</td>
<td>SYMBOL, OFST</td>
</tr>
<tr>
<td>DBLOCK</td>
<td>SYMBOL, CHECK1, CHECK2, CONVRT</td>
</tr>
</tbody>
</table>
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).
APPENDIX E

EXPERIMENT INSTRUCTIONS AND DATA COLLECTION FORMS

This appendix contains the data collection forms necessary to compile data 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

47
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
   Understandability 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 = 0.5*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 = 0.75*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:

\[(\text{azimuth in degrees}) \times (\pi/128)\]
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 SYMHL(IAI,IAH,IAK,IAH,IAK,IAK,A,DX,UD,

AUX,XX,YY,ZZ)

COMMON/I800C/3060001,FAH(21,3),SHERE,FAILA(21,3),

FAIL=FAH(21,3),SHERE,FAILN,FAILN,FAILN,FAILN,FAILN,FAILN,FAILN,

FAILN=FAILN,FAILN,FAILN,FAILN,FAILN

DIMENSION SUM(8,8)

THIS SUBROUTINE WAS WRITTEN BY LT MARY A. SMART.

ATLASIC(EMER) FOR A THESIS SPONSORED BY THE
INFORMATION AND PRESENTATION GROUP OF THE AIR
FORC AVIONICS LAB, WRIGHT-PATTERSON AIR FORCE.

BASE, Uno, 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 AIRBURN

ELECTRONIC TIC/IN HAMMING SYSTEM (ETMS) DEVELOPED

BY THE AVIONICS LAB AND WRITTEN BY THE SYSTRAN

COMPANY. SYSTRAN DOCUMENTATION GIVES AN OVERVIEW

OF THE ETMS SOFTWARE.

THE ENTIRE MODUL CONSISTS OF 15 SUBROUTINES IN THE

FOLLOWING HIERARCHY:

LEVEL 01 SYMHL(MAIN)

LEVEL 11 CNTV1

CHECK1

CHECK2

FLAG

LEVEL 11 IN

IEMI

BRI1A

BRI1B

BRI1C

BRI1D

BRI1E

BRI1F

BRI1G

BRI1H

BRI1I

BRI1J

BRI1K

LEVEL 11 RT

OK

OF$1

ALGORITHM FOR MAIN

STEP 1: INITIALIZE ALL VARIABLES WITH ARGUMENTS

RECEIVED FROM PRSCT(AEIMS).

STEP 1: CONVERT REAL WORLD LATITUDE AND LONGITUDE

TO LATITUDE AND LONGITUDE IN PATCH COORDI-

NATES.

STEP 2: DETERMINE THE HEADING AND CALL THE APPROPRIATE

BOUNDARY CHECKING ROUTINE.

STEP 3: DETERMINE WHICH SYMBOL BELONGS AT A PARTICULAR

LATITUDE AND LONGITUDE.
```c
STEP 4: READ THE SYMBOLS.

JNB = 1
JNB <= 1 => GO TO 27
JNB => GO TO 34

DA = DA + 1
IF DA > 50 THEN DA = 1

IF DA <= 5 THEN DA = 1

DA = DA + 1
IF DA > 50 THEN DA = 1

PRINT "AIRCRAFT HEADING AND SPEED"
TYPE 75, (HEADING, 0)
FORMAT(14, DEGREES, 4, 2, "KNOTS")
PRINT ""

PUSHX = (((SIN(A) = (A^2 + B^2 - 2 * A * B * COS(A)))/DX)

PUSHY = (((COS(B) = (A^2 + B^2 - 2 * A * B * COS(A)))/DY)

CALL CONVNT(PUSHX, POSY)

CONS = ((DH = SIN(A)) / (2 * DX))

CONS = ((DH = COS(A)) / (2 * DX))

AX = AX + (FX)/60
AY = AX + (FY)/60

TYP = 70, (AY, AX)
FORMAT( "INITIAL LATITUDE: " + F6.2, " INITIAL LONGITUDE: " + F6.2)
PRINT ""

GX = AX + CONS + (FX + CONS) / 60
GQ = AX + FIX(GQ)

FX = (DX - IQX) / 60

GY = AX + CONG + (FY + CONG) / 60
GQ = AX + FIX(GY)

FY = (DY - IQY) / 60

CX = (AX + (2 * CONS) + (FX + (2 * CONS)) / 60

CX = FIX(CX)
```
FXX=(CX-I)p6

CY=(IY+(2*CUN4))+(FAI+(2*CUNIL))/6

IY=1FX(CY)

FYY=(CY-ICY)*60

UX=(10X-CUIC)+(FX-CUIC)/60

IAX=1FX(AX)

FAX=(AN-IX)*60

UY=(1GY+CONS)+(FY+CONS)/60

IY=1FX(NY)

FYY=(NY-IYY)*60

DZ=(IBX+(2*CUN3)+(FX+(2*CUN3))/60

IPX=(2*IPX)

FPX=(IPX+(2*IPX)*60

DPY=(IBY+(2*CUN4)+(FY+(2*CUN3))/60

DPY=IPY+IX(DPY)

DPT=(IPY+(2*IPY)*60

IX=(IAX+(DNUMUS(A))/(2*DY))=(NX-2)/2

IAX=(IAX+DNUMUS(A))/(2*DX))=(NX-2)/2

IX=1MX

IY=1MY

JX=1MX

JY=1MY+(NY-2)

KX=1MX+(NX-2)

KY=1MY+(NY-2)

PRINT*,'CORNER COORDINATES'

TYPN,AX,AY,BX,BY,CX,CY,DPY

PRINT*,

DO 5 J=1, JTHLN

DO 5 J=1, 1

RAMAH(M,J)=0

TENAH(M,J)=0

CONTINUE

SORT(1,1)=AX

SORT(1,2)=AX

SORT(1,3)=AX

SORT(1,4)=DPX

SORT(2,1)=AY

SORT(2,2)=AY

SORT(2,3)=AY

SORT(2,4)=DPY

XMAX=0

XMIN=0000

DQ 17 N=1, 4
begin

COMMON/DBLOCK/SYMTAB(21, 3), JTABH, HANTAB(21, 3)
COMMON/DBLOCK/LECH(21, 3), LINEA, LEQNT, CHECK

ALGORITHM(CHECK1)

STEP 01 INITIALIZE VARIABLES WITH ARGUMENTS RECEIVED FROM MAIN,

STEP 02 DO FOR ALL VALUES;
   COMPARE TABLE VALUES AGAINST MAXIMUM AND MINIMUM VALUES. IF VALUE IS #THIN
   THESE VALUES PLACE THE VALUE IN THE HANTAB TABLE.

STEP 03 COUNT THESE VALUES,

STEP 04 RETURN TO MAIN,

LCTH$=0
Kmb
J=1
OU IN [J, JTBH]
XVAL=SYMTAB(1, 1)
YVAL=SYMTAB(1, 2)
ZVAL=SYMTAB(1, 3)

C

IF(XVAL > GT, YMAX) OR (YVAL > LT, YMIN)
   OR (XVAL > GT, XMAX) OR (XVAL > LT, XMIN)
   GOTO 10
RAHTAB(J, J)=XVAL
RAHTAB(J, J)=YVAL
RAHTAB(J, J)=ZVAL
K#1
J#1
LCTH=LCTH+1
OUTICHECK
IF K #LCTH GOTO 10
CHECK=1, 0
RETURN
END
# Algorithm (Check2)

**Step 1:** Initialize variables with arguments received from main.

**Step 2:** Do for all values:
- Determine if value is in window. If value is within these bounds, place the value in a temporary table.

**Step 3:** Count these values.

**Step 4:** 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 erasable table.

**Step 5:** Count these values.

**Step 6:** Return to main.

---

```
# Example Code Snippet

IF (AY < NE, YMIN) GO TO 5
SAB = (PY = AX) / (GX = AX)
SBC = (CY = HY) / (CX = BX)
SDE = (DPY = CY) / (DPX = CX)
SAD = (AY = DPY) / (AX = DPX)

SAXY = SAB
SBXY = SBC
SCXY = SDC
SAXY = SAD

IF (DPY < NE, YMIN) GO TO 10
SAB = (DPY = AX) / (DPX = AX)
SBC = (AY = HY) / (AX = BX)
SDE = (PY = CY) / (BX = CX)
SAD = (CY = DPY) / (CX = DPX)

SAXY = SBC
SBXY = SDC
SCXY = SAD
SAXY = SBA

IF (CY < NL, YMIN) GO TO 15
SAB = (CY = DPY) / (CX = DPX)
SBC = (DPY = AX) / (DPX = AX)
```

---

**Explanation:**

- **Variables:** SAB, SBC, SDE, SAD, SAXY, SBXY, SCXY, SAD
- **Steps:** Initializing, checking within bounds, counting, and returning.
- **Conditions:** Comparisons of values against window and rotated bounds.
<table>
<thead>
<tr>
<th>DATA PUSIAN</th>
<th>15.5, 25.5, 35.5, 45.5, 55.5, 65.5, 75.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA PUSIAU</td>
<td>85, 90, 95, 100, 105, 110, 115, 120, 125</td>
</tr>
<tr>
<td></td>
<td>130, 135, 140, 145, 150, 155, 160, 165</td>
</tr>
</tbody>
</table>

This data is values of real world latitude and longitude.
AlgoRiThm(CONVrT)

STEP 01 INITIALIZE VARIABLES WITH ARGUMENTS RECEIVED FROM MAIN.

STEP 11 RETRIEVE DATA FROM BLOCK DATA.

STEP 21 DO FOR ALL VALUES AND THE OFFSET VALUE TO CONVERT VALUES TO VALUES IN PATCH COORDINATES.

STEP 31 RETURN TO MAIN.

J=1
DU SW I=1, JMLN
  X=POSTAR(1, I)
  Y=POSTAR(1, J)
  Z=POSTAR(1, Y)
  IFX=FIX(X)
  FX=(X-IFX)*60
  IY=IFY(Y)
  FY=(Y-IFY)*60
  IFX=IFX+POSX
  FX=FX+POSX
  IFY=IFY+POSY
  FY=IFY+POSY
  X=IFX+(FX/60)
  Y=IFY+(FY/60)
  SYMTAB(J,1)=X
  SYMTAB(J,2)=Y
  SYMTAB(J,3)=Z
  J=J+1
CONTINUE
RETURN
END
**ALGORITHM(FLAG)**

**STEP 01** INITIALIZE VARIABLES WITH VALUES RECEIVED FROM MAIN.

**STEP 11** DO FOR ALL VALUES IN THE HAMTAB TABLE.

**STEP 11** CHECK THE FLAG VALUE (HAMTAB(L,3)) TO FIND WHICH SYMBOL IS TO BE DRAWN.

**STEP 21** RETURN TO MAIN AFTER ALL SYMBOLS ARE DRAWN.

```plaintext
L=1
4 IF(L .GT. LCOUNT) GOTO 999
J=L
6 IF(L .EQ. 1) IFLAG=1
8 IF(1 .LE. LCOUNT) GOTO 7
9 CALL MCF(J,HAMTAB,IFLAG)
11 PRINTA("HIGHWAY HAS BEEN DRAWN")
12 GOTO 30
7 IF(1 .LT. LCOUNT) GOTO 8
13 CALL ELECT(J,HAMTAB,IFLAG)
15 IFLAG=0
17 PRINTA("ELECTRICAL LINES HAVE BEEN DRAWN")
18 GOTO 30
8 IF(1 .GT. LCOUNT) GOTO 9
19 CALL IRIDEC(J,HAMTAB,IFLAG)
21 PRINTA("BRIDGE A HAS BEEN DRAWN")
22 GOTO 30
9 IF(1 .EQ. LCOUNT) GOTO 10
23 CALL HDRIDEC(J,HAMTAB,IFLAG)
25 IFLAG=0
27 PRINTA("BRIDGE B HAS BEEN DRAWN")
28 GOTO 30
10 IF(1 .LT. LCOUNT) GOTO 11
29 CALL FTR(J,HAMTAB,IFLAG)
31 PRINTA("THE POINTER HAS BEEN DRAWN")
32 GOTO 30
11 IF(1 .GT. LCOUNT) GOTO 12
33 CALL LNDST(J,HAMTAB,IFLAG)
34 GOTO 13
```

**DIMENSION HAMTAB(21,3)**
0035      FLAG=0
0036      PRINTA,*THE LANDING STRIP HAS BEEN DRAWN*
0037      GOTO 30
0038      C
0039      IF(NHTAH(L,3) .NE. 0)GOTO 14
0040      CALL TANG(T,HHTAH,IFLAG)
0041      IFLAG=0
0042      PRINTA,*THE TARGET SYMBOL HAS BEEN DRAWN*
0043      GOTO 30
0044      C
0045      IF(NHTAH(L,3) .NE. 9)GOTO 15
0046      CALL RED(R,HHTAH,IFLAG)
0047      IFLAG=0
0048      GOTO 30
0049      C
0050      PRINTA,*ERROR IN FLAG*
0051      RETURN
0052      END
SHRMRTHE DRAK(MAT,IFLAG)

C THIS SHRMRTHE WAS ADAPTED FROM A PROGRAM
C WRITTEN BY DAVE RANGER OF SNL INC.
C PREVIOUS PROGRAM NAME: FINAL, V3.0
C 9 SEPTEMBER 1980

C IMPLICIT INTEGER(*7)
D0041 DIMENSION HAMAT(10), RAMPHR(6), IUSTAT(2), MAT(5, 5)
D0042 BYTE 10BYTE(2)
D0043 EQUVALENCE (1UMYTE(1), IUSTAT(1))
D0044 MLEN=400
D0045 RSET=*23MH
D0046 TIMP=*3AH
D0047 MSK=1
D0048 FDG=2
D0049 COP=100000
D0050 WY=*70MS
D0051 DF=1
D0052 UF=2

C CALL ABHLMCE(*XR*,0)
D0053 HAMDAT(1)=NUP
D0054 IF(IFLAG,LR,1)HAMDAT(1)=RSET
D0055 HAMDAT(2)=HYN,ON,DF,ON,DF
D0056 HAMDAT(3)=COP,ON,FDG,ON,MSK
D0057 HAMDAT(4)=3400
D0058 HAMDAT(5)=2000
D0059 HAMDAT(6)=MAT(1,1)
D0060 HAMDAT(7)=MAT(1,2)
D0061 HAMDAT(8)=MAT(2,1)
D0062 HAMDAT(9)=MAT(2,2)

C CALL GETADR(RAMPKR,HADAT)
D0063 RAMPKR(2)=w
D0064 RAMPKR(3)=w

C CALL MTO10(KLR,1,2,3, IUSTAT, RAMPKR, IER)
D0066 TYPE*, "DIRECTIVE STATUS=", IER
D0067 TYPE*, "IUSTAT =", IUBYTE(1)

C RETURN
D0069 END
THE FOLLOWING ALGORITHM IS THE SAME FOR ALL SYMBOL
DRAWING SUBROUTINES,

ALGORITHM

STEP 01 INITIALIZE VARIABLES WITH ARGUMENTS
FROM FLAG,

STEP 11 SCALE THE STARTING POINT TO VIDEO SCREEN SIZE,

STEP 21 DRAW THE PICTURE,

STEP 51 RETURN TO FLAG,

SUBROUTINE HH(J, RAMTAB, IFLAG)

IMPLICIT INTEGER(H)

DIMENSION RAMTAB(21, 3), HHM(5, 5), RAMTBL(21, 3)

ICOUNT=5

CALL OFST(L, RAMTAB, RAMTBL)

HMM(1, 1)=FIX(RAMTBL(J, 1)+0.5)
HMM(1, 2)=FIX(RAMTBL(J, 2)+0.5)
HMM(2, 1)=HMM(1, 1)+3
HMM(2, 2)=HMM(1, 2)

CALL DRAM(HHM, IFLAG)

[FLAG#]

ICOUNT=ICOUNT-1

IF(ICOUNT , LQ , 0) GOTO 250

HMM(2, 2)=RAMTBL(J, 2)+14

GOTO 10

250 RETURN

END
SHADOWC = LSELCE(J, RAMTH, IFLAG)
GAMMA  = IMPLICIT INGLREL
DIMENSION RAMTH(21, 5), LSELCE(5, 5), RAMTHL(21, 3)
ICOUNT = 0
CALL OPLST(J, RAMTH, RAMTHL)
ELEC(1, 1) = IFIX(RAMTHL(J, 1) + 0.5)
ELEC(1, 2) = RAMTHL(J, 2)
ELEC(2, 1) = LSELCE(1, 1) + 23
ELEC(2, 2) = LSELCE(1, 2)
RAMTHL(J, 1) = LSELCE(2, 1) + 36
CALL DRAW(LSELCE, IFLAG)
IFLAG = 0
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. 5) GOTO 999
GOTO 4
999 RETURN
END
C
C
C

C

SUBROUTINE RP4DA(J,RPAIH,IFLAG)
IMPLICIT INTEGER(K)
DIMENSION RPAIH(21,3),RQDA(5,5),HOLD(5),RPAIH(21,3)
C
CALL UFST(J,RPAIH,RPAIH)
C
RQDA(1,1)=FIX(RPAIH(J,1)+0.5)
RQDA(1,2)=FIX(RPAIH(J,2)+0.5)
RQDA(1,1)=RQDA(1,1)+23
HOLD(1)=RQDA(2,1)
HOLD(2)=RQDA(1,2)+23
HOLD(2)=HOLD(2)+23
C
CALL DRAM(RQDA,IFLAG)
C
IFLAG=0
C
RQDA(1,1)=HOLD(1)
RQDA(1,2)=HOLD(2)
RQDA(2,1)=RQDA(1,1)+35
RQDA(2,1)=RQDA(2,1)
RQDA(2,2)=RQDA(1,2)
HOLD(2)=RQDA(2,2)
C
CALL DRAM(RQDA,IFLAG)
C
RQDA(1,1)=HOLD(1)
RQDA(1,2)=HOLD(2)
RQDA(2,1)=RQDA(1,1)+23
RQDA(2,2)=RQDA(1,2)+23
C
CALL DRAM(RQDA,IFLAG)
RETURN
END
SUBROUTINE DRAW(J, KAMTAB, IFLAG)
IMPLICIT INTEGER(8)
DIMENSION KAMTAB(21, 5), BRD(5, 5), HOLD(5), RAMTH(21, 5)

CALL UST(J, KAMTAB, KAMTH)

BRD(1, 1) = IFIX(RAMTHL(J, 1) + 5)
BRD(1, 2) = IFIX(RAMTHL(J, 2) + 0, 5)
HOLD(1) = BRD(1, 1) + 3
HOLD(2) = BRD(2, 2) + 23

CALL DRAW(BRD, IFLAG)

IFLAG = 0

BRD(1, 1) = HOLD(1)
BRD(1, 2) = HOLD(2)
BRD(2, 1) = HOLD(1) + 35
HOLD(1) = BRD(2, 1)
HOLD(2) = BRD(2, 2)

CALL DRAW(BRD, IFLAG)

UPD(1, 1) = HOLD(1)
UPD(1, 2) = HOLD(2)
HOLD(2) = BRD(1, 1) + 23
BRD(2, 2) = BRD(1, 2) - 23

CALL DRAW(BRD, IFLAG)

BRD(1, 1) = IFIX(RAMTHL(J, 1) + 0, 5)
BRD(1, 2) = IFIX(RAMTHL(J, 2) + 0, 5) + 69
BRD(2, 1) = BRD(1, 1) + 23
HOLD(1) = BRD(2, 1)
HOLD(2) = BRD(2, 2) + 23

CALL DRAW(BRD, IFLAG)

BRD(1, 1) = HOLD(1)
BRD(1, 2) = HOLD(2)
BRD(2, 1) = HOLD(1) + 35
HOLD(1) = BRD(2, 1)
HOLD(2) = BRD(2, 2)

CALL DRAW(BRD, IFLAG)

BRD(1, 1) = HOLD(1)
BRD(1, 2) = HOLD(2)
SUBROUTINE PTK(J, RMTAH, IFLAG)
IMPLICIT INTRIN(IC)
DIMENSION RMTAH(21,3), PTKH(5), HOLD(5), RMTIH(21,3)
C
CALL OFST(J, RMTAH, RMTIH)

PTKH(1,1) = IFIX(RMTIH(J,1) + H, 5)
PTKH(1,2) = IFIX(RMTIH(J,2) + H, 5)
PTKH(2,1) = PTKH(1,1) + 150
HOLD(1) = PTKH(2,1)
PTKH(2,2) = PTKH(2,2)
HOLD(2) = PTKH(2,2)

CALL DHAH(PTKH, IFLAG)

IFLAG =

PTKH(1,1) = HOLD(1)
PTKH(1,2) = HOLD(2)
PTKH(2,1) = PTKH(1,1) + 43
HOLD(1) = PTKH(2,1)
PTKH(2,2) = PTKH(1,2) + 43
HOLD(2) = PTKH(2,2)

CALL DHAH(PTKH, IFLAG)

PTKH(1,1) = HOLD(1)
PTKH(1,2) = HOLD(2) + 86
PTKH(2,1) = PTKH(1,1) + 43
PTKH(2,2) = PTKH(1,2) + 43

CALL DHAH(PTKH, IFLAG)

RETURN
END
SHRDHTIL LNDST(J,HMTAB,IFLAG)
DIMENSION HMTAB(21,3),LNDM(5,5),HOLD(5),HMTBL(21,3)
CALL UFSX(J,HMTAB,HMTBL)
LNDM(1,1)=IFIX(HMTBL(J,1)+0.5)
LNDM(1,2)=IFIX(HMTBL(J,2)+0.5)
LNDM(2,1)=LNDM(1,1)+75
HOLD(1)=LNDM(2,1)
LNDM(2,2)=LNDM(1,2)
HOLD(2)=LNDM(2,2)
CALL DRAW(LNDM,IFLAG)
IFLAG=B
LNDM(1,1)=HOLD(1)+50
LNDM(1,2)=HOLD(2)+25
LNDM(2,1)=LNDM(1,1)
HOLD(1)=LNDM(2,1)
LNDM(2,2)=LNDM(1,2)+75
HOLD(2)=LNDM(2,2)
CALL DRAW(LNDM,IFLAG)
LNDM(1,1)=HOLD(1)
LNDM(1,2)=HOLD(2)
LNDM(2,1)=LNDM(1,1)+25
LNDM(2,2)=LNDM(1,2)+75
CALL DRAW(LNDM,IFLAG)
RETURN
END
SUNSHINE INL TAHG(J, RMA, AB, IFLAG)
IMPLICIT [INTEGER(A-M]
DIMENSION RAMAH(3), ATAR(5, 5), CONST(J, RAMBL(2, 3)
CALL OFST(J, RMA, ATAR, RAMBL)
CONST(J) = IFIX(RAMFL(J, 1) + 0.5)
CONST(2) = IFIX(RAMFL(J, 2) + 0.5)
ATAR(1, 1) = CONST(1) + 13
ATAR(1, 2) = CONST(2)
ATAR(2, 1) = ATAR(1, 1) + 30
ATAR(2, 2) = ATAR(1, 2)
CALL DRAM(ATAR, IFLAG)
IFLAG = 0
ATAR(1, 1) = CONST(1)
ATAR(1, 2) = CONST(2) + 13
ATAR(2, 1) = ATAR(1, 1)
ATAR(2, 2) = ATAR(1, 2) + 30
CALL DRAM(ATAR, IFLAG)
ATAR(1, 1) = CONST(1)
ATAR(1, 2) = CONST(2) + 13
ATAR(2, 1) = ATAR(1, 1)
ATAR(2, 2) = ATAR(1, 2) + 30
CALL DRAM(ATAR, IFLAG)
ATAR(1, 1) = CONST(1) + 15
ATAR(1, 2) = CONST(2)
ATAR(2, 1) = ATAR(1, 1) + 30
ATAR(2, 2) = ATAR(1, 2)
CALL DRAM(ATAR, IFLAG)
RETURN
END
SUBROUTINE HLDM(J, RAMTAB, IFLAG)

IMPLICIT INTEGER(N)

DIMENSION RAMTAB(21,3), HLM(5,5), TEMP(5), HOLD(5), RAMBL(21,3)

CALL OFST(J, RAMTAB, RAMTH)

COUNT=0

DO 5
  TEMP(1)=FIX(RAMTH(J,1)+0.5)
  BLDM(1,1)=HLM(1,1)
  BLDM(1,2)=HLM(2,1)
  HOLD(1)=BLDM(1,1)
  BLDM(1,2)=BLDM(2,1)
  BLDM(2,2)=BLDM(1,2)+75
  HOLD(2)=BLDM(2,2)
  CALL DRAM(BLDM, IFLAG)

IFLAG=0

COUNT=COUNT+1

DO 15
  BLDM(1,1)=HOLD(1)
  BLDM(1,2)=HOLD(2)
  BLDM(2,1)=HLM(1,1)+23
  HOLD(1)=BLDM(2,1)
  BLDM(2,2)=BLDM(1,2)
  HOLD(2)=BLDM(2,2)
  CALL DRAM(BLDM, IFLAG)

COUNT=COUNT+1

IF (COUNT .GE. 5) GOTO 6

DO 25
  BLDM(1,1)=HOLD(1)
  BLDM(1,2)=HOLD(2)
  BLDM(2,1)=HLM(1,1)
  BLDM(2,2)=BLDM(1,2)+75
  CALL DRAM(BLDM, IFLAG)

COUNT=COUNT+1

DO 30
  TEMP(1)=FIX(RAMTH(J,1)+0.5)
  TEMP(2)=FIX(RAMTH(J,2)+0.5)
  RAMTH(J,1)=EXP(1)+23
  RAMTH(J,2)=EXP(2)
  GOTO 5

DO 56
  BLDM(1,1)=FIX(RAMTH(J,1)+2.5)+23
  BLDM(1,2)=FIX(RAMTH(J,2)+0.5)+75
  BLDM(2,1)=BLDM(1,1)
  HOLD(1)=BLDM(2,1)
  BLDM(2,2)=BLDM(1,2)+75
  HOLD(2)=BLDM(2,2)
  CALL DRAM(BLDM, IFLAG)
C

1

C

CALL DRAW(HLDM,IFLAG)

C

CALL DRAW(HLDM,IFLAG)

C

CALL DRAW(HLDM,IFLAG)

C

RETURN

END
SUBROUTINE GFST(L,RMTAB,KAMTHL)
COMMON/DARK/L,XMIN,XMAX,YMIN,YMAX
LIST/STICesar/KAMTAB(21,3),KAMTH(21,3)

ALGORITHM(GFST)

STEP 1: INITIALIZE VARIABLES WITH ARGUMENTS FROM
THE SYMBOL DRAWING ROUTINE.

STEP 2: ADD OFFSET VALUE.

STEP 3: RETURN TO THE APPROPRIATE ROUTINE.

1001 LAMTH(L,1)=.002**((KAMTAB(L,1)-XMIN)/(XMAX-XMIN))
1002 LAMTH(L,2)=.002**((KAMTAB(L,2)-YMIN)/(YMAX-YMIN))
1003 RETURN
1004 END
SUBROUTINE FLAG(LCOUNT, RAMTAB)
DIMENSION RAMTAB(21, 3)

C
C THIS ROUTINE HAS THE SAME ALGORITHM AS THE
C FIRST FLAG.
C
L#1
IF(L, CT, LCOUNT)GOTO 999

JML
IFLAG=0

IF(L, EQ, 1)IFLAG=1

IF(RAMTAB(L, 3), NE, 95)GOTO 16

CALL FFT88(J, RAMTAB, IFLAG)

IFLAG=0

GOTO 30

IF(RAMTAB(L, 3), NE, 105)GOTO 17

CALL FFT88(E, RAMTAB, IFLAG)

IFLAG=0

GOTO 30

IF(RAMTAB(L, 3), NE, 115)GOTO 18

CALL FFT88(J, RAMTAB, IFLAG)

IFLAG=0

GOTO 30

IF(RAMTAB(L, 3), NE, 125)GOTO 19

CALL FFT88(G, RAMTAB, IFLAG)

IFLAG=0

GOTO 30

PRINT *, "ERROR IN FLAG" 

GOTO 4

RETURN

END
SUBROUTINE flag(count, ramtab)

DIMENSION ramtab(21,3)

C DUE 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 FFTIM, FFBRR, FFTLM, AND FFTELE. PROGRAM 2
C GENERATES FFTPI, FFTTA, FFTTD, AND FFTTE.

ALGORITHM(FLAG)

STEP 01 INITIALIZE VARIABLES WITH ARGUMENTS
RECEIVED FROM MAIN.

STEP 02 DO FOR ALL VALUES!
CHECK THE FLAG (RAMTAB(L,3)) AND CALL
THE APPROPRIATE SYMBOL DRAWING ROUTINE.

STEP 03 RETURN TO MAIN AFTER ALL SYMBOLS HAVE BEEN
DRAWN.

L=1
4 IF(L .GT. LCOUNT)GOTO 999
J=L
1FLAG=0
2FLAG=1
C IF(HAMTAB(L,1) .NE. 95)GOTO 16
CALL FFTBRR(J, RAMTAB, IFLAG)
GOTO 30
C IF(HAMTAB(L,1) .NE. 105)GOTO 17
CALL FFTIM(J, RAMTAB, IFLAG)
IFLAG=0
GOTO 30
C IF(HAMTAB(L,1) .NE. 115)GOTO 18
CALL FFTIND(J, RAMTAB, IFLAG)
GOTO 30
C IF(HAMTAB(L,1) .NE. 125)GOTO 19
CALL FFTBD(J, RAMTAB, IFLAG)
IFLAG=0
GOTO 30
C PRINT *, "ERROR IN FLAG"
L=L+1
SUBROUTINE FTILND(J,RA,RA1,IFL)

DIMENSION RA1(21,3),RA(21,3),X(128),Y(128)

COMPLEX DATA(128,2)

DATA NN/128/2,

LJ=J

IPTS=128
ILG=128/5
RNTVL=9/(IPTS=1)

CALL UFB(L,RA,RA1)

X(J)=RA(J,J)

Y(J)=RA(J,J,2)

DO 50 I=2,ILG

X(I)=X(I-1)+RNTVL

Y(I)=Y(I-1)

50 CONTINUE

K=ILG+1

KN=IPTS-ILG-1

Y(K)=Y(1)-RNTVL

X(K)=X(1)+RNTVL

K=K+1

DO 55 I=K,KK

X(I)=X(I-1)-(0.5*RNTVL)

Y(I)=Y(I-1)+RNTVL

55 CONTINUE

K=KK+1

DO 50 I=K,K,1

X(I)=X(I-1)+RNTVL

Y(I)=Y(I-1)-RNTVL

50 CONTINUE

DO 70 I=1,120

DATA(1,1)=COMPLEX(X(I),0,0)

DATA(1,2)=COMPLEX(Y(I),0,0)

70 CONTINUE

CALL FOUR1(DA,NN,2,1,8,WORK)

CALL ZERU(DA)

CALL FOUR2(DA,NN,2,1,1,WORK)

DO 84 I=1,128

NM(1,1)=REAL(DA(1,1))

NM(1,2)=REAL(DA(1,2))

84 CONTINUE

CALL SCALE(NM,IMAT)

DO 90 I=2,128

ILM(D1,1)=IMAT(I-1,1)

ILM(D1,2)=IMAT(I-1,2)

90 CONTINUE

CALL DRAW(ILND,IFL)

IFLAG

C
S:HWIVIUL FIBM3(J, RMAHAB, IFLAG)
DIMNSTH RMAHAB(21, 3), RMAHBL(21, 3), X(128), Y(128)
DIMNSH1 RMAH1(128, 2), RMAH2(128, 2), RHHA(2), RMAH5(IW), IBRD(5, 5)
DATA RMAH(128, 2)
C
I = J
JPTS=2M
ILEG=28/6
RN(V1)=9/(IPTS=1)
C
CALL OFST(L, RMAHAB, RMAHBL)
C
X(1) = RMAHAB(J1, 1)
Y(1) = RMAHAB(J1, 2)
C
DO 10 I=2, ILLG
   X(I) = X(I-1) + RNTVL
   Y(I) = Y(I-1) + RNTVL
   CONTINUE
   K = ILEG + 1
   KK = ILLG + (128/6) + 1
   DO 20 I = KK, KK
      X(I) = X(I-1) + RNTVL
      Y(I) = Y(I-1) + RNTVL
      CONTINUE
      K = KK + 1
      KK = KK + (128/6) + 1
      DO 30 I = KK, KK
         X(I) = X(I-1) + RNTVL
         Y(I) = Y(I-1) + RNTVL
         CONTINUE
         K = KK + 1
         KK = KK + (128/6) + 1
         X(K) = X(1)
         Y(K) = Y(K) + (42 * RNTVL)
         K = K + 1
         DO 40 I = K, K
            X(I) = X(I-1) + RNTVL
            Y(I) = Y(I-1) + RNTVL
            CONTINUE
            K = KK + 1
            KK = KK + (128/6) + 1
            DO 50 I = K, K
               X(I) = X(I-1) + RNTVL
               Y(I) = Y(I-1) + RNTVL
               CONTINUE
               K = KK + 1
               KK = KK + (128/6) + 1
               DO 60 I = K, K
                  X(I) = X(I-1) + RNTVL
                  Y(I) = Y(I-1) + RNTVL
                  CONTINUE
C
K=K+1
DO 60 I=K,IP+2
X(I)=X(I-1)+HR1
Y(I)=Y(I-1)+HR1
60 CONTINUE
C
DO 70 I=1,128
DATA(I,1)=CMPLX(X(I),Y(I))
DATA(I,2)=CMPLX(Y(I),X(I))
70 CONTINUE
C
CALL FOURT(DATA,IN,2,1,0,WORK)
CALL ZERD(DATA)
CALL FOURT(DATA,IN,2,1,1,WORK)
C
DO 80 I=1,128
RMAT(I,1)=REAL(DATA(I,1))
RMAT(I,2)=REAL(DATA(I,2))
80 CONTINUE
C
CALL SCALE(RMAT,IMAT)
C
DO 90 I=2,128
IHBR(I,1)=IMAT(I-1,1)
IHBR(I,2)=IMAT(I-1,2)
90 CONTINUE
CALL DRAWH(IHBR,IFLAG)
IFLAG=0
CONTINUE
RETURN
END
DIMENSION HAT(128,2), MHT(128,2), HII(2), WORK(50), IHLB(5)

COMPLEX DATA(128,2)
DATA IH/128,2/

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

DO 10 I=2,128

X(I)=X(I-1)

Y(I)=Y(I-1)+HNTVL

10 CONTINUE

K=128+1

K=128+1

DO 20 I=K,128

X(I)=X(I-I)+HNTVL

Y(I)=Y(I-I)

20 CONTINUE

VALX=X(KK)

VALY=Y(KK)

KK=KK+1

KK=KK+1

X(K)=VALX-(8*HNTVL)

Y(K)=VALY

K=K+1

DO 30 I=K,KK

X(I)=X(I-1)

Y(I)=Y(I-1)+HNTVL

30 CONTINUE

KK=KK+1

KK=KK+1

X(K)=VALX

Y(K)=VALY

K=K+1

DO 40 I=K,KK

X(I)=X(I-1)+(5*HNTVL)

Y(I)=Y(I-1)

40 CONTINUE
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS. (U)

DEC 80  M A SMART

AFIT/GCS/EE/80-14
SINHONITH \ FILLER(J,MANTAI,IFAIL)
DIMENSION MANTAI(21,3),RATUL(21,3),X(128),Y(128)
DIMENSION MHAL(128,2),MHAI(128,2),HILE(5,5),HNY(2),MNKK(50)
COMPLEX DATA(J,2R,2)
DATA(WI/128.2/)
L=1
IP=128
ILEG=128/3
RNTVL=4/(1-IP=1)
C
CALL OFBT(L,MANTAI,RFMT)
C
X(J)*=RMTL(L,J,1)
Y(J)*=RMTL(L,J,2)
DO 1=1,ILEG
X(J)=X(J)+RNTVL
Y(J)=Y(J)
1 CONTINUE

C
K=ILEG+1
KK=IP=1-ILEG-1
C
X(K)=X(KLEG)+0.5*RNTVL
Y(K)=Y(K)

C
DO 20 I=K,KK
X(J)=X(J)+RNTVL
Y(J)=Y(J)
20 CONTINUE

C
K=KK+1
X(K)=X(KK)+0.5*RNTVL
Y(K)=Y(K)

C
DO 30 I=K,IP=1
X(J)=X(J)+RNTVL
Y(J)=Y(J)
30 CONTINUE

C
DO 40 I=1,128
DATA(J,1)=CMPLX(X(J),0,0)
DATA(J,2)=CMPLX(Y(J),0,0)
40 CONTINUE

C
CALL FORIT(DATA,NN=2,1,8,MNKK)
CALL ZER0(DATA)
CALL FORIT(DATA,NN=2,1,1,MNKK)
C
DO 50 I=1,128
RHAI(J,1)=REAL(DATA(J,1))
RHAI(J,2)=REAL(DATA(J,2))
50 CONTINUE

C
CALL SCALL(RHAI,IMAT)
C  
C  SUBROUTINE FFTPIH(J, RMAJ, IFLAG)
C  DIMENSION RMAJ(21, 3), RMTBL(21, 1), IPTR(5, 5), X(128), Y(128)
C  DIMENSION RMAT(128, 2), IMAT(128, 2), NN(2), WORK(58)
C  COMPLEX DATA(128, 2)
C  DATA NN/128, 2/
C  
C  L=I  
IPTS=128
ILEG=128/3
RMTBL/8/(IPTS=1)
C  
C  CALL OFSTL(RMAJ, RMTBL)
C  X(J)=RMTBL(J, 1)
Y(J)=RMTBL(J, 2)
DO 10 I=2, ILEG
X(I)=X(I-1)+RMTVL
Y(I)=Y(I-1)
10 CONTINUE
C  
C  J=ILEG
K=ILEG+1
KK=IPTS-ILEG-1
C  
C  DO 20 I=K, KK
X(I)=X(I-1)=(8, 5*RMTVL)
Y(I)=Y(I-1)=(8, 5*RMTVL)
20 CONTINUE
C  
C  K=KK+1
Y(K)=Y(K-1)
X(K)=X(K-1)
C  
C  K=K+1
DO 28 I=K, IPTS
X(I)=X(I-1)=(8, 5*RMTVL)
Y(I)=Y(I-1)=(8, 5*RMTVL)
28 CONTINUE
C  
C  DO 32 I=1, 128
DATA(I, 1)=COMPLEX(X(I), 0, 0)
32 CONTINUE
C  
C  DO 40 I=1, 128
DATA(I, 2)=COMPLEX(Y(I), 0, 0)
40 CONTINUE
C  
C  CALL FOURT(DATA, NN, 2, 1, 0, WORK)
C  CALL ZERRG(DATA)
C  CALL FOURT(DATA, NN, 2, 1, 1, WORK)
C  
C  DO 50 I=1, 128
RMAT(I, 1)=REAL(DATA(I, 1))
RMAT(I, 2)=REAL(DATA(I, 2))
50 CONTINUE
C  
C  CALL SCALE(RMAT, IMAT)
C  DO 60 I=2, 128
C
C SUHFDAIHH: FFTAHG(j,KAMTAB,IFLAG)
C DIMENSION KAMTAB(21,3),RAHTBL(21,3),X(128),Y(128)
C DIMENSION HHAT(128,2),HAI(128,2),MH(2),WORK(50),IAPG(5,5)
C COMPLEX DATA(128,2)
C DATA MIV/128/27
C L=J
C [PTS=128
C [LEG=128/2
C NTNL=9/(1PTS=1)
C CALL OFST(L,KAMTAB,RAHTBL)
C X(1)=RAHTBL(J,1)+NTNL
C Y(1)=RAHTBL(J,2)
C DO 10 I=2,1LEG
C X(I)=X(I-1)+NTNL
C Y(I)=Y(I-1)
C 10 CONTINUE
C K=K+1
C K=PTS/(1LEG+2)+1
C X(K)=X(1)+(2*NTNL)
C Y(K)=Y(K-1)
C K=K+1
C DO 20 I=K,KK
C X(I)=X(I-1)+NTNL
C Y(I)=Y(I-1)
C 20 CONTINUE
C K=K+1
C K=PTS-1LEG+1
C Y(K)=Y(1)+NTNL
C X(K)=X(1)
C K=K+1
C DO 30 I=K,KK
C X(I)=X(I-1)
C Y(I)=Y(I-1)+NTNL
C 30 CONTINUE
C K=KK+1
C X(K)=X(1)
C Y(K)=Y(I-1)+NTNL
C 30 CONTINUE
C
C

DO 70 I=1,128

DATA(T(1))=CMPLX(X(I),Y(I))

DATA(T(2))=CMPLX(Y(I),X(I))

C

CONTINUE

C

CALL FUMI(T(DATA,NN,2,1,0,2HUNK))

CALL ZERO(T(DATA))

CALL FUMI(T(DATA,NN,2,-1,1,2HUNK))

DO 80 J=1,128

C

RHAT(1,J)=REAL(DATAT(1,J))

RHAT(1,2)=REAL(DATAT(1,2))

C

CONTINUE

C

CALL SCALE(RHAT,IMAT)

C

DO 90 K=2,128

ITARG(1,K)=IMAT(1=1,1)

ITARG(1,2)=IMAT(1=1,2)

ITARG(2,1)=IMAT(1=1,1)

ITARG(2,2)=IMAT(1=1,2)

C

CALL DRAM(1TARG,IFLAG)

C

IFLAG=0

C

CONTINUE

C

RETURN

C

END
SUBROUTINE FTPHD(J,RMAT,IFLAG)

DIMENSION RMATAB(21,128), RMAT(128,2), X(128)
DIMENSION Y(128), IOR(128,5), IMAT(128,2), RMATAB(21,13)
COMPLEX DATA(128,2)
DATA nin/128,2/
DATA IPTS=128, ILLEP=IPTS/3
DATA WINTL=(100*(IPTS=1))

I = J
CALL OSTALL(RMATAB,RMAT)
X(1)=RMATAB(J,1)
Y(1)=RMATAB(J,2)
DO 10 I=2, ILEG
X(I)=X(I-1)+RNTVL
Y(I)=Y(I-1)
10 CONTINUE
K=IPTS+ILLEP=1
K=ILEG+1
DO 20 I=K, KK
X(I)=X(I-1)+RNTVL
Y(I)=Y(I-1)
20 CONTINUE
K=KK+1
DO 30 I=K, IPTS
X(I)=X(I-1)+RNTVL
Y(I)=Y(I-1)
30 CONTINUE
DO 50 J=1, 128
DATA(J,1)=CMPLX(X(J),0,0)
DATA(J,2)=CMPLX(Y(J),0,0)
50 CONTINUE
CALL FOURT(DATA, NN, 2, 1, 8, WORK)
CALL ZEROT(DATA)
CALL FOURT(DATA, NN, 2, -1, 1, WORK)
DO 60 I=1, 128
RMAT(1,1)=REAL(DATA(I,1))
RMAT(1,2)=REAL(DATA(I,2))
60 CONTINUE
CALL SCALE(RMAT, IMAT)
DO 70 I=2, 128
IMAT(1,1)=IMAT(1,1)
IMAT(1,2)=IMAT(1,2)
70 CONTINUE
RETURN
END
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
15.5000 10.5000 10.7000 15.5000
15.7000

XMIN, XMAX, YMIN, YMAX
15.21949 15.7000 10.5000 11.06120

THE RAMTEK TABLE
15.5000 10.7517 20.0300
DIRECTIVE STATUS+ 1
IOSTAT - 1
DIRECTIVE STATUS+ 1
IOSTAT - 1
DIRECTIVE STATUS+ 1
IOSTAT - 1
BRIDGE HAS BEEN DRAWN

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 30.50 INITIAL LONGITUDE: 35.50

CORNER COORDINATES
35.5000 33.5000 35.21940 30.78060 35.5000
35.78060

XMIN, XMAX, YMIN, YMAX
35.21940 35.78060 30.5000 31.06120

THE RAMTEK TABLE
35.5000 33.7517 40.0300
DIRECTIVE STATUS+ 1
IOSTAT - 1
DIRECTIVE STATUS+ 1
IOSTAT - 1
DIRECTIVE STATUS+ 1
IOSTAT - 1
BRIDGE HAS BEEN DRAWN

AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS

INITIAL LATITUDE: 20.50 INITIAL LONGITUDE: 25.50

CORNER COORDINATES
25.5000 20.5000 25.21940 20.78060 25.5000
25.78060

XMIN, XMAX, YMIN, YMAX
25.21940 25.78060 20.5000 21.06120

THE RAMTEK TABLE
25.5000 23.7517 30.0300
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**

**Initial Latitude:** 80.50  **Initial Longitude:** 85.50

**Corner Coordinates:**
- 85.50 80.50 55.2194 80.7236 85.50 81.6
- 85.2194 80.7236

**XMIN, XMAX, YMIN, YMAX:**
- 85.2194 80.7236 81.061

**The RAMTEK Table:**

<table>
<thead>
<tr>
<th>Directive Status</th>
<th>IOSTAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

**40.45.30...30...10...10...64.64...1.**

**Aircraft Heading and Speed**

**0 Degrees, 1 Knots**

**Initial Latitude:** 40.50  **Initial Longitude:** 45.50

**Corner Coordinates:**
- 45.50 40.50 45.2194 40.7236 45.50 41.1
- 45.7236 40.7236

**XMIN, XMAX, YMIN, YMAX:**
- 45.2194 40.50 40.50 41.631

**The RAMTEK Table:**

<table>
<thead>
<tr>
<th>Directive Status</th>
<th>IOSTAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Bridge has been drawn.
Aircraft Heading and Speed
0 Degrees. 1 Knots

Initial Latitude: 30.50  Initial Longitude: 35.50

Corner Coordinates:
35.56600 33.50000 35.21940 30.78600 35.56600 31.00000
35.70600 30.78600

XMIN, XMAX, YMIN, YMAX
35.21940 35.70600 30.56600 31.066120

The Ramtek Table
35.5000 33.7517 60.00000
Directive Status 1
IOSTAT - 1

The Pointer Has Been Drawn

Aircraft Heading and Speed
0 Degrees. 1 Knots

Initial Latitude: 60.50  Initial Longitude: 65.50

Corner Coordinates:
65.56600 60.50000 65.21940 60.78600 65.56600 61.00000
65.70600 60.78600

XMIN, XMAX, YMIN, YMAX
65.21940 65.70600 60.56600 61.066120

The Ramtek Table
65.5000 63.7617 70.00000
Directive Status 1
IOSTAT - 1

The Landing Strip Has Been Drawn

Aircraft Heading and Speed
0 Degrees. 1 Knots

Initial Latitude: 50.50  Initial Longitude: 55.50

Corner Coordinates:
55.56600 50.50000 55.21940 50.78600 55.56600 51.00000
55.70600 50.78600

XMIN, XMAX, YMIN, YMAX
55.21940 55.70600 50.56600 51.066120

The Ramtek Table
55.5000 53.7517 60.00000
Directive Status 1
IOSTAT - 1

The Pointer Has Been Drawn
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 71.78060
75.78060 70.78060

XMIN, XMAX, YMIN, YMAX
75.50000 75.21940 70.70060 70.50000 71.78060

THE RANTEK TABLE
75.50000 70.7517 50.0000

DIRECTIVE STATUS = 1
IHOSTAT = 1
DIRECTIVE STATUS = 1
IHOSTAT = 1
DIRECTIVE STATUS = 1
IHOSTAT = 1

THE TARGET SYMBOL HAS BEEN DRAWN

"
AIRCRAFT HEADING AND SPEED
180 DEGREES. 1. KNOTS
INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00
CORNER COORDINATES
35.00000 65.00000 35.22954 65.05620 35.45074 64.
35.05620 64.68745
XMIN, XMAX, YMIN, YMAX
35.66086 84.45074 85.68745 85.05620
85.35.0., 0., 2.209, 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 65.00000 35.22954 65.05620 35.45074 64.
35.05620 64.68745
XMIN, XMAX, YMIN, YMAX
35.66086 84.45074 85.68745 85.05620
35.35.0., 0., 3.313, 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 65.00000 35.22954 65.05620 35.45074 64.
35.05620 64.68745
XMIN, XMAX, YMIN, YMAX
35.66086 84.45074 85.68745 85.05620
85.35.0., 0., 4.414, 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 65.00000 35.22954 65.05620 35.45074 64.
35.05620 64.68745
XMIN, XMAX, YMIN, YMAX
35.66086 84.45074 85.68745 85.05620
85.35.0., 0., 5.515, 10., 10.,...64.64, 0., 1.
AIRCRAFT HEADING AND SPEED
0 DEGREES. 1. KNOTS
INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00
CORNER COORDINATES
35.00000 65.00000 35.22954 65.05620 35.45074 64.
35.05620 64.68745
XMIN, XMAX, YMIN, YMAX
35.66086 84.45074 85.68745 85.05620
85.35.0., 0.,...65.515, 10., 10.,...64.64, 0., 1.
AIRCRAFT HEADING AND SPEED
270 DEGREES. 1. KNOTS
INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00
CORNER COORDINATES
35.60000 85.62360 34.82647 85.35688 35.10335 85.
35.35687 85.17353
XMIN, XMAX, YMIN, YMAX
34.82647 35.35687 85.00000 85.53041

AIRCRAFT HEADING AND SPEED
315 DEGREES. 1. KNOTS
INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00
CORNER COORDINATES
35.60000 85.62360 35.24660 85.31557 35.55817 85.
35.31557 84.75941
XMIN, XMAX, YMIN, YMAX
35.60000 35.55817 84.75941 85.31557

AIRCRAFT HEADING AND SPEED
360 DEGREES. 1. KNOTS
INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00
CORNER COORDINATES
35.60000 85.62360 35.39309 84.92716 35.31724 84.
34.92716 84.66332
XMIN, XMAX, YMIN, YMAX
34.92716 35.31724 84.53786 85.00000

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
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
An Evaluation of Detail in Dynamic Visual Displays

Mary A. Smart 2Lt USAF

Air Force Institute of Technology (AFIT-ENA)
Wright-Patterson AFB, Ohio 45433

Information, Presentation, and Controls Group,
Air Force Avionics Lab (AFVAL-AAAT-1)
Wright-Patterson AFB, OH 45433

Approved for public release; Distribution unlimited.

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.