AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING. (U)
MAR 79 R K CRANE
AFGL-TR-79-0248
UNCLASSIFIED
AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING

Robert K. Crane

Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

March 1979

Final Report
15 February 1978 - 30 September 1978

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
**Title:** Automatic Weather Radar Echo Assessment and Tracking

**Authors:**
- Robert K. Crane

**Performing Organization:**
Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

**Contract or Grant Number:** F19628-78-C-0078

**Report Date:** March 1979

**Number of Pages:** 145

**Distribution Statement:** Approved for public release; distribution unlimited.

**Supplementary Notes:**

**Key Words:**
- weather radar
- tangential shear
- echo areas
- automatic cell detection
- convective cells
- automatic cell tracking
- severe storms

**Abstract:**
Previously developed algorithms for automatic radar cell detection and tracking were adapted for real-time use on the AFGL Echo Track and Significance Estimator. Additional significance estimation algorithms were developed to reduce the number of detected cells to a manageable number for display and interpretation.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>5</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>7</td>
</tr>
<tr>
<td>1.1 Program Objectives</td>
<td>7</td>
</tr>
<tr>
<td>1.2 Summary of Results</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Software Development</td>
<td>9</td>
</tr>
<tr>
<td>1.4 Organization of the Report</td>
<td>10</td>
</tr>
<tr>
<td>2. Background</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Overview of Automatic Processing Scheme</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Cell Detection</td>
<td>25</td>
</tr>
<tr>
<td>2.3 Tracking</td>
<td>26</td>
</tr>
<tr>
<td>3. Programs for the AFGL Tracking and Significance Estimator</td>
<td>29</td>
</tr>
<tr>
<td>3.1 Processing Options</td>
<td>29</td>
</tr>
<tr>
<td>3.2 Cell Detection</td>
<td>29</td>
</tr>
<tr>
<td>3.3 Tracking</td>
<td>30</td>
</tr>
<tr>
<td>4. Program Status and Recommendations</td>
<td>33</td>
</tr>
<tr>
<td>5. References</td>
<td>34</td>
</tr>
<tr>
<td>Appendix A - Operating Instructions, CSS Files, TET Files, Task Establishment Maps, Definition of Variables, List of Arrays, List of Common Blocks</td>
<td>A-1</td>
</tr>
<tr>
<td>Appendix B - Flow Charts</td>
<td>B-1</td>
</tr>
<tr>
<td>Appendix C - Real-Time Version</td>
<td>C-1</td>
</tr>
<tr>
<td>Appendix D - Post-Mission Analysis Version</td>
<td>D-1</td>
</tr>
<tr>
<td>Appendix E - Sample Output</td>
<td>E-1</td>
</tr>
<tr>
<td>Appendix F - Options to Increase: Number of Significant Cells, Number of Active Cells</td>
<td>F-1</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

The results described in this report are the culmination of a sequence of contracts with the Air Force Geophysics Laboratory: "Parameterization of Weather Radar Data for Use in the Prediction of Storm Motion and Development", F19628-76-C-0264; "Development of Techniques for Short-Range Precipitation Forecasts", F19628-77-C-0058; and the current contract F19628-78-C-0076. Data used for the refinement of the algorithms were obtained by ERT under contracts with the Bureau of Reclamation, U.S. Department of the Interior, Contract No. 14-06-D-7673, and the Federal Aviation Administration, Amendment Agreement No. 4 to the Bureau of Reclamation contract.

Mr. A. Koscielny was the computer programmer operator for the Joint Agency Doppler Technology Tests in Norman, Oklahoma. Messieurs J. Leslie and G. Gustafson provided the programming support required to prepare the Interdata 7-32 computer programs.
1. INTRODUCTION

1.1 Program Objectives

The goal of the work reported herein is the real time operation of the cell detection and tracking algorithms previously developed by Environmental Research & Technology, Inc. (ERT) for the Air Force Geophysics Laboratory (AFGL). Specifically, the work included the following tasks: (1) encoding the automatic echo assessment and forecasting algorithms developed under Contract No. F19628-77-C-0058 on the Echo Track and Significance Estimator (ETSE) Interdata Model 7-32 Computer; (2) testing the algorithms for real time operation; (3) operating the computer and writing assembly level programs on the Interdata 7-32 computer during the Joint Agency Doppler Technology tests at the National Severe Storms Laboratory, Norman, Oklahoma, March through June 1978; and (4) refining the automatic assessment and forecasting algorithms based upon a critical radar meteorological analysis of the use of the algorithms.

The cell detection and tracking algorithms were developed to automatically process weather radar data to provide real time identification of severe weather and short range (0-20 minute) forecast of regions of potential hazard to aircraft operation.

1.2 Summary of Results

The cell detection and tracking algorithms previously developed under Air Force Contract (Crane, 1978; Crane, 1977) were designed for real time operation on a medium scale computer based on our experience with an extremely limited data sample (five consecutive azimuth scans) and with the CDC 6600 computation facility at AFGL. The subsequent development of a viable real time weather radar data processing system for use by the Air Force, however, required access to a significantly larger set of weather radar data.

The initial development of the cell detection technique had been undertaken for the Federal Aviation Administration (FAA) (Crane, 1976) for use in automatic air traffic hazard detection. Their continued interest in the processing scheme and its evaluation resulted in a contract between ERT and the FAA to process several hours of radar observations made simultaneously with aircraft penetrations. ERT was also under
contract with the Bureau of Reclamation, Department of Interior (BuRec) during the period of performance of this contract for the acquisition and analysis of significant amounts of radar data for the development of precipitation augmentation strategies for the High Plains (HIPLEX). As a part of that contract, the cell detection and tracking algorithms were installed on the CYBER-74 Computer System in Denver and used to obtain statistical data on the spatial organization of precipitation production within high plains storms.

The three concurrent programs, the development of real time techniques for the Air Force, hazard detection algorithm evaluation for the FAA, and the spatial organization analysis for BuRec provided the experience with a significantly larger data base needed for the refinement of the tracking algorithms (Task 4) and the development of a viable real time processing program.

The work under this contract was organized in the four tasks listed in Section 1.1: (1) encoding the algorithms for real time operation on the Interdata 7-32 computer; (2) testing the algorithms for real time operation; (3) operating the ETSE in Norman, Oklahoma during the Joint Agency Doppler Technology tests; and (4) refining the algorithms for improved tracking and short range forecasting. ERT provided a programmer operator for the 1978 Joint Agency Doppler Technology tests in fulfillment of Task 3. The detection and tracking algorithms refined as a result of Task 4 have been coded and installed on the Interdata 7-32 computer in fulfillment of Task 1. Their description is the subject of this report. Program listings and a copy of the operating instructions are included in Appendix C and D. The algorithms have been tested in compliance with Task 2 and, as coded and operated, performed in real time.

The algorithms operate in real-time on the Interdata 7-32 computer as required. Real-time operation on the Interdata 7-32 in the manner used in non-real-time analysis, however, requires additional programs to fetch and store the raw radar data. For real-time operation at the level of performance of the previously developed non-real-time program, a new operating system is required for the Interdata 7-32 computer which utilizes the real-time interrupt capability of the machine to run the cell detection and tracking programs as background programs with the data averaging and storage programs in foreground. Such an undertaking was
beyond the scope of this contract, and a simpler program has been specified which allows operation in real-time on alternate scans; one to fetch and store the data, the second to process the data. This program, a modification to the TSE program provided by Raytheon (Boak et al., 1977), is listed in Appendix C.

1.3 Software Development

The previously developed cell detection and tracking programs were extensively modified for use on the Interdata 7-32 computer. The tracking program described in the final report of the previous contract (Crane, 1978) was completely rewritten starting with a new set of algorithms. Experience with the larger volume of radar data available from BuRec forced the program revision. Two major problems existed with the initial tracking program; excessive computer storage requirements for the large numbers of cells encountered in practice and an inherent inability of the program logic to separately establish individual smoothed track velocities for each cell. The new tracking program develops the volume cell attributes discussed in the previous contract report, establishes the existence of cell clusters, provides an estimate of cell significance, and maintains both instantaneous and smoothed velocity estimates for each cell.

The cell detection program encoded for use on the Interdata 7-32 computer is a streamlined version of the original cell detection program. The fixed contour outlines are developed not as line segments enclosing the contoured area but as azimuth strobes within the echo region in conformity with the ETSE display scheme. Attributes are not generated for the fixed contours. The cell detection subroutine operates as before. Addressing in the arrays used in the subroutine has been extensively altered to increase operating speed.

The programs can operate over a wide range of reflectivity thresholds but should be used at a relatively high reflectivity threshold, processing only data with reflectivity values above, say, 40 dBZ. This threshold was selected to reduce the number of cells being processed in the computer. The reduction in the number of cells being processed improves operating speed and matches the output requirements. Experience with storms in the high plains indicates that severe storms produce large
numbers of cells. Cell counts for a 20 dBZ reflectivity threshold are over 150 during the active period of a storm; the total number of separate cells observed during a storm often exceeds several thousand. By way of contrast, the output requirements established by the remote display system (communicated by AFGL personnel), are for not more than 12 cells at any one time. The more than an order of magnitude reduction in active cell count can best be accomplished by increasing the threshold reflectivity value.

In addition to reflectivity threshold selection, significance is established using the integrated tangential shear of the radial velocity for each cell. Tangential shear cell detection as previously coded (Crane, 1977) is not attempted in the streamlined version of the program. The tangential shear data are used to develop a shear attribute for each cell detected on the basis of reflectivity alone. The program operates by detecting all cells that occur above the processing threshold but saving for tracking and output only the 12 cells having the highest reflectivity, integrated tangential shear product. Internally, the program processes 31 volume cells but only the 12 volume cells with the highest reflectivity, shear product are output after each volume scan. The program can be modified to process and output data for more cells by changing array sizes and test limits.

The object of the work reported in this contract was to streamline the original version of the cell detection and tracking programs for real-time use on the Interdata 7-32 computer with the operating system provided by Interdata. Many of the features of the original program, such as the generation of fixed contour attributes and the independent detection of tangential shear peaks, were removed to establish the real-time program. These features may be recovered only if the original version of the program, the program operating on the AFGL CW-6000 computer, is installed on the Interdata 7-32 for non-real-time processing.

1.4 Organization of the Report

This report considers only the software developed for use on the ITS1, documentation for Tasks 1, 2 and 4. Task 3 covered the programmer computer operator for the 1978 measurements in Norman, Oklahoma. The results of that task were reported in the quarterly reports and will not be considered further.
Background material and algorithm refinement based upon results from the FAA and BuRec programs are considered in Section 2. Section 3 documents the program for the ETSF. Section 4 summarizes program status and makes recommendations for future work. Program listings, flow charts, variable definition, and operating instructions will be found in the appendices.
2. BACKGROUND

2.1 Overview of Automatic Processing Scheme

Conventional weather radars produce large amounts of data - a significant fraction of which is highly redundant. Doppler radars produce even larger amounts of data. Significant weather events may be imbedded in the mass of redundant data. It is the goal of the automatic processing scheme to extract the relevant information from the mass of data to (1) reduce the data transmission requirements for the communication of weather data obtained from a radar, (2) to screen the data prior to display to meteorologists, (3) to preprocess the data for automatic hazard detection, and (4) to prepare the data for use in objective short range forecasting.

The processing scheme is structured to use the cell detection algorithm in on-radar-site computers to perform the bulk of the data reduction. The cell data are then communicated to regional computers (or to a second program in a stand alone radar data processor) for tracking and interpretation. For a national network of weather radars, the tracking program would accomplish the task of netting different radars and developing a single, best estimate description of the current weather for use in displays, hazard detection and warning, and short range forecast.

For this contract the cell detection and tracking algorithms are operated in a single computer; the final output is track data for the 12 most significant cells. The track data include smoothed cell velocities which are used in the tracking program for data association and may be used externally for short range forecast.

The output cell and track data are for significant features in the larger mass of radar observations. Significance is defined in an ad hoc manner using cell intensity, area, vertical development, and tangential shear data. Parameters that are intuitively associated with significant events such as severe hail, severe thunderstorm turbulence, and tornadoes have been selected for the determination of a significant cell. Operational experience with the processing algorithms and a large sample of data is required before the values of the thresholds used to establish cell significance can be refined. The current algorithm has been
partially tested using aircraft penetration data. For the measurements currently available, a positive correlation has been obtained between the location of significant cells and aircraft turbulence. An example of this association is presented in Figures 1 and 2 (output obtained from the work at ERT sponsored by the FAA). The significant cells are indicated by the tightly clustered symbols for reflectivity values greater than 40 dBZ. The time marks are 6 km apart along the aircraft track. The aircraft was within a typical cell diameter of two significant cells between 1640 and 1641, a time marked by the strongest acceleration fluctuations (turbulence) during the penetration. By way of contrast, the remainder of the penetration was quiet and did not show strong acceleration fluctuations and was not in the vicinity of significant cells.

The display in Figure 1 immediately identifies the locations of the significant cells and graphically presents the essential data contained in the radar observations. In contrast, a section of a conventional contour display and of the cell display are presented in Figure 3. The essence of the data is immediately evident in the cell display. The important 45 dBZ cell that results from the strong updraft depicted between 1640 and 1641 MDT in the aircraft data is observed in the cell display but not in the contour display. This cell produced the strongest turbulence. Even with a color display, this important region would not be evident although the higher reflectivity contours in Figure 3a would stand out more vividly, in the manner of the darkened symbols in Figure 3b.

A display such as Figure 3b is readily interpreted but is not the normal weather radar display familiar to trained radar meteorologists. The cell display provides the important details but at a scale that is smaller than used by most meteorologists and since many cells may be observed at any single time, a display of all the cells may prove to be confusing. The number of cells active at any one time with reflectivities above 15 dBZ and the number of significant cells for a set of observations of showers in Kansas (output from data processed by ERT for BuRec) are depicted in Figure 4 as a function of time and volume scan sequence. The total number of cells present within a 25 to 150 km annulus of Goodland, Kansas approached 200 during the most active part
PENETRATION #2

TIME

○ 1639:00 - 1640:30
△ 1640:30 - 1642:00
□ 1642:00 - 1643:30

INTENSITY (dBZ)

△ <40
△ 40-50
▲ >50

Figure 1  Aircraft Penetration 22 July 1976 as Observed by the Grover Radar
Figure 3  Contour and Cell Data, Grover Radar Data
Figure 4  Active Cells and Total Water Production, Goodland, Kansas
10 June 1976

Water Production Rate (Kmt/hr)
of the storm. The number of significant cells was as high as 30. A display of all the cells would be very difficult to interpret. A display of only the significant cells, though readily interpreted by a computer, may still be difficult for a meteorologist to interpret. A further reduction in the number of significant cells is required. The program developed under this contract utilizes tangential shear data in addition to reflectivity information to further reduce the number of cells for display as the most significant cells. The utility of this algorithm for significant cell selection still needs verification.

The small cells detected and tracked by the algorithms developed under this and prior contracts are well behaved in time and space. The cells show vertical development, persist, and have average velocities that approximate a steering level wind. Summary statistics for the June 10, 1976 storm displayed in Figure 4 are presented in Figures 10 to 9. These data provide statistical summaries of several cell characteristics representing either the values averaged over the lifetime of a cell (average) or the peak value obtained during the cell lifetime (peak). Lifetime data are presented in Figure 10. The data in Figure 5 depict the statistics of the highest reflectivity values reached by a cell during its lifetime. The data are drawn from a sample of 900 cells whose lifetimes exceeded 10 minutes. These data show that reflectivity alone was not used as a criterion for significance. Over 77 cells had a peak reflectivity in excess of 50 dBZ while only 8 significant cells had a peak reflectivity in excess of 50 dBZ. In the processing used to obtain these data, significance was defined based upon the vertical structure and horizontal dimension of the cell as well as its reflectivity. A high reflectivity echo that was observed only at one elevation angle was not considered to be a significant cell.

The cells detected by the processing scheme have relatively small areas. The average cell area for all cells and for significant cells do not differ as indicated in Figures 6 and 7. The peak cell diameter is of the order of 3 km. The cells tend to be vertical structures. Statistically, the area of a cell at the lowest elevation angle does not differ from the area of the cell at the height at which it has a maximum reflectivity value. Note that the area at any height is defined by a contoured region 3 dB below its peak reflectivity value at that height. At another
Figure 5 Reflectivity Statistics, Goodland, Kansas
10 June 1976

PEAK REFLECTIVITY
1976 DAY 162

NUMBER OF OCCURRENCES

ALL CELLS

SIGNIFICANT CELLS

LARGEST CELL IN A CLUSTER

PEAK REFLECTIVITY (dBZ)
Figure 6  Cell Area Statistics, Goodland, Kansas - 10 June 1979

SURFACE PEAK Z
AVG. DIA. = 2.3 km 2.3 km
PEAK DIA. = 2.9 km 2.9 km

- AVERAGE SURFACE AREA
- PEAK SURFACE AREA
- AVERAGE PEAK Z AREA
- PEAK PEAK Z AREA
Figure - Significant cell area statistics, Goodland, Kansas

DAY 162 SURFACE PEAK Z

AVG. DIA. 2.3km
PEAK DIA. 3.0km
AVG. SURFACE AREA 2.2km
PEAK SURFACE AREA 3.0km
AVG. PEAK Z AREA
PEAK PEAK Z AREA
1976 DAY 162

Figure 8  Cell Velocity Statistics, Goodland, Kansas - 10 June 1976

Number of Occurrences

- All Cells
- Significant Cells

East - Average 22 Velocity (m/s)
Figure 9  Cell Velocity Statistics, Goodland, Kansas - 10 June 1976

1976
DAY 162

NUMBER
OF
OCURRENCES

1000-

100-

10-

1

ALL CELLS

SIGNIFICANT CELLS

NORTH - AVG. VELOCITY (m/s)

-20 -10 0 10 20
Figure 10  Cell Lifetime Statistics, Goodland, Kansas - 10 June 1976
height, the contour used to define its area will have a different reflectivity value.

The tracking algorithms track each cell individually. The distributions of cell velocity averaged over the cell lifetime are depicted in Figures 8 and 9. These data show no significant differences between the velocities of all the cells and of significant cells. The individual track velocities may differ from the mean (or steering level wind) velocity by as much as 4 m/s (rms in each component of the wind). Observations show that the deviations are not entirely random. Larger scale convergence and divergence patterns are evident in the cell trajectories. On a smaller scale, it is evident that cells affect each others motion. A tendency has been observed for cells to follow each other along the same track even though they may not have initially developed along the track.

The cells persist for a range of lifetimes. On average, cells with reflectivities above 15 dB last for a little over 12 minutes. Significant cells last over 30 minutes on average. These results show that ideally, a full cycle of radar observations (a volume scan) should be acquired in between four and five minutes to get more than two observations of a cell during its lifetime. Practically, a longer time, six to seven minutes, is required for the processing algorithms as implemented under this contract. Processing speed can be increased by preparing a new operating system for the Interdata 7-32 computer but this was beyond the scope of the contract. The motion of significant cells, if detected early during its development, may be extrapolated for upwards of 20 minutes before they disappear indicating that short range forecasting of cell location is feasible.

2.2 Cell Detection

The cell detection algorithms have been previously defined (Crane, 1977, 1978) and will not be detailed again. Flow charts for the processing algorithm are presented in Appendix B. Briefly, a cell is a region within a contour, a fixed number of quantization steps below a local maxima that includes no other cells. For most observations, a quantization step of 1 dB and the use of contours 3 dB below a local reflectivity maxima seems to work best. The quantization step and contour threshold
were empirically established by Crane (1976) using data from Wallops Island, Virginia. The 1 dB step and 3 dB threshold produced a detection probability better than 0.95 on these scans and a false alarm rate of less than 2 per scan. By increasing the threshold, the false alarm rate was reduced but at the expense of a lowered detection probability.

Cell detection is performed for all localized reflectivity maxima that exceed a processing threshold. In the post mission processing versions of the cell detection program, attributes are obtained for all the cells detected above the lowest threshold fixed reflectivity processing contour (fixed contour or echo region). The streamlined edition of the cell detection program provides output only for the most significant cells.

2.3 Tracking

An entirely new tracking algorithm was developed during the period covered by this contract primarily for application to the post mission processing requirements of the FAA and BuRee contracts. This program was subsequently modified for real time application under this contract. The real time version generates volume cells and volume cell tracks.

The same tracking algorithm is used to generate the 3-dimensional cell from the successive azimuth scans within a volume scan sequence and to track the volume cell in time. Each new volume cell is identified using cell data from a single azimuth scan that cannot be associated with other, previously established tracks. Association is attempted with the newly detected volume cell on each successive azimuth scan until the track is terminated. A track is terminated when no new data are obtained for a volume cell during a complete volume scan cycle.

Association is established using the location of the cell on a azimuth scan as compared with the volume cell location extrapolated to the time of the azimuth scan together with the differences in the heights of the last observed data and the current data, differences in cell areas, and differences in reflectivities. A measure of the goodness of an association between a cell and track, is established for each possible track, cell combination. The final cell track pairings are those that minimize the sum of the measures (maximize the goodness of association) for all the cells and tracks that may be possibly paired during the association.
process. The set of cells and tracks that may possibly be paired are defined as a cluster.

The volume cell track is the primary entity maintained by the tracking program and successive radar observations are associated with the expected position of the cell along its track at the time of the radar observation. The tracking algorithm was developed in this manner to allow the use of data from more than one radar system since all that is needed for the association algorithm is the location and time of the cell centroid as reported by each radar together with other attributes such as reflectivity, tangential shear, etc.

Two velocities are maintained for each cell track - an instantaneous velocity, the difference in cell centroid location divided by the time interval between observations, and a smoothed velocity obtained by low pass digital filtering of the raw velocity data. The coefficients in the filter were established by trial and error using a large data sample. The initialization of the track velocity for each cell is important due to the extrapolation process used for tracking. Experiments with the tracking program show that adequate results are obtained if a zero velocity is used for the initial velocity estimate but better results can be obtained if an estimate of the steering level wind is used for the initial tracking velocity. The program automatically updates the initial velocity estimate after processing sufficient data to establish a stable estimate. The measure of success of the tracking program was taken as the rate required for the smoothed track velocities to stabilize.

The cell detection and tracking programs were initially developed to process a large number of cells, up to 512 active cell tracks at any one time and to calculate upwards of 30 attributes for each cell, cluster, and fixed contour (echo) region. Processing this amount of data is not possible in a real time environment with storage and display limitations. The basic algorithms for cell detection and tracking have been maintained. The number of tracks to be processed has been reduced by increasing the reflectivity threshold for processing and by incorporating the tangential shear information in the decision process for saving the most important 12 to 16 cells of 30 or more cells that exceed the reflectivity processing threshold. Further storage savings have been accomplished by reducing the number of attributes for each cell.
The cell detection and tracking programs were not coded in efficient manner for operation on the ETSI Interdata 7-32 computer. Extensive program revision was performed to reduce the number of subroutine calls and to revise the addressing procedure to reduce the time required to fetch or store a variable. The result is a streamlined cell detection and tracking program that will handle a reasonable number of active cells during the time required for an azimuth scan. Specifically, over 100 cells can be detected and processed in less than 52 seconds using the Interdata 7-32 programs. The processing time can be reduced further by dynamically varying the reflectivity processing threshold to maintain fewer than say 20 detected cells but this has not been necessary.

The programs to fetch the raw radar data and prepare the data for use in the cell detection program are included in Appendix C and D. The real program is designed to permit data gathering and cell detection on alternate scans. Real-time processing using the cell detection algorithms on every azimuth scan is possible with the Interdata 7-32 computer but will not occur unless considerable effort is expended to develop a new operating system for the computer tailored to use the interrupt and background/foreground processing capabilities of the computer to provide quasi-real-time cell detection and simultaneous real-time radar data acquisition, averaging and storage. Sufficient time is available for all the programs to operate on all the data from a volume scan within the time of a volume scan but the processing of the data from the lower elevation angles will lag behind data acquisition and only catch up on the higher elevation angles.
3. PROGRAMS FOR THE AIGI TRACKING AND SIGNIFICANCE ESTIMATOR

3.1 Processing Options

Two programs exist for operation on the LSI. A post mission processing program is available called CRANL that reads previously prepared data from the disk and performs cell detection and tracking. This program is not intended for real time operation and threshold levels may be reduced to allow detection of a large number of cells. The program is intended for post mission data analysis when time is not at a premium and larger amounts of output can be handled by the user. The input data must be prepared for storage on the disk using a modification to the LSI programs generated by Raytheon. The program is called IR1 and is listed in Appendix D.

The operational program uses the same cell detection and tracking subroutines but is called from a modified version of the Raytheon provided LSI programs. These programs store the data on one a ninth scan and process the data on the next. Data from the first, third, and fifth scans are processed during the second, fourth, and sixth. During the seventh scan the displays are prepared and the programs are reinitialized for the next cycle of the seven scan sequence.

All the programs generated under this contract for use on the IF data 'A'3' computer are listed in Appendix C and D. CRANL is the post mission main program which calls CONDOR and TRACK. The output from this program is stored on disk for subsequent listing. This program requires the use of a preprocessing program PPRDSC (modified for IR1) written by Avi personnel and listed in Appendix D. The real time program includes modified versions of the LSI programs supplied by Raytheon. The modified programs are TSEMAX and the subroutines RMAP and PPRDSC. The CONDOR and TRACK subroutines are called from RMAP. These programs are presented in Appendix C.

3.2 Cell Detection

The cell detection subroutines CONDOR and RMAP were substantially modified to reduce processing time spent in addressing the data arrays. The principal modification was to change all the arrays to single dimen-
sional arrays and to explicitly perform the address calculations in the program. In this way, multiple references to the same array location would not involve the time consuming recalculation of the address for each array reference.

CONTOR was extensively modified to remove the fixed contour attribute generation algorithm. Contour data are still prepared but in the azimuth strobe format used by the TSI display program. No radial-to-radial association is required for this streamlined version of the program, significantly decreasing the length and complexity of the program.

PIAKD has been changed only to accept the modified system of addressing and to select only the 10 most significant cells for further processing. Reference to the fixed contour identification tag was also removed from PIAKD since the tags were produced in the association, attribute generation logic of the CONTOR subroutine which was removed for the Interdata version of the program.

V Tracking

The new tracking program consists of the subroutine TRACK which calls COMPAR to perform the cell to track associations. The subroutine COMPAR searches the track list and the cell list from the last scan, and finds all possible pairs for which the goodness of fit measure does not exceed a preselected threshold. The measure is given by

\[ M = 1.4 \left\| \frac{A_c}{L_c} - \frac{A_t}{L_t} \right\|^2 + \left\| \frac{H_c}{L_c} - \frac{H_t}{L_t} \right\|^2 + k \cdot \left( \frac{\left\| \left( x_c, y_c \right) - \left( x_t, y_t \right) \right\|^2}{w} \right) \]

where \( M \) is the measure, the subscripts \( c \) refer to cell and \( t \) to track and \( x \) is reflectivity, \( A \) area, \( H \) height and \( \bar{X} \) centroid locations. The weights \( k \) were set by trial and error. The current values are listed in Table 1. The best cell-track pairing has the lowest measure; pairings with a measure greater than \( M_0 \) are not allowed.

Several pairings are possible in a cluster of cells. Subroutine RESOLA selects the best set of cell, track pairs in a cluster. The attributes are updated in ATRAK which is called from COMPAR if there is no cell cluster or from RESOLA if a cluster exists. The subroutine ATRAK
is also called to store cell data in the VR array each time a track is updated. This information is used in calculating the measure \( M \). The VR array data are either from the lowest elevation angle on which the track was observed or from the last elevation angle at which it was observed. The measure used to evaluate the cell, track pairing is the minimum measure obtained using either the last or lowest elevation angle data.

At the end of a volume scan cycle, STRAK is called to calculate the attributes and to output the track data. Only 12 tracks are output from STRAK in the operational version although a maximum of 32 tracks are maintained at any one time. The list of the 16 attributes maintained for each track is given in Table 2.

**TABLE 1**

<table>
<thead>
<tr>
<th>TRACKING, WEIGHTS, MAXIMUM MEASURE AND VELOCITY FILTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_X )</td>
</tr>
<tr>
<td>( W_A )</td>
</tr>
<tr>
<td>( W_H )</td>
</tr>
<tr>
<td>( W_L )</td>
</tr>
<tr>
<td>( M_P )</td>
</tr>
</tbody>
</table>

Velocity Filter:

\[
(V)_N = a \frac{Ax}{At} + b (V)_N + c V_N
\]

\[
(V)_E = a \frac{Ax}{At} + b (V)_E + c V_E
\]

\( V \) velocity; \( N,E \) refer to Northward and Eastward \((v,x)\) components

\( Ax,Ay \) = change in position between volume scans

\( At = \) time between scans

\( \bar{v} \) = average velocity, all cells

\( a = .4 \quad b = .3 \quad c = .3 \)
<table>
<thead>
<tr>
<th>No.</th>
<th>Attribute</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Observation Time</td>
<td>(seconds from start of year)</td>
</tr>
<tr>
<td>2</td>
<td>East Location</td>
<td>(km)</td>
</tr>
<tr>
<td>3</td>
<td>North Location</td>
<td>(km)</td>
</tr>
<tr>
<td>4</td>
<td>Average Reflectivity</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>5</td>
<td>Volume</td>
<td>(km³)</td>
</tr>
<tr>
<td>6</td>
<td>Peak Reflectivity</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>7</td>
<td>Height of Reflectivity Peak</td>
<td>(km)</td>
</tr>
<tr>
<td>8</td>
<td>Reflectivity at Lowest Elevation Angle</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>9</td>
<td>Area of Cell on Lowest Elevation Angle</td>
<td>(km²)</td>
</tr>
<tr>
<td>10</td>
<td>Height of Cell at Lowest Elevation Angle</td>
<td>(km)</td>
</tr>
<tr>
<td>11</td>
<td>Reflectivity at Cell Top</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>12</td>
<td>Height of Cell Top</td>
<td>(km)</td>
</tr>
<tr>
<td>13</td>
<td>Track Identifier</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Smoothed Track Velocity - East</td>
<td>(m/s)</td>
</tr>
<tr>
<td>15</td>
<td>Smoothed Track Velocity - North</td>
<td>(m/s)</td>
</tr>
<tr>
<td>16</td>
<td>Integrated Tangential Shear</td>
<td>(m/s/km)</td>
</tr>
</tbody>
</table>
4. PROGRAM STATUS AND RECOMMENDATIONS

The processing programs are operational on the Interdata 7-32 computer. Experience must now be gained in using the system for the observation of weather. A number of parameters (Table 1) were set in the program on the basis of our experience with the data obtained from our contracts with the FAA and BuRec. It is anticipated that a different radar system operating in a different environment may need a different set of parameters. These parameters, such as the tracking measure weights and the constants in the velocity smoothing filter, are readily changed in the program. Experience with a larger data set is required to obtain the best estimate values for the parameters.

It is recommended that the cell detection and tracking programs be used on available data to develop the required operational experience to adequately use the new displays. The new output attributes are in a form that may be readily adapted to objective warning and forecasting systems.
5. REFERENCES


APPENDIX A

Operating Instructions*
CSS Files*
TET Files*
Task Establishment Maps*
Definition of Variables
List of Arrays
List of Common Blocks

*Listings for Real-Time and Post-Mission Programs
REAL-TIME PROGRAM OPERATING INSTRUCTIONS

TO EXECUTE REAL-TIME CELL DETECTION
AND TRACKING PROGRAM (ERT)

1. Compile each subroutine file:
   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:
   DCALOBJ ERT1:File Name*

3. Delete old task file:
   DE ERT1:CRANE.TSK

4. Create a new task file:
   ESTAB ERT1:GARY

5. Dispose I/O devices by editing the CSS file:
   EDITR CREAL.CSS

6. Execute the program:
   CREAL

*List of subroutine file names:

TSEMAIN   ATRAK
TSEDATA   BTRAK
REALTM    COMPAR
INPARM    RESOLV
CONTOR    COMBIN
PEAKD     STRAK
TRACK

A-2
REAL TIME CSS FILE

07/19/79 16:56:49
***LISTING FOR CREAL.CSS

$N

$JOB
L: BG, ERT1: TSEPARAM
T: BG
AS4, ERT1: TSEPARAM, DAT
ST: 01, 02, 03, 04, 05, 06, 09
L: LIB, ERT1: TSERTL, RTL
SE PA 2/0, 1/145, 0
L CRANE, ERT1: CRANE
T CRANE
AS3, CON:
AS5, CON:
AS8, ERT1: PLY
AS9, ERT1: DAT1024
AS10, PPRI:
AS7, PACK: TSEPPI, PAR
AS6, PR:
$IFNULL 07
   AS5, 07
$ENDC
$IFNULL 07
   AS5, ERT1: TSEDEF, PAR
$ENDC
AS4, ERT1: TSEPARAM, DAT
ST
$EXIT
*

A-3
REAL TIME TET FILE

07/19/79  16:55:46
***LISTING FOR ERT1:GARY. TET
$N
! LOG
 JOB ERT
 REMOTE
 ES TASK
 MXSPACE 2800
 OPTIONS F
 GET 400
 PRIORITY 10,10
 TCOM TASK/2/RW
 TCOM ZSTORE/3/RW
 TCOM DONE/4/RW
 TCOM EXTRA/5/RW
 IN ERT1:TSEMAIN
 IN ERT1:TSEDATA
 IN ERT1:REALTM
 IN ERT1:INPARM
 IN ERT1:CONTOR
 IN ERT1:PEAKD
 IN ERT1:TRACK
 IN ERT1:ATRAK
 IN ERT1:BTRAK
 IN ERT1:COMPAR
 IN ERT1:RESOLV
 IN ERT1:STRAK
 IN ERT1:COMBIN
 RESOLVE ERT1:TSERTL
 EDIT FVRTL
 BU TASK. ERT1:CRANE
 MAP
 END
*

A-4
PROGRAM SEGMENTS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>TYPE</th>
<th>NAME</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IMPU</td>
<td></td>
<td>143.75K</td>
</tr>
<tr>
<td>3</td>
<td>TCOM</td>
<td>ZSTORE TCM</td>
<td>4.25K</td>
</tr>
<tr>
<td>5</td>
<td>TCOM</td>
<td>EXTRA TCM</td>
<td>1.25K</td>
</tr>
<tr>
<td>15</td>
<td>RTL</td>
<td>TSERTL RTL</td>
<td>7.75K</td>
</tr>
</tbody>
</table>

PROGRAM LABELS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>008000</td>
<td>TSEMAIN</td>
<td>000620</td>
<td>TSEDATA</td>
<td>003838</td>
<td>ERTRMAP</td>
<td>010A80</td>
<td>ICLK</td>
<td>010A80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01BB70</td>
<td>COS</td>
<td>010C60</td>
<td>SIN</td>
<td>010DF8</td>
<td>I</td>
<td>010E60</td>
<td>R</td>
<td>010A80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011000</td>
<td>R</td>
<td>011110</td>
<td>SORT</td>
<td>011210</td>
<td>ALOG10</td>
<td>0112D8</td>
<td>ALOG</td>
<td>0115F8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011439</td>
<td>EXP</td>
<td>011580</td>
<td>FLOAT</td>
<td>0115A8</td>
<td>IFIX</td>
<td>0115F8</td>
<td>MOD2</td>
<td>0115F8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011649</td>
<td>ABS</td>
<td>011670</td>
<td>IABS2</td>
<td>011690</td>
<td>H</td>
<td>011700</td>
<td>S</td>
<td>011790</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008100</td>
<td>I</td>
<td>0117A0</td>
<td>.O</td>
<td>011D48</td>
<td>.MES</td>
<td>011790</td>
<td></td>
<td>011790</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TASK ENTRY-POINTS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000664</td>
<td>PPRDSC</td>
<td>00387C</td>
<td>RMAP</td>
<td>004204</td>
<td>CONTOR</td>
<td>0049A4</td>
<td>PEAKD</td>
<td>0049A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0098EC</td>
<td>TRACK</td>
<td>0090A4</td>
<td>ATRAK</td>
<td>00A87C</td>
<td>BTRAK</td>
<td>00A9F4</td>
<td>COMPAR</td>
<td>00A9F4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008FAC</td>
<td>RESOLV</td>
<td>00EFCC</td>
<td>STRAK</td>
<td>010764</td>
<td>COMBIN</td>
<td>010A82</td>
<td>ICLK</td>
<td>010A82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010872</td>
<td>COS</td>
<td>010880</td>
<td>COS</td>
<td>010CBA</td>
<td>SIN</td>
<td>010D22</td>
<td>SIN</td>
<td>010D22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0108F0</td>
<td>.I</td>
<td>010EEA</td>
<td>.A</td>
<td>011002</td>
<td>.R</td>
<td>011112</td>
<td>SORT</td>
<td>011112</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011120</td>
<td>.SORT</td>
<td>011210</td>
<td>ALOG10</td>
<td>011222</td>
<td>.ALOG10</td>
<td>0112DA</td>
<td>ALOG</td>
<td>0112A0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0112F2</td>
<td>ALOG</td>
<td>01131A</td>
<td>.LOGRT2</td>
<td>011432</td>
<td>EXP</td>
<td>01144A</td>
<td>EXP</td>
<td>01144A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011580</td>
<td>FLOAT</td>
<td>0115AA</td>
<td>IFIX</td>
<td>0115FA</td>
<td>MOD2</td>
<td>011642</td>
<td>ABS</td>
<td>011642</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011670</td>
<td>IABS2</td>
<td>011690</td>
<td>.H</td>
<td>011700</td>
<td>.S</td>
<td>011790</td>
<td>.O</td>
<td>011790</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0117B2</td>
<td>.I</td>
<td>011920</td>
<td>.I</td>
<td>011D48</td>
<td>.MES</td>
<td>011790</td>
<td></td>
<td>011790</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL COMMON BLOCKS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>011D68</td>
<td>CALB</td>
<td>011D78</td>
<td>MUSIG</td>
<td>011D80</td>
<td>READ2</td>
<td>011D80</td>
<td>SECTOR</td>
<td>011D80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011D90</td>
<td>REFL</td>
<td>012980</td>
<td>CAL</td>
<td>0129C0</td>
<td>RUNSUM</td>
<td>0139D0</td>
<td>TLI5</td>
<td>0139D0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0139E8</td>
<td>CALR</td>
<td>016208</td>
<td>SWITC H</td>
<td>020098</td>
<td>AZ2</td>
<td>0200B0</td>
<td>AZM</td>
<td>0200B0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0200C8</td>
<td>PNTS</td>
<td>0200D8</td>
<td>INTL</td>
<td>0200E8</td>
<td>ZLOOK</td>
<td>020258</td>
<td>ECONST</td>
<td>020258</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>020260</td>
<td>MAPPAR</td>
<td>020270</td>
<td>CNT</td>
<td>020288</td>
<td>DATA1</td>
<td>020610</td>
<td>DATA2</td>
<td>020610</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>021198</td>
<td>DATR3</td>
<td>0214A0</td>
<td>NVLIS</td>
<td>0214B0</td>
<td>FILTER</td>
<td>0214C8</td>
<td>KTA</td>
<td>0214C8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0214C8</td>
<td>CDRAVS</td>
<td>0222D0</td>
<td>CONST</td>
<td>022300</td>
<td>VPARM</td>
<td>022308</td>
<td>DVAL</td>
<td>022308</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>022310</td>
<td>CNTR5</td>
<td>022320</td>
<td>FLGS</td>
<td>022330</td>
<td>TMAX</td>
<td>022338</td>
<td>FNDRM</td>
<td>022338</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0223E8</td>
<td>FIXED</td>
<td>0223F8</td>
<td>PRSTOR</td>
<td>023AE0</td>
<td>THRESH</td>
<td>023AE7</td>
<td>CONPK</td>
<td>023AE7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>023AF0</td>
<td>RSLV</td>
<td>023C78</td>
<td>COM</td>
<td>023C78</td>
<td>COM</td>
<td>023C78</td>
<td>COM</td>
<td>023C78</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LIBRARY ENTRIES:

<table>
<thead>
<tr>
<th>SEG</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F0002</td>
<td>U</td>
<td>0F0052</td>
<td>V</td>
<td>0F006A</td>
<td>P</td>
<td>0F00FA</td>
<td>Q</td>
<td>0F00FA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0F0182</td>
<td>OR</td>
<td>0F02CB</td>
<td>OH</td>
<td>0F17E2</td>
<td>CONMSG</td>
<td>0F186A</td>
<td>FLOAT2</td>
<td>0F186A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TASK COMMON BLOCKS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
<th>NAME</th>
<th>ADDR</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>030000</td>
<td>ZSTORE</td>
<td>050000</td>
<td>EXTRA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END
POST-MISSION PROGRAM OPERATING: INSTRUCTIONS

TO EXECUTE POST-MISSION CELL DETECTION
AND TRACKING PROGRAM (CRANE)

1. Compile each subroutine file:
   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:
   DCAIORJ ERT1:File Name*

3. Delete old task file:
   DF ERT1:CRANE.TSK

4. Create a new task file:
   ESTAB ERT1:CRANE

5. Dispose I/O devices by editing the CSS file:
   EDITR CRANE,CSS

6. Execute the program:
   CRANE

*List of subroutine file names:

<table>
<thead>
<tr>
<th>Subroutine File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRANE</td>
<td></td>
</tr>
<tr>
<td>RESOLV</td>
<td></td>
</tr>
<tr>
<td>INFARM</td>
<td>COMBIN</td>
</tr>
<tr>
<td>CONTOR</td>
<td>STRAK</td>
</tr>
<tr>
<td>PFAKD</td>
<td></td>
</tr>
<tr>
<td>TRACK</td>
<td></td>
</tr>
<tr>
<td>ATRAK</td>
<td></td>
</tr>
<tr>
<td>BTRAK</td>
<td></td>
</tr>
<tr>
<td>COMPAR</td>
<td></td>
</tr>
</tbody>
</table>
POST-MISSION CSS FILE

**Listing of the CRANF.CSS File**

```
$JOB
SF PA 1 140 .2 0 .3 0
1 11B,FRT1:TSR:RTI
1 CRANF,FRT1:CRANF
T CRANF
AS2,NULL:
AS3,PR:
AS4,NULL:
AS6,FRT1:User Defined Output File*
AS7,FRT1:User Defined Calibration File**
AS9,FRT1:User Defined Input Data File***
ST
$TERMJOB
$EXIT
```

*File to which program output is to be sent*

**Disk file containing values for the parameters: I0UF, DRR, MAMV, MANS, SLOPE, OLDATA*

***Disk file generated by program "FRT"
POST-MISSION TET FILE

TET/32 R02-03

JOB ERT

REMOTE

ES TASK

MXSPACE 2800

OPTIONS F

GET 400

PRIORITY 10.10

TCOM TASK/2/RW

TCOM ZSTORE/3/RW

TCOM DONE/4/RW

TCOM EXTRA/5/RW

IN ERT1:CRANE

IN ERT1:INPARM

IN ERT1:CONTOR

IN ERT1:PEAKD

IN ERT1:TRACK

IN ERT1:ATRAK

IN ERT1:STRAK

IN ERT1:COMPAR

IN ERT1:RESOLV

RESOLVE ERT1:TSERTL

EDIT FVRTL

BU TASK, ERT1:CRANE

MAP

MAP

MAP
POST-MISSION TASK ESTABLISHMENT LOAD MAP

DATE 07/19/79 TIME 13:10:27

JOB: ERT
**** CTOP=01E5FE UTOP=01E1F8 MIN PARTITION= 12150K ****

PROGRAM SEGMENTS:

<table>
<thead>
<tr>
<th>SEG</th>
<th>TYPE</th>
<th>NAME</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IMPU</td>
<td></td>
<td>120 50K</td>
</tr>
<tr>
<td>3</td>
<td>TCOM</td>
<td>ZSTORE</td>
<td>TCM 4.25K</td>
</tr>
<tr>
<td>5</td>
<td>TCOM</td>
<td>EXTRA</td>
<td>TCM 2.50K</td>
</tr>
<tr>
<td>15</td>
<td>RTL</td>
<td>TSERTL</td>
<td>RTL 7.75K</td>
</tr>
</tbody>
</table>

PROGRAM LABELS:

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>000100 ERTRMAP</td>
<td>00E850 COS</td>
</tr>
<tr>
<td>000BC8 A</td>
<td>00ECE0 R</td>
</tr>
<tr>
<td>00EFB8 ALOG</td>
<td>00F110 EXP</td>
</tr>
<tr>
<td>00F2D8 ABS</td>
<td>00F310 IABS2</td>
</tr>
<tr>
<td>00F968 .MES</td>
<td></td>
</tr>
</tbody>
</table>

TASK ENTRY-POINTS:

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>001FD4 CONTOR</td>
<td>002774 PEAKD</td>
</tr>
<tr>
<td>0064C BTRAK</td>
<td>008BC4 COMPAR</td>
</tr>
<tr>
<td>00E534 COMBIN</td>
<td>00E852 COS</td>
</tr>
<tr>
<td>00E9B2 SIN</td>
<td>00EADA 1</td>
</tr>
<tr>
<td>00EDF2 SORT</td>
<td>00EEBA SORT</td>
</tr>
<tr>
<td>00EFBA ALOG</td>
<td>00EFD2 ALOG</td>
</tr>
<tr>
<td>00F12A EXP</td>
<td>00F26A FLOAT</td>
</tr>
<tr>
<td>00F312 IABS2</td>
<td>00F332 S</td>
</tr>
<tr>
<td>00F54E 01</td>
<td>00F96A .MES</td>
</tr>
</tbody>
</table>

LOCAL COMMON BLOCKS:

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>00F988 TLIS</td>
<td>00F9A0 CONST</td>
</tr>
<tr>
<td>019E78 AZM</td>
<td>019890 REF1</td>
</tr>
<tr>
<td>01A4D0 ZLOOK</td>
<td>01A640 ECONST</td>
</tr>
<tr>
<td>01A670 DATA1</td>
<td>01A9F8 DATA2</td>
</tr>
<tr>
<td>01B898 FILTER</td>
<td>01B8AB KTA</td>
</tr>
<tr>
<td>01C6C0 DVAL</td>
<td>01C6CB CNTR5</td>
</tr>
<tr>
<td>01C6F0 PWORK</td>
<td>01C7A0 FIXED</td>
</tr>
<tr>
<td>01DEA0 CONPK</td>
<td>01DEA8 RSLV</td>
</tr>
</tbody>
</table>

LIBRARY ENTRIES:

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F0002 U</td>
<td>0F0052 V</td>
</tr>
<tr>
<td>0F01B2 OR</td>
<td>0F02CB 0H</td>
</tr>
</tbody>
</table>

TASK COMMON BLOCKS:

<table>
<thead>
<tr>
<th>LABEL</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>030000 ZSTORE</td>
<td>050000 EXTRA</td>
</tr>
</tbody>
</table>

END
VARIABLE LIST

+TL1 - lower threshold (dBZ)
+TL2 - higher threshold (dBZ)
+RQUNT - threshold quantization factor (dBZ)
+ILX - intermediate values in dBZ conversion
+IOFF - reflectivity offset to insure positive values
+DB - calibration constant
+BVVEL - constant used in calculation of velocity
+BVVAR - constant used in calculation of variance
+BVWORD - square of variance
+BLEVAT - raw elevation (encoder units)
+BLEVAA - elevation in degrees
+BLESN - elevation in degrees
+T - time (seconds from start of year)
+AC - raw azimuth (deg)
+ACT - azimuth in degrees
+A - azimuth in radians
+ACCHK - azimuth in deg. * 359
+K - radial counter
+BGNAA - start azimuth (deg)
+ENDAA - stop azimuth (deg)
+SINA - sin(A)
+COSA - cos(A)
+SN - check on first azimuth of new scan
+NAC - offset multiplier for arrays in CONTO & PEAKD
+IDAY +DAY - Julian date data collected
+HR - hour
+MIN - minute
+SEC - second
+DELTA - azimuth increment (radians)
+PHI - elevation angle (radians)
+COSPHI - cos(PHI)
+SINPHI - sin(PHI)
+EARTH - 6.4857E-5 (km^-1)
+COSPH2 - COSPH1 *EARTH
+ALAST - azimuth of previous radial (deg)

* = INTEGER*2

* = REAL
*IFLAG - intermediate print flag
*TLS - lower threshold (TL1 in CRANE)
*TATRMN - test on area
*NEMC - array addressing offset
*NCEL - cell counter
*IEMAX, *JMAX - array limits
*NEMI - address variable
*NCL - maximum number of positions
*TL1 - lower threshold
*IEM - event counter on first threshold
*IEM2 - event counter on second threshold
*IPE - peak start location
*IP - peak counter
*IEVENT - event number

+ - INTEGER*2

* - REAL
PEAKD

*NBDRA, +NCADR, +NBKA, +NCKA - address variables
+NAX, +NA - radial counter
+LI - number of variables in UP (Array)
*LMDP, +NAN, +NAN1, +LMM, +IDX, +NCLM, +LDBM, +LDX, +NPDP, +ID2 - address variables
*FQUANT - threshold quantization factor
*KOFST, +LIMIT, +NDP, +MXTR, +KMAX - Array limits
+II - event number
+IEM - number of events on radial
+IEA, +K1E, +K1EM - address variables on IEM
+ICEST - event start position
+ICESP - event end position
*JEB, +KA, +KB - address variable on event and peak
+IPL - peak start location
+IP - peak stop location
+NTHRES - threshold counter for peak
+LDB - dB below peak value used to define peak
+IR1 - range to peak
+II - dBZ value at IR1
+IT - dBZ above threshold at IR1 for LDB thresholds
+JMXM - limit on IT
+KA - address variable on peak thresholds
+IPT - number of thresholds associated with peak
+IR - limit on IPT
+IBGN - first position in event
+IND - last position in event
+IV - reflectivity values within event
+KA - address variable on peak threshold values within event
+JMXJM - limit on number of contours per radial
+IREG - start or stop range of contour
+IPE - contour counter
+IADDR & +IEQL - address variables on contour threshold and number
+KC, +KA, +KZ - address variables on contour thresholds
+TCVL, +TCVM, +TCVLB - threshold values on a peak
+NPC - number of contours by threshold on this radial
+NPL - contour counter by threshold
+IIBM - start range of contour
+IIH - IIHM + 1
+IH - stop range of contour
+K, +KY, +KZ - address variables on next threshold
+LPE - number of contours, this threshold, on radial
+LPL - contour counter, this threshold
+NFCCEL - ID number for possible cell, this threshold
+TATC - ID TATR(NPCCEL) points to
+JE1 - first event previous radial
+JE2 - last event previous radial
+JEM - address variable on events, previous radial
+IPB - number of peak thresholds
+KB, +KBB, +KBA, +KRC - address variables on contour thresholds
+TBVL - threshold value on a peak
+NPI - number of contours on a threshold, previous radial
+NPI - contour counter, by threshold, previous radial
* - INTEGER*2 * - RFAI
PEAKD (continued)

+LPCEL - ID number for possible cell, previous radial
+TEQL - ID TATR(LPCEL) points to
+NPC - equals NPCEL if associated
+JEQL - next higher threshold
+JN1, +JN2, +JN3, +JN4, +JN5 - address variable on peak
+IST - start range of contour
+ISP - stop range of contour
*R - area per azimuth degree at peak range
+IU - reflectivity at that range
*RU - reflectivity weighted area
*SAZ - sin(azimuth)
*CAZ - cos(azimuth)
+KNN, +KN - address variable on NPCEL
+INDX - address variable on second threshold of peak
+IND, +INDX, +LNX - address variable on LPCEL
+IN, +IEQL - address variable on LPCEL
+JEQL - area address of NPCEL pointed to by area of LPCEL
+IPTT - number of peak thresholds
+KTI, +KTA, +KTB - address variables on contour threshold
+NPCT - number of contours, this threshold
+IEQL - threshold value
+INDXT - address variable on NPCEL
+NIMN - number of possible cells tested so far
+NIDP - limit on number of possible cells
+IE - event number
+IPT - number of peak thresholds in event
+NPC - number of thresholds this peak
+NCVM - threshold value on peak
+NPL - contour counter

* - REAL

+ - INTEGER*2
PEAKD (continued)

+I - each position out the radial
+IA - address variable on I
+IEQL - ID value at each position
+J - event number on previous radial
+JA - address variable on J
+IPB - number of peak thresholds in event
+KA,+KAP,+KAM - address variables on event and threshold
+NP - number of contours, this threshold, on previous radial
+ITERM - code for eliminating possible cell
+MG - address in UP (array) for measure of cell significance
+LMT - limit on number of cells to be carried in order of significance
+JKI - address variable on previous radial
+JKL - address variable on current radial

+ - INTEGER*2
**VKM** - \( \cos (\text{elevation}) \times \text{conversion m to km} \)

**SAVM** - unit area convert from \( m^2 \) to \( km^2 \)

**MA** - address in \( ECL \) (array) of measure of cell significance

**M** - cell counter

**M1, M2, M3, M4, M5, M6, M7** - address of each attribute in \( ECL \) (array) by cell

**KOFST, KAN2** - array addressing offset

**VKME** - convert reflectivity weighted line of sight distance to horizontal dist.

**FNSN, NSCAN** - scan counters

**KTL** - time

**JDAY, IDAY** - Julian date

**JHR, IHR**

**JMIN, IMIN** - start time of first scan in sequence

**JSEC, ISEC**

**NC** - cell number (current scan)

**NCG, NCGB, NCBS** - NC to pass through common

**NVMIN, NVMX, NCMX** - number of significant cells detected this scan

**IELSN** - current elevation angle

**IESNL** - elevation angle last scan

**ATRAK**

**NCEC** - address offset on \( NC \)

**NVRC** - address offset on \( NV \)

**NC** - cell counter for \( ECL \) (array)

**NV** - cell counter for \( VCL \) (array)

**NCA** - address variable on \( NC \)

**NVA** - address variable on \( NV \)

**IZ** - cell reflectivity

**X** - cell position east [(-)west]

**Y** - cell position north [(-)south]

**H** - cell height

**IZL** - offsetted reflectivity

**Z** - reflectivity

**HI** - height of last cell

**IP** - peak reflectivity

**BTRAK**

**NCEC** - address offset on \( NC \)

**NVVR** - address offset on \( NV \)

**NC** - cell counter for \( ECL \) (array)

**NV** - cell counter for \( VR \) (array)

**NVA** - address variable on \( NV \)

**NCA** - address variable on \( NC \)

**+ - INTEGER*2**

**- - REAL**
COMPAR

+NCMX - number of cells detected this scan
+1M, +JM - array limits
+NC - cell numbers detected this scan
+NCEC - address offset on NC
+(NC1 to NC6) - address variables on NC
+NV - cell numbers tracked from previous scans
+NVVC - address offset on NV for VCL (array)
+NVVR - address offset on NV for VR (array)
+NLR - address variable on NV
*A%TEST - estimate of cell NV's movement from last scan to this scan
*DDEL - time since last scan
*DELT, DELY - distance on X & Y coordinates between cell NV and NC
*DEIW - a measure of the association between NV and NC using reflectivity, location, area, height
+IO - overflow of D and ID (arrays)
+NSCAN - scan number
+IO - overflow of IDV (arrays)
*DX - DELL of a previous association with this NC cell
+NVT - NV previously associated with this NC cell
+NCT - NC previously associated with this NV cell
*DIK - DELL of a previous association with this NV cell
+VMX - number of active cell tracks
+NCR - NC cell to test NV cells against
+NVB - NV \} associated
+NCB - NC \} associated
*HCT - update height

+ - INTEGER*2

* - REAL
RESOLV

*NVT - NV associated with NCT
+I - number of NV associations on this NCT
*JX - number of NC cells associated with 1 NV cell
*NCT - NC associated with NVT
+I - number of NC associations on this NVT
*JX - number of NV cells associated with 1 NC cell
*NVT - NV cell that NC cell is associated with (highest DELK - down)
*IV - number of cells associated to this NV cell
*KV - maximum number of associations to one NV cell
*IWS - NV cell association on
*LC - counts NC cells checked
*NCT - same as NVT but NC on NV
*JC - same as IA but on an NC cell
*KC - same as KV but on an NC cell
*ICS - NC cell association on
*LV - counts NV cells checked
*NV - NV cell that NC cell is associated with
*KA, KA2, KA1 - address variables on KC
*NC - NC cell that NV cell is associated with
*MMSM - counts associations on NV other than NC
*DELK - measure of compare
*DWT - minimum measure of compare
*NV - cell with DWT
*KNC - counts cells with too many associations
*DELT - test on cell velocity
*NVT1 - next NV cell associated with NC (in order of DELK)
*DWT1 - minimum measure on NVT1
*NV1 - cell with DWT1
*DELW1 - test on NV1
*DELW2 - test on NV
*HTE - update height

+ - INTEGER*2
# - INTEGER*4
* - REAL
STRAK

*NVA - address offset on NV
*(NV1 to NV41) - address variables on NV for VCL (array)
*NVR - address offset on NV
*(NR1 to NR6) - address variables on NV for VR (array)
*NW - cell counter
*VXT - velocity east [(-)west]
*VYT - velocity north [(-)south]
*DELT - time since last scan
*I2VAL - reflectivity
*IPTC - percent of scans cell was detected
*VXC - sum of eastward velocity components
*VYC - sum of northward velocity components
*NFN - number of scans processed
*NSN - number of velocity values summed in VXC and VYC
*VX - average eastward velocity of all cells
*VY - average northward velocity of all cells
*NVSCN - number of volume scans
*KTL - time of last scan

* - INTEGER*2

* - RFAL
LIST OF ARRAYS - THEIR SIZE AND CONTENT

In CRANE

ANC(1028) - header information - INTEGER*4
Zee(1024) - raw data - INTEGER*4
Iref(1024) - reflectivity - INTEGER*2
IVel(1024) - radial velocity - INTEGER*2
IVar(1024) - variance - INTEGER*2
RE(1025) - decoded reflectivity - INTEGER*2
IVell(256) - decoded velocity - INTEGER*2
IDvel(256) - tangential shear - INTEGER*2
Zary(91) - convert reflectivity to dBZ - REAL

In CONTOR

ICl(44) - start position of event - INTEGER*2
IC2(44) - stop position of event - INTEGER*2
IDc(22) - number of peaks in each event - INTEGER*2
IPRNG(34) - location of peaks - INTEGER*2
IC21(22) - start position of event on second threshold - INTEGER*2
IC22(22) - stop position of event on second threshold - INTEGER*2

Array contains indicated parameter(s) on the current radial, offsetted from the same parameter(s) on the previous radial
In PEEKD

T(80) - all possible thresholds a peak may have - INTEGER*2
TC(1980)* - thresh of each peak - INTEGER*2
IPTC(44)* - number of thresholds in each event - INTEGER*2
ICNRT(1980)* - contour counter - INTEGER*2
IPC1(5400)* - start range of contour segment - INTEGER*2
IPC2(5400)* - end range of contour segment - INTEGER*2
IPC3(5400)* - number of peaks within the segment - INTEGER*2
TATR(1400) - temporary attribute array - stores peak attributes until a
cell is detected or peak discarded - REAL
IACT(70) - overflow (too many peaks) - INTEGER*2

UP(6) - cell attributes - REAL

1 - area
2 - reflectivity
3 - location in km east of radar
4 - location in km north of radar
5 - tangential shear
6 - a measure of relating cell significance = const * area + tan shear

*array contains indicated parameter(s) on the current radial, offsetted
from the same parameter(s) on the previous radial
PEAKD & TRACK:  BCl (7x16x2)  (1,N,NAN)

<table>
<thead>
<tr>
<th></th>
<th>variable</th>
<th>real</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>area surf</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>dBZ</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>east</td>
<td>real</td>
</tr>
<tr>
<td>4</td>
<td>north</td>
<td>real</td>
</tr>
<tr>
<td>5</td>
<td>range</td>
<td>real</td>
</tr>
<tr>
<td>6</td>
<td>height</td>
<td>real</td>
</tr>
<tr>
<td>7</td>
<td>tang shear</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>defined in TRACK</td>
<td></td>
</tr>
</tbody>
</table>

ATRAK:  VCl (25x5.2)  (1,N)

<table>
<thead>
<tr>
<th></th>
<th>variable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>east (X)</td>
<td>real</td>
</tr>
<tr>
<td>2</td>
<td>north (Y)</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>dBZ</td>
<td>int*4</td>
</tr>
<tr>
<td>4</td>
<td>area surf</td>
<td>real</td>
</tr>
<tr>
<td>5</td>
<td>time</td>
<td>int*4</td>
</tr>
<tr>
<td>6</td>
<td>height</td>
<td>real</td>
</tr>
<tr>
<td>7</td>
<td>range</td>
<td>real</td>
</tr>
<tr>
<td>8</td>
<td>NT; track ID</td>
<td>int*4</td>
</tr>
<tr>
<td>9</td>
<td>#scans each det.</td>
<td>int*4</td>
</tr>
<tr>
<td>10</td>
<td>H</td>
<td>real</td>
</tr>
<tr>
<td>11</td>
<td>EE</td>
<td>real</td>
</tr>
<tr>
<td>12</td>
<td>EEY</td>
<td>real</td>
</tr>
<tr>
<td>13</td>
<td>EE(H-HL)\times Area</td>
<td>real</td>
</tr>
<tr>
<td>14</td>
<td>H summit ((HI) last)</td>
<td>real</td>
</tr>
<tr>
<td>15</td>
<td>dBZ summit</td>
<td>int*4</td>
</tr>
<tr>
<td>16</td>
<td>dBZ peak</td>
<td>int*4</td>
</tr>
<tr>
<td>17</td>
<td>H peak</td>
<td>real</td>
</tr>
<tr>
<td>18</td>
<td>X</td>
<td>real</td>
</tr>
<tr>
<td>19</td>
<td>Y</td>
<td>real</td>
</tr>
<tr>
<td>20</td>
<td>time</td>
<td>int*4</td>
</tr>
<tr>
<td>21</td>
<td>Vel x</td>
<td>real</td>
</tr>
<tr>
<td>22</td>
<td>Vel y</td>
<td>real</td>
</tr>
<tr>
<td>23</td>
<td>tang shear</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>updated each scan</td>
<td></td>
</tr>
</tbody>
</table>

- lowest elevation angle only
- def in CRANE
- updated each scan
- def in STRAK
- updated each vol scan
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>int*4</td>
</tr>
<tr>
<td>X</td>
<td>real</td>
</tr>
<tr>
<td>Y</td>
<td>real</td>
</tr>
<tr>
<td>Z</td>
<td>real</td>
</tr>
<tr>
<td>(H-Hil.) xArea</td>
<td>real</td>
</tr>
<tr>
<td>(not used)*</td>
<td></td>
</tr>
<tr>
<td>dBE peak</td>
<td>int*4</td>
</tr>
<tr>
<td>H peak</td>
<td>real</td>
</tr>
<tr>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>dBE</td>
<td>int*4</td>
</tr>
<tr>
<td>area surf</td>
<td>real</td>
</tr>
<tr>
<td>height</td>
<td>real</td>
</tr>
<tr>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>dBZ summit</td>
<td>int*4</td>
</tr>
<tr>
<td>H summit</td>
<td>real</td>
</tr>
<tr>
<td>ABS(NT) TRAKID</td>
<td>int*4</td>
</tr>
<tr>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>ΔX/AT</td>
<td>real</td>
</tr>
<tr>
<td>ΔY/AT</td>
<td>real</td>
</tr>
</tbody>
</table>

*Array locations not used in the current version of the program*
track cells from previous scan (NV) to cells in current scan (NC)

IUC1(16) - points to NV each NC is associated with - INTEGER*2
  i.e. IUC1(NC) = NV

IC(16) - stores the "measure" of the above association - INTEGER*2

IUC2(16) - if one NC is associated with more than one NV then IUC2(NC) ≠ 0 - INTEGER*2

IUV1(52) - points to NC each NV is associated with - INTEGER*2

IV(52) - stores the "measure" of the above association - INTEGER*2

IUV2(52) - if one NV is associated with more than one NC then IUV2(NV) ≠ 0 - INTEGER*2

IC(52x10) - if IUC2(NC) ≠ 0 - INTEGER*2

IC(y, 1) = number of conflicts

IC(y, 2-10) - the NVs associated with

C(52x9) - stores the measures of each NC-NV association - REAL

ID(52x10) - same as the IC(array) for conflicts on NV - INTEGER*2

D(52x9) - stores the measures of each NV-NC association - REAL

In RESOLVE

IV(52, 7) - ordered IC(array) or ID(array) - INTEGER*2

V(584) - stores the measures of the tested associations - REAL
## List of Common Blocks and Their Associated Routines

<table>
<thead>
<tr>
<th>Block</th>
<th>Block Name/Size*</th>
<th>Data</th>
<th>Contor</th>
<th>Peakd</th>
<th>Track</th>
<th>Atrak</th>
<th>Btrak</th>
<th>Compar</th>
<th>Resolv</th>
<th>Strak</th>
<th>Combin</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZM</td>
<td>(4.5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZ2</td>
<td>(6)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDRAYS</td>
<td>(897)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLST</td>
<td>(257)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNT</td>
<td>(3)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNTRS</td>
<td>(3)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONPIC</td>
<td>(1)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONST</td>
<td>(12)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA1</td>
<td>(226)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA2</td>
<td>(737)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA3</td>
<td>(193)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVAL</td>
<td>(1)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ECONST</td>
<td>(2)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTRAX</td>
<td>(597.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILTER</td>
<td>(5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIXED</td>
<td>(.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLGS</td>
<td>(.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTL</td>
<td>(3)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTA</td>
<td>(2)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTB</td>
<td>(1)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAPPAR</td>
<td>(4.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVLIS</td>
<td>(4)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNTRSS</td>
<td>(4)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRSTOR</td>
<td>(1465)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWORK</td>
<td>(43.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFIL</td>
<td>(775)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLV</td>
<td>(72.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWITCH</td>
<td>(1047.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THRESH</td>
<td>(.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TILS</td>
<td>(5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMNAX</td>
<td>(.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VPARM</td>
<td>(2)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZLOOK</td>
<td>(91.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZSTORE</td>
<td>(1028)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Size is number of 32 bit words
APPENDIX B

FLOW CHARTS
START

Program CRANE
Initialize Tables, Read And Decode Radar Data

Subroutine CONTOR
Find Fixed Contours & Calculate Attributes

Subroutine PEAKD
Find Peak Referenced Contours And Calculate Attributes

Subroutine TRACK
Initialize Track Tables

Subroutine COMPAR
Associate Current To Previous Scan

Subroutine RESOLV
Resolve Conflicts In Compar

Subroutine STRAK
Output Track Attributes

Subroutine ATRAK
Define Lowest Elevation Attributes
Subroutine BTRAK
Define Last Elevation Attributes

END

COMPUTER PROGRAM STRUCTURE
Loop on Volume Scan

Loop on Azimuth Scan (Fixed Elevation)

Loop on Azimuth (Single Radial)

Read & Scale

Reformat Data to Reflectivity, Velocity & Velocity Variance Estimates

Fixed Contour

Detect Larger Echo Regions Using Fixed Threshold Contours

Detect Cells

Detect Cells Using Variable Threshold Contours

Sum Attributes

Accumulated Intermediate Results to Calculate Attributes

Associate Cells & Echoes

Determine Height Dependencies of Cells & Echoes by Identifying the Same Cells on Successive Azimuth Scans

Associate Cells & Echoes

Determine Cell Tracks by Identifying the Same Cell in Successive Volume Scans

Output Cell Track Data

Overall Processing Scheme
REAL TIME PROCESSING SCHEME:
START

Input Calibration Data

Loop on Azimuth Scan (Constant Elevation)

Loop on Azimuth (Single Radial)

Read RADAR DATA
Reflectivity
Velocity
Variance

Decode RADAR DATA

Call CONTOR

NO

RADIAL Counter Reset To 1

NO

EOF

YES

Call TRACK

Call STRAK

END

PROGRAM CRANE
Event Identification
Subroutine CONTOR
Peak Detection
Subroutine PLAAD
Loop on Events this Radial (C - Radial)

Loop on Events Prior Radial (B - Radial)

Stop Range, B < Start, C

Start, B > Stop, C

Any Prior B Radial Association

Yes

Combine B & C

Combine B with Prior B & C

Any B Radial Association

New End Region Initialize Attributes

START

STOP

Event Association

Subroutine PEAKD
INPUT AFTER EACH SCAN

Loop on cell this scan
Loop on track

Compare

Store association in UV, UC arrays

Yes

No

Loop on all

Find all cells and tracks in cluster
Find test match

Yes

Update Attributes

End of volume scan

Yes

Output Track Attributes

No

Exit
**LISTING FOR ERT1:TSEMAIN.FTN**

**TSEMAIN PROG**

**$TITL**

**FILE TSEMAIN - MAIN PROGRAM FOR ETSE**

**IMPLICIT INTEGER*2 (A-Z)**

**REAL DB,BITVEL**

**INTEGER*2 RHO, TFSO, ZTH, TFS, K, BETA**

**INTEGER*2 INDCTR, ANG(7), DISPLA, MMU, SIGMA, ELEV(2)**

**INTEGER*4 TIME, RSERVE(i6), SECOND, NRCEAD**

**INTEGER*4 NEXTIM, ITIME**

**COMMON /CALB/ DB, BITVEL, NRC**

**COMMON /MUSIG/PCTMIN, MMU, SIGMA**

**COMMON /SECTOR/INDCTR**

**COMMON /RERDZ/NRCEAD, OLD**

**COMMON /SECTOR/INDCTR**

**COMMON /EXTRARHO, GRND, ZTH, BETA, K, RRArea(12, 24), RHO2, ZTH2, BETA2,**

**IPCT2, MMU2, SIGMA2, MINUTE2, ANG(2), CHANGE**

**EQUIVALENCE (DISPLA.ANG(±)), (DEMAND, ANG(6)).(ELEV(i),ANG(4))**

**EQUIVALENCE (ANGLE, ANG<2)), (OLDATA, ANG(7)>**

**DATA DISPLAY/4, NLEVEL/3/**

**REIND 4**

**INDCTR=0**

**10000 CONTINUE**

**$ASSM**

**FREZE**

**COPY SVC1**

**STM 0, RSAVE**

**READ SVC 1, READBLOK READ IN DISPLAY, ANGLES**

**LH R0: READBLOK+SVC 1 STA READ IN DEVICE STATUS**

**BNZ IOERR BRANCH IF NOT ZERO**

**LH R0: ANGLE LOAD IN BEGINNING ANGLE**

**CHI R0: 360 LESS THAN 360?**

**BNL ALLD NO, FULL SCAN**

**LIS R0: 1**

**STH R0: INDCTR INDCTR=1**

**ALLD LM 0, RSAVE**

**$FORT**

**READ<7, 333> IOUT, DBB, MAXV, MAXS, SLOPE**

**333 FORMAT(I3)**

**BITVEL=MAXV**

**DB=65.28**

**READ<5, 110>RHO, ZTH, BETA, IPCTMN, MMU, SIGMA, MINUTE**

**110 FORMAT(I3)**

**CHANGE=0**

**PCTMN=IPCTMN*0.01**

**SECOND=MINUTE*60**

**OLD=OLDATA**

**50 CONTINUE**

**CALL ICLOCK(2, TIME)**

**NEXTIM=TIME+SECOND**

**SVELE=RNG(5)**

**ANG(5)=ANG(4)+5**

**DO 51 N=1, NLEVEL**

**CALL PPRDSC(ANG)**

**CALL RMAP**

**ANG(5)=SVELE**

**51 CONTINUE**

**CALL PPRDSC(ANG)**
CALL TSEPLT(DISPLA)
IF(CHANGE)55, 55, 52
CHANGE=0
IF(RHO2. NE. (-I))RHO=RHO2
IF(PCT2. NE. (-I))PCTMN=PCT2
IF(MMU2. NE. (-I))MMU=MMU2
IF(SIGMA2 NE. (-I))SIGMA=SIGMA2
IF(ZTH2. NE. (-I))ZTH=ZTH2
IF(BETA2. NE. (-I))BETA=BETA2
IF(MINUT2. NE. (-I))MINUTE=MINUT2
DO 54 I=1, 6
IF(ANG2(I). NE. (-I))ANG(I)=ANG2(I)
CONTINUE
PCTMIN=IPCTMN+0.01
SECOND=60*MINUTE
IF(DEMAND=1)56, 75, 50
56 CALL ICLOCK(2, ITIME)
IF(ITIME. GT. TIME)GO TO 58
IF(NEXTIM LE. 86400)GO TO 50
NEXTIM=NEXTIM-86400
58 IF(NEXTIM LT. ITIME)GO TO 50
CONTINUE
$ASSM
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
STM R0, RSAVE
L R1, NEXTIM
ST R1, SVCTIM
SVC 2, TIMBLK
LM K0, RSAVE
RETURN TO FORTRAN
$FORT
GO TO 50
75 PAUSE 0
GO TO 50
$ASSM
ALIGN 4
IOERR	EXBR R1, R0
NHI R1, X'00FF'
CHI R1, X'88'
BE ALLED YES, FULL SCAN, DEFAULT DISPLAY
CHI R1, X'90'
BE ALLED YES, FULL SCAN, DEFAULT DISPLAY
SVC 2, ERRCODE
SVC 2, ERRBLOK
SVC 2, PAUSE
B READ
TRY AGAIN
ALIGN 4
READBLOK EQU *
DB X'58'
DB 4
DC H'O'
DC A(DISPLA)
DC A(OLDATA+1)
DSF 3
ALIGN 4
ERRCODE EQU *
DB 0
DB 6
DC H'O'
DC A(ERCD)
ALIGN 4
ERRBLOK EQU *
( DB 7
  DC H'30'   30 CHRS
ERCD  DC C'
  DC C'I/O ERROR IN RTN TSEMA
  ALIGN 4
PAUSE  EQU *
  DB 0.1
  ALIGN 4
TIMBLK  EQU *
  DB 0
  DB 10  CODE 10
  DB 0.0
SVCTIM  DC F'0'  TIME OF DAY
$FORT
  END
  *
**FILE TSEDATA - DATA INPUT SUBROUTINE FOR ETSE**

**SUBROUTINE FPRDSC(ANG)**

IMPLICIT INTEGER*2 (A-Z)

INTEGER*4 PPRIC1(1028), PPRIC2(1028), RSAVE(16)

+ REAL DB, BITVEL

INTEGER*2 IREF(1024), IVEL(1024), IVELL(1024),
+ IREF2(1024), IVEL2(1024)

INTEGER*4 PPRANG

INTEGER*2 ANGC(6), RE(1025), HR(258), TL1, TL2, ROUANT

EQUIVALENCE (PPRANG, PPRIC1(3)), (PP, PPRIC2(2))

EQUIVALENCE (DECOD(5), IREF(1)), (DECOD(517), IVEL(1)),
+ (DE2COD2(5), IREF2(1)), (DE2COD2(517), IVEL2(1))

COMMON /REFL/ RE, HR, NCL, NIDP, INCL, IMX,
+ IMN, TL1, TL2, ROUANT, IVEL(258)

COMMON /CAL/ DB, BITVEL, NRC

COMMON /SECTOR/ INDCTR

COMMON /ZSTORE/ PPRIC1

COMMON /RUNSUM/ PPRIC2

CALL CONMSGC6, 'PPRDSC'

REWIND 9

NAZ=0

ELEV1=ANG(4)*11.37778

ELEV2=ANG(5)*11.37778

IF (INDCTR.EQ.2) INDCTR = 0

IF (INDCTR.EQ.0) GO TO 1

CW = ANGC(3) - ANGC(2)

IF (CW.GT.180) CW = CW - 360

IF (CW.LT.(-180)) CW = 360 + CW

IF (CW.EQ.0) CW = 0

IF (CW.GT.0) CW = 15

BGNA = 11.37778 * ANGC(2)

ENDA = 11.37778 * ANGC(3)

BGNA=MOD(BGNA, 4096)

ENDA=MOD(ENDA, 4096)

IF (ENDA.GT.4086) ENDA = 0

STPFLG = 0

NITG=4

NRC=768

IF (CW.GT.0) AND. ENDA.LT. BGNA) STPFLG = 1

IF (CW.EQ.0) AND. ENDA.GT. BGNA) STPFLG = 1

IF (BGNA.GT.10) AND. BGNA.LT.4086) GO TO 1

BGNA = 0

STPFLG=2

1 CONTINUE

10000 CONTINUE

**$ASSM**

FREZE

CROSS

COPY SVC1.

R0 EQU 0

R1 EQU 1

R2 EQU 2

R3 EQU 3
R5 EQU 5
R6 EQU 6
R14 EQU 14
R15 EQU 15
R13 EQU 13
R6 EQU 6
STM 0, RSAVE
L R4, WAITREAD+SVC1.SAD
AIS R4, 15
ST R4, WAITREAD+SVC1.EAD
BAL R13, WREAD
LIS R1, 4
L R0, PPRI(R1)
NHI R0, 3
AIS R0, 1
SLLS R0, 8
AIS R0, 4
SLLS R0, 2
SIS R0, 1
L R4, PPRLBLK+SVC1.SAD GET BEGINNING ADDRESS
AR R4, R8 COMPUTE END ADDRESS
ST R4, PPRLBLK+SVC1.EAD STORE IN EAD
L R4, WAITREAD+SVC1.SAD
AR R4, R8
ST R4, WAITREAD+SVC1.EAD SAME NUMBER
L R4, OUTBLK+SVC1.SAD
AHI R4, 1028*4-1
ST R4, OUTBLK+SVC1.EAD
L R4, PPRLBLK2+SVC1.SAD GET NEXT BEGINNING ADDR
AR R4, R8 COMPUTE END ADDRESS
ST R4, PPRLBLK2+SVC1.EAD STORE IN SVC BLOCK
L R4, OUTBLK2+SVC1.SAD
AHI R4, 1028*4-1
ST R4, OUTBLK2+SVC1.EAD
LDAI R15, PPRI
LDAI R14, PPRI2

DETECT BAL R13, WREAD READ IN AN AZIMUTH
LH R1, 8R15) LOAD IN AZIMUTH DATA
NHI R1, X'FFF' AND OUT UNWANTED BITS
BAL R13, WREAD GET ANOTHER AZIMUTH
LH R2, 8R15) GET AZIMUTH DATA
NHI R2, X'FFF' AND OUT UNWANTED BITS
CR R1, R2 COMPARE TWO AZIMUTHS
BL CWISE IF R1<R2, RADAR IS GOING CWISE
LIS R3, 0 DIRECTION FLAG
B WHATIZIT CONTINUE
CWISE LIS R3, 15 DIR 15LAG = CW
WHATIZIT LH R4, INDCTR SECTOR SCAN OR FULL CIRCLE?
BZ EDETECT FULL CIRCLE
CH R3, CH IS DIRECTION OF ROTATION CORRECT?
BNE DETECT WRONG DIRECTION, WAIT
OR R3, R3 WHICH DIRECTION IS IT?
BZ CCW COUNTER CLOCKWISE
LH R5, STPFLG
THI R5, 2 CASE 2?
BNZ CWCASE2 CW CASE 2
CH R2, BGNA ANGLE < BGNA?
BL WAIT YES, GET READY
B DETECT NO, TRY AGAIN

CWCASE2 CHI R2, X'800' ANGLE > 180?
BL CMP1 NO, ALL OK
SHI R2, X'1000' YES, SUBTRACT 360
B CMP1
CCW LH R5, STPFLG (-6
BNZ CCPASE2
CMP2 CH R2, BGNA ANGLE > BGNA?
    BL DETECT  NO, TRY AGAIN
WAIT
BAL R13, WREAD YES, GET READY
LR R1, R2

LH R2, B(R15) GET NEXT AZIMUTH
NHI R2, X'FFF' AND OUT UNWANTED BITS
CR R1, R2
BL CW3
LIS R3, 0

CMP6
CH R3, CW
    BNE DETECT
    OR R3, R3
    BZ CCW2
    THI R5, 2 CASE 2?
    BNZ WCCASE2 YES, BRANCH
CMP3 CH R2, BGNA ANGLE > BGNA?
    BL WAIT
READ1
    LH R2, R2(R15)
    NHI R2, X'FFF'
    CHI R2, 681
    BP ZER0
ELDET
    CH R2, ELEV1
    BM DETECT
    CH R2, ELEV2
    BP DETECT
    AIS R6, 1 INCREMENT COUNTER
    BNE DETECT DO IT TWICE TO BE SURE!
READ
    BAL R13, GOREAD YES, START READING
    LH R2, PPRRANG
    NHI R2, X'FFF'
    OR R3, R3
    BZ CCW1 BRANCH IF COUNTER CLOCKWISE
    THI R5, 1 CASE 1?
    BNZ CWCASE1 YES, BRANCH
    CH R2, ENDA ANGLE > ENDA?
    BNL DONE YES, ALL DONE
    B READ NO, KEEP READING
CW3 LIS R3, 15
    B CMP6
ZERO LIS R2, 0
    B ELDET
    59.5 DEG = 0
    B WCCASE1
    CH R2, BGNA ANGLE < BGNA?
    BNL READ YES, KEEP READING
    CH R2, ENDA NO, ARE WE DONE YET?
    BNL DONE YES
    B READ NO, KEEP READING
CCW1 THI R5, 1 CHECK FOR CASE 1
    BNZ CWCASE1 ANGLE < ENDA?
    CH R2, ENDA YES, ALl DONE
    BL DONE NO, CONTINUE
    B READ NO, KEEP READING
CCWCase1 CH R2, BGNA ANGLE > BGNA?
    BL READ NO, KEEP READING
    CH R2, ENDA YES, CHECK FOR FINISHED
    BL DONE
    B READ
CCW2 THI R5, 2
    BNZ WCCASE2
    CMP4 CH R2, BGNA OK
    BNL WAIT NOT YET
    B READ1
    WCCASE2 CHI R2, X'800'
    BNL CMP4
    AHI R2, X'1000'
CCWCASE2 CHI R2, $X'800'$
BNL CMP2
AHI R2, $X'1000'$
B CMP2

WCHCASE2 CHI R2, $X'800'$
BL CMP3
SHI R2, $X'1000'$
B CMP3

EDETECT BAL R13, WREAD
LH R2, 12(R13)
NHI R2, $X'0FF'$
CHI R2, 681
BP ZERO1

CPEV CH R2, ELEV1
BL EDETECT
CH R2, ELEV2
BP EDETECT
LH R2, PPRANG
NHI R2, $X'0FF'$

READ2 BAL R13, GOREAD
LH R4, PPRANG
NHI R4, $X'0FF'$

NXT OR R3, R3
BZ CCLK0K4
CR R4, R2
BL NXT2
B READ2

ZERO1 LIS R2, 0
B CPEV

FUDGE LIS R2, 6
B READ2

NXT2 BAL R13, GOREAD
LH R4, PPRANG
NHI R4, $X'0FF'$

OR R3, R3
BZ CCLK0K4
CR R4, R2
BP DONE
B NXT2

CC0K4 CR R4, R2
BP NXT2
B READ2

CCW4 CR R4, R2
BL DONE
B NXT2

DONE LH R15, PPR1
NHI R15, $X'0FF'$
STH R15, PPR1
SVC 1, OUTBLK
LH R0, OUTBLK+SVC1. STA
BNZ ERROR
LM 0, RSAVE

#FORT RETURN
#ASSM
WREAD SVC 1, WAITREAD
LH R0, WAITREAD+SVC1 STA READ RETURNED STATUS
BNZ ERROR
BR R13
GOREAD SVC 1 PPR1BLK2
LM 0 RSAVE

C-8
DECOD(I) = PPR1(I)
JSIZ = NRC/NITG
IMX = JSIZ + 1
K = 5
DO 10 I = 1, JSIZ
  REF = 0
  VEL = 0
  DO 20 J = 1, NITG
    RAW = PPR1(K)
  K = K + 1
  10 CONTINUE
  IF (REF .LE. 0) GO TO 111
  REF = REF .+ 0390625-DB
  IF (REF .LT. -39) REF .= REF .+ 100
  IREF(I) = REF
  VEL = VEL .+ BITVEL / 128.
  IVEL(I) = VEL - IVEL(I)
  IVEL(I) = VEL
  GO TO 101
  111 IREF(I) = 0
  IVEL(I) = 0
  101 CONTINUE
  STM 0, RSAVE
  SVC 1, OUTBLK
  SVC 1, WAITBLK
  SVC 1, PPR1BLK
  LM 0, RSAVE

DO 40 I = 1, 4
  40 CONTINUE
  IF (REF .LE. 0) GO TO 411
IF (REF. LT. -39) REF = REF + 100
IREF2(I) = REF
VEL = VEL + BITVEL / 128.
IVEL2(I) = VEL - IVEL(I)
IVELL(I) = VEL
GO TO 201

411 IREF2(I) = 0
IVEL2(I) = 0
IVELL(I) = 0
201 CONTINUE

*ASSM

STM 0.RSAVE
SVC 1.OUTBLK2 OUTPUT LAST AZIMUTH
SVC 1.WAITBLK WAIT FOR READ TO FINISH
LH R0.PPRIBLK2+SVC1.STA READ STATUS
BNZ ERROR IF NOT ZERO, ERROR
LH R0.PPRIBLK+SVC1.STA
BNZ ERROR
LH R0.OUTBLK+SVC1.STA
BNZ ERROR
LH R0.OUTBLK2+SVC1.STA
BNZ ERROR
LH R6.NAZ
AIS R6.2 ADD TWO TO AZIMUTH CTR
STH R6.NAZ
CHI R6.446 TOO MANY AZIMUTHS?
BLR R6.4 NOT, KEEP GOING
LIS R6.2 YES, INDCTR=2
STH R6.INDCTR
B DONE QUIT
ERROR SVC 2.ERRCODE CONVERT ERROR CODE
SVC 2.ERRBLOK OUTPUT MSG TO CONSOLE
SVC 2.PAUSE TASK PAUSED
LM 0.RSAVE RESTORE FORTRAN REGISTERS

***$P2 IS FORTRAN STMT NO 2
B $P2 START OVER
PAUSE EQU *
DB 0.1 PAUSE
ALIGN 4
ERRCODE EQU *
DPT DB 0 CODE 6
DB 6
DC H'0'
DC A(ERCD) DESTINATION
ALIGN 4
ERRBLOK EQU *
DB 0.7 PRINT CONSOLE MSG
DC H'15' CODE 7
DC A(PPRI) PRINT 15 CHRS
ERCD DC C' ERROR CODE
DC C'I/O ERROR'
ALIGN 4
WAITBLK EQU *
DB X'06' WAIT ONLY
DB 10 LU 10
DB 0.0
DSF 5
ALIGN 4
WAITREAD EQU *
DB X'59' READ AND WAIT
DB 10 LU 10
DB 0.0
DC A(PPRI)
DC C(PPRI) C-10
DSF 3
ALIGN 4
PPRIBLK2 EQU *
DB X'51' READ
DB 10 LU 10
DB 0.0
DC A(PPRI2)
DC A(PPRI2)
DSF 3
OUTBLK2 EQU *
DB X'31' READ
DB 9
DB 0.0
DC A(DECOD2)
DC A(DECOD2)
DSF 3
ALIGN 4
PPRIBLK EQU *
DB X'51' READ
DB 10
DB 0.0
DC A(PPRI)
DC A(PPRI)
DSF 3
ALIGN 4
OUTBLK EQU *
DB X'31' WRITE
DB 9
DB 0.0
DC A(DECOD)
DC A(DECOD)
DSF 3
RETURN
END
* 07/19/79 16:33:21
***LISTING FOR ERT1:REALTM. FTN
$N
$ASSM
ERTRMRP PROG
$FORT
$TITL FILE TSERMAP -- PRINT OUT DATA FIELDS-CHANGED FOR ERT READ BY CLB
SUBROUTINE RMAP
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 NRC, VAR, STORE(10), PLACE, OLDDATA
INTEGER*4 ANC(1028), IDTIME, IZT, ITZ, IZB, IZS
+ SRC(1028), DST(1028)
INTEGER*2 RH02, ZTH2, BETA2, PCT2, SIGMA2, ANG2, CHANGE
INTEGER*2 TWENTY, HOUR, MINUTE, SECOND, TP, ELEVAT, AZ
INTEGER*2 AZIM, TC, TA, T
INTEGER*2 Y, I, THETAl, NRC1, II, THETA, RHO, STOP, GRND
INTEGER*2 MEAN, POWER, SIGMA, TP2, TP3, SEGNO, Q, J, ZTH, MMU
INTEGER*2 BEGIN, SUM1, SUM2, JMIN, M, K, L, I
INTEGER*2 ZTH, ZTH2, BETA2, PCT2, MMU2, SIGMA2, ANG2, CHANGE
INTEGER*2 TWENTY/iS/, TWO/2/
REAL RNN, RRAREA
REAL PCTMIN, AZT. ELEVAA, BGNA, AZCHK, ENDA
INTEGER*2 FLAG, IQ, IB, IE, IM
INTEGER*2 DBB, SLOPE
INTEGER*4 RSAVE(16), RISAV
* COMMON /TLIS/- T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, IMIN, ISEC
COMMON /CALR/- SRC, DST, IREF(1024)
COMMON /SWITCH/- ICl(44), IC2(44), TC(1980), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1980), ITPC(44),
+ NEMB, NENEC, NAC
COMMON/AZ2/-SINA, COSA, DELTAZ, ISCANF, NEL, RI, SA
COMMON/AY2/-AZT, AZLAST, AZSTAR, NA, ELEVAA
COMMON/REI/-RE(1625), HR(258), NCL, NID, NIDP, INCL
X, IMX, IIMX, TL1, TL2, ROQUANT, IDVEL(258)
COMMON /PNTRS/- NCMX, NVMIN, NNUMX, IELEN, NSCAN, IESNL, NVSCN, NT
COMMON /INTL/- MHSN, MNSN, HM, FNSN
COMMON /EXTRA/-RHO, GRND, ZTH, BETA, K, RRAREA(12, 24), RH02, ZTH2, BETA2,
1PCT2, MMU2, SIGMA2, MINUT2, ANG2(6), CHANGE
COMMON/ZSTORE/ANC
COMMON /ZLOOK/- ZLOOKF, ZARY(91)
COMMON /ZCONST/- EARTH, VMK
COMMON /ZMAPPAR/- DAY, HOUR, MINUTE, SECOND, DBB, MAXV, MAXS, IOUT, SLOPE
COMMON /ZCNT/- COSPHI, SINPHI, COSPHI
COMMON /ZDATA1/- ECL(224), NOFST, KOFST, ICLAD, N1N1
COMMON /ZDATA2/- VLC(736), MXVC, NVC1
COMMON /ZDATA3/- VR(192), MXVR, NVR1
COMMON /ZVLS/- NVARM, NCARM, NV, ICO, IO, JO, JVR, KTL
COMMON /FILTER/-_TATRMN, AREAMN, DAZM
EQUIVALENCE(ANC(5), JREF(2)), (ANC(517), JIDVEL(2))
DATA TWENTY/16/
DATA TWO/2/
DATA PCTMIN/ 05/, MMU/0/, SIGMA/0/, OLDDATA/0/
DATA ZERO/0/

C ******** ** **** INITIALIZE ARRAY

C

T=0
AZLAST=-999
DO 901 J=1, MXVC
901 VLC(J)=0.
902 VR(J)=0.
903 RE(1)=0
904 ICC=0
905 CALL CONMSG(7, 'TSERMAP')
906 TL1=TL1/RQUANT
907 TL2=TL2/RQUANT
908 REWIND 4
909 REWIND 8
910 DO 3 IX=1,91
911 II=IX-IZOFF
912 ZX=FLOAT(II)/10.
913 3 ZARY(IX)=10. *ZX
914 00001 CONTINUE
915 00750 FORMAT(1X, 'ENTER PRF. 0=768. 1=922. 2=1075. 3=1229')
916 RCKM= 075
917 FLAG=0
918 333 FORMAT(13)
919 READ(7, 333)OLDATA
920 01002 FORMAT('AZIM ELEV RAN ',8(' REF VEL STD'))
921 REWIND 9
922 K=0
923 10000 CONTINUE
924 *ASSM
925 R0 EQU 0
926 R1 EQU 1
927 R2 EQU 2
928 R3 EQU 3
929 R4 EQU 4
930 R5 EQU 5
931 R6 EQU 6
932 R7 EQU 7
933 R8 EQU 8
934 R9 EQU 9
935 R10 EQU 10
936 R11 EQU 11
937 R12 EQU 12
938 R13 EQU 13
939 R14 EQU 14
940 R15 EQU 15
941 STM R0, RSAVE
942 FREZE
943 COPY SVC1.
944 L R3, ANCBLK+SVC1. SAD ANCBLK START
945 AIS R3,15 READ ANCILLARY ONLY
946 ST R3, ANCBLK+SVC1. EAD
947 SVC 1, ANCBLK THE READ
948 LH R0, ANCBLK+SVC1. STA
949 BNZ ERROR
950 LIS R1,0
951 STH R1, STOP
952 LIS R5, 0 WORKING REGISTER
953 LIS R9, 10 MULTIPLICAND
954 LHI R11, 100 MULTIPLICAND
955 L R0, ANC(R1) DATA
956 STBR R0, R5 HOUR
957 LIS R10, 15 MASK
958 NR R10, R5 1 HOUR
959 SRLS R5, 4 10 HOUR
960 NHI R5, 3
961 MHR R5, R9 (*10)
962 AMR R5, R10 TOTAL
963 STH R5, HOUR HOURS
964 EXBR R0, R0
965 STBR R0, R5 MINUTES
966 LIS R10, 15 MASK (-13)
NR  R10, R0
SRLS R5, 4 10 MINUTE
NHI R5, 7 MASK
MHR R5, R9 (+10)
AHR R5, R10 TOTAL MINUTES
STH R5, MINUTE
EXHR R0, R0
STBR R0, R5 DAYS
LIS R10, 15 MASK
NR R10, R5 1 DAY
SP: S R5, 4 10 DAY
MHR R5, R9 (+10)
AHR R5, R10
SRLS R0, 6 100 DAYS
NHI R0, X'F' MASK
MHR R0, R11 (+100)
AHR R5, R0 TOTAL DAYS
STH R5, DHR DAYS
AI S R1, 4 NEW DATA
L R0, ANC(R1)
LHI R10, 256
STH R10, NRC1 STORE
LHI 10, 1028+4+1
R 10, NRCBLK+SVC1 SAD
ST 10, NRCBLK+SVC1 EAD STORE END ADDRESS FOR READ
ST 10, NRCEAD STORE AWAY FOR SECOND READ
NHI R5, X'CO' MASK TP
SRLS R5, 6
STH R5, TP STORE(UNFIXED)
EXHR R0, R0
STBR R0, R5 SECONDS
LIS R10, 15 MASK
NR R10, R5 1 SECOND
SRLS R5, 4
NHI R5, 7 MASK 10 SECONDS
MHR R9, R5 (+10)
AHR R9, R10 TOTAL SECONDS
STH R9, SECOND STORE
AI S R1, 8 MORE DATA
LHL F0, ANC(R1) ELEVATION
NHI F0, X'FFF' ANGLE
STH F0, ELEVAT STORE ANGLE
LM R0, RSAVE

$FORT
NRC=NRC1
GRND=PCTMIN=NRC
REWIND 9
17 ELEVEN=0
TA=T
DR=(2**TP)*RCKM
K=K+1
NR:1=NRC
10001 CONTINUE
$ASM
STM R0, RSAVE READ IN DATA
SVC 1, NRCBLK
LH R0, NRCBLK+SVC1 STA LOAD IN STATUS
BNZ ERROR IF NOT ZERO, I/O ERROR
LIS R1, 0
LH R0, ANC(R1)
BM MINUS IF STOP=0
LIS R1, 1 STORE 1
STH R1, STOP
MINUS LIS R1, 8 GET AZIMUTH
LH R0, ANC(R1) ANGLE
NHI R0, X'FFF' C-14 MASK
DIM R1, 4
AIS R1, 4
LHL RO, ANC(RI) ELEVATION
NHI RO, X'FFF' ANGLE
STH RO, ELEVAT STORE IT
$FORT
DO 101 I=IMX, IMX
RE(I)=JREF(I)
JDVEL(I)=JDVEL(I)
101 CONTINUE
00500 CONTINUE
ICE=ICE+1
ELEVAA=ELEVAT/11. 37778
IELSN=IFIX(ELEVAA)
T=((DAY*24+HOUR)+MINUTE)*60+SECOND
AZT=AZ/11. 37778
FLAG=FLAG+1
A=AZT* 0.01743
AZCHK=AZT
IF(K. NE. 1) GO TO 105
BGNA=AZT
ENDA=AZT+359.
105 CONTINUE
IF(K. LT. 100) GO TO 106
IF(AZT.GT. 100.) GO TO 106
AZCHK=AZT+359.
106 CONTINUE
IF(AZCHK.GT. ENDA) K=1
SINA=SIN(A)
COSA=COS(A)
DELT=0. 019187
NA=K
RE(<258)=0
IF(NA. EQ. 1) NAC=1
NAC=NAC+1.
IF(NAC.GT. 1) NAC=0
IF(NA. NE. 1. OR. AZLAST.LT.-990.) GO TO 224
IDAY=DAY
IHR=HOUR
IMIN=MINUTE
ISEC=SECOND
PHI=ELEVAA+DAZM
COSPH=COS(PHI)
SINPH=SIN(PHI)
COSPH2=COSPH*COSPH*EARTH
CALL TRACK
224 CALL CONTOR
AZLAST=AZT
IF(STOP. NE. 1) GO TO 17
800 CALL STRAK
STOP
$ASSM
ALIGN 4
ERROR SVC 2. ERRCODE DECODE ERROR BITS
SVC 2. ERRBLOK OUTPUT ERROR MSG TO CONSOLE
SVC 2. PAUSE
LM 0. RSAVE RETURN TO FORTRAN
$FORT
IF(K. EQ. 0) GO TO 10000
IF(K. EQ. 1) GO TO 10001
GO TO 1
10003 CONTINUE
$ASSM
ALIGN 4
ERRCODE EQU *
DB 0
C-15
DB 6
DC H'0'
DC A<ERCD>
ALIGN 4

ERRBLOK EQU *
DB 0
DB 7
DC H'28'

ERCD DC C'
DC C'I/O ERROR IN RTN RANGE'
ALIGN 4

PAUSE EQU *
DB 0.1
ALIGN 4

ANCBLK DB X'59'
DB 9
DB 0.0
DC A(ANC) START ADDRESS
DC A(ANC) END ADDRESS
DSF 3

NRCBLK DB X'59'
DB 9
DB 3.0
DC A(ANC) START ADDRESS
DC A(ANC) END ADDRESS
DSF 3

$FORT
END
APPENDIX D

POST-MISSION ANALYSIS VERSION

(note: BLOCK DATA, CONTOUR, PEAKD, TRACK, ATRAK, STRAK, COMPAR, RESOLV, COMBIN and STRAK are identical in both versions)
SQUEZ

$ASSM
SCAT
SQUEZ C
TS DATA PROG

$TITL FILE TS DATA - DATA INPUT SUBROUTINE FOR ETSE
SUBROUTINE PPRDSC(ANG)
IMPLICIT INTEGER*2 (A-Z)
INTEGER*4 PPRI1(1024), PPRI2(1024), RSAVE(16)
INTEGER*4 OUT(260), OUT2(260)
INTEGER*4 PPRANG
INTEGER*2 ANG(6)
EQUIVALENCE (PPRANG, PPRI1(3), PPRI2(2))
EQUIVALENCE (OUT(1), PPRI1(1), OUT2(1), PPRI2(1))
COMMON /SECTOR/ INDCTR
COMMON /ZSTORE/ PPRI
COMMON /RUNSUM/ PPRI2
AVEN=3
CALL CONMSG(6, 'PPRDSC')
2 RE WIND 9
N A Z=0
ELEV1=ANG(4)+11 37778
ELEV2=ANG(5)+11 37778
IF (INDCTR EQ. 2) INDCTR = 0
IF (INDCTR .EQ. 0) GO TO 1
CW = ANG(3) - ANG(2)
IF (CW .GT. 180) CW = CW - 360
IF (CW .LT. -180) CW = 360 + CW
IF (CW .LT. 0) CW = 0
IF (CW .GT. 0) CW = 15
BGNA = 11 37778 * ANG(2)
ENDA = 11 37778 * ANG(5)
BGNA=MOD(BGNA, 4096)
ENDA=MOD(ENDA, 4096)
IF (ENDA GT. 4096) ENDA=0
STPFLG = 0
IF (CW .GT. 0) AND. ENDH .LT. BGNA) STPFLG = 1
IF (CW .EQ. 0) AND. ENDH .GT. BGNA) STPFLG = 1
IF (BGNA GT. 10 AND. BGNA LT. 4086) GO TO 1
BGNA=0
STPFLG=2
1 CONTINUE
10000 CONTINUE

$ASSM
FREIZE
CROSS
COPY SVC1.
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
LET ONLY 256 WORDS OF VIDEO OUT

**SI R4,2048**

**ST R4,OUTBLK+SVC1,EAD**

**L R4,PPRIBLK2+SVC1.SAD GET NEXT BEGINNING ADDR**

**AR R4,R0 COMEPT END ADDRESS**

**ST R4,PPRIBLK2+SVC1,EAD STORE IN EAD**

**ST R4,WAITREAD+SVC1,EAD SAME NUMBER**

**LET ONLY 256 WORDS OF VIDEO OUT**

**SI R4,2048**

**ST R4,OUTBLK2+SVC1,EAD**

**LDRI R15,PPRI**

**LDRI R14,PPR12**

**LCS R6,1 SET R6 TO -1 FOR COUNTER**

**DETECT**

**BAL R13,WREAD READ IN AN AZIMUTH**

**LH R1.8(R15) LOAD IN AZIMUTH DATA**

**NHI R1,X'FFFF AND OUT UNWANTED BITS**

**BAL R13,WREAD GET ANOTHER AZIMUTH**

**LH R2.8(R15) GET AZIMUTH DATA**

**NHI R2,X'FFFF AND OUT UNWANTED BITS**

**CR R1.R2 COMPARE TWO AZIMUTHS**

**BL CWISE IF R1<R2, RADAR IS GOING CWISE**

**LIS R3,0 DIRECTION FLAG**

**B WHATIZIIT CONTINUE**

**LIS R3,15 DIR 15LAG = CW**

**WHATIZIT LH R4,INDCTR SECTOR SCAN OR FULL CIRCLE?**

**BJEDETECT FULL CIRCLE**

**TH R3,CW IS DIRECTION OF ROTATION CORRECT?**

**BNE DETECT WRONG DIRECTION, WAIT**

**OP R2.R2 WHICH DIRECTION IS IT?**

**BZ CCW COUNTER CLOCKWISE**

**LH R5,STPFLG**

**THI R5,2 CASE 2?**

**CMP1 CH R2,BGNA ANGLE < BGNA?**

**BL WAIT YES, GET READY**

**B DETECT NO, TRY AGAIN**

**CWCA9E2 CHI R2.X'800 ANGLE > 180?**

**BL CMP1 NO, ALL OK**

**SHI R2.X'1000 YES, SUBTRACT 360**

**B CMP1**

**CCW LH R5,STPFLG**

**THI R5,2 CHECK FOR CASE 2**

**BNC CWCA9E2**

**CMP2 CH R2,BGNA ANGLE > BGNA?**

**BL DETECT NO, TRY AGAIN**

**WAIT**

**PAI R13,WREAD NO, GET READY**
LH R2.8(R15) GET NEXT AZIMUTH
NHI R2.X'FFF' AND OUT UNWANTED BITS
CR R1.R2
BL CW3
LIS R3.0
CH R3.CW
BNE DETECT
OR R3.R3
BZ CCW2
THI R5.2 CASE 2?
BNZ NWCASE2 YES, BRANCH
CMP3 CH R2.BGNA ANGLE > BGNA?
BL WAIT NO, KEEP WAITING
READ1 LH R2.12(R15)
NHI R2.X'FFF'
CHI R2.681
BP ZERO
ELDET CH R2.ELEV1
BM DETECT
CH R2.ELEV2
BP DETECT
AIS R6.1 INCREMENT COUNTER
BNP DETECT DO IT TWICE TO BE SURE!
READ BAL R13.GOREAD YES, START READING
LH R2.PPRANG
NHI R2.X'FFF'
OR R3.R3
BZ CCW1 BRANCH IF COUNTER CLOCKWISE
THI R5.1 CASE 1?
BNZ CWCASE1 YES, BRANCH
* WANT SEVERAL PPIS
B READ
CH R2.ENDA ANGLE > ENDA?
BNL DONE YES, ALL DONE
B READ NO, KEEP READING
CCW1 LIS R3.15
B CMP6
ZERO LIS R2.0 59.5 DEG = 0
B ELDET
CWCASE1 CH R2.BGNA ANGLE < BGNA?
BNL READ YES, KEEP READING
* WANT SEVERAL PPIS
B READ
CH R2.ENDA NO, ARE WE DONE YET?
BNL DONE YES
B READ NO, KEEP READING
CCW1 THI R5.1 CHECK FOR CASE 1
BNZ CWCASE1
CH R2.ENDA ANGLE < ENDA?
BL DONE YES, ALL DONE
B READ NO, CONTINUE
CWCASE1 CH R2.BGNA ANGLE > BGNA?
BL READ NO, KEEP READING
CH R2.ENDA YES, CHECK FOR FINISHED
BNL DONE
B READ
CCW2 THI R5.2
BNZ NWCASE2
CMP4 CH R2.BGNA
BNL WAIT OK
B READ1 NOT YET
WWCASE2 CHI R2.X'800'
BNL CMP4
AHI R2.X'1000'
P CMP4 0-4
CCWCASE2 CHI R2,X'800' ANGLE < 180? BNL CMP2 NO. ALL OK
AHI R2,X'1000' YES, ADD 360
BNL CMP2 NOW CHECK FOR BEGINNING
WCWCASE2 CHI R2,X'800' ANGLE > 180? BL CMP3 NO. ALL OK
SHI R2,X'1000' YES. SUBTRACT 360
BNL CMP3

EDetect BAL R13,WREAD READ IN NEW AZIMUTH
LH R2.12(R15) READ IN ELEVATION
NHI R2,X'FFF' AND OUT UNWANTED BITS
CHI R2.681
BP ZERO1
CPEV CH R2.ELEV1
BL EDETECT
CH R2,ELEV2
BP EDETECT WITHIN RANGE?
LH R2.PPRANG YES, GET AZIMUTH
NHI R2,X'FFF'
CHI R2.6
BL FUDGE

READ2 BAL R13,GOREAD START READING
LH R4,PPRANG
NXT OR R3.R3
BZ CCLK4 COUNTERCLOCKWISE
CR R4.R2
BL NXT2
B READ2
ZERO1 LIS R2.0
B CPEV
FUDGE LIS R2.6 IF 0, MAKE IT 6
B READ2
NXT2 BAL R13,GOREAD GET NEW AZIMUTH
LH R4,PPRANG
NHI R4,X'FFF'
OR R3.R3
BZ CCW4
* WANT SEVERAL PPI S
B NXT2
CR R4.R2 ANG > END? B DONE YES, FINISHED
B NXT2 NO, KEEP READING
BZ CCLK4 CR R4.R2
B READ2
BZ CCW4 CR R4.R2
B DONE
BL NXT2
DONE LH R15,PPRI
NHI R15,X'0FF'
STH R15,PPRI
SVC 1,OUTBLK
LH R0,OUTBLK+SVC1.STA
BNZ ERROR
LM 0,RSAVE

$FORT RETURN
$ASSM
ALIGN 4
WREAD SVC 1,WAITREAD
SVC 1,WAITREAD+SVC1.STA READ RETURNED STATUS
BNZ ERROR IF NOT ZERO, ERROR
BR R13
GOREAD SVC 1,PPRIBLK2 READ IN ONE AZIMUTH
LISK R12,12 COUNTER FOR MAIN BUFFER
LHI R4.16 COUNTER FOR NEW BUFFER
AVE LISK R8.0 ZERO POWER ACCUMULATOR
LISK R10.0 ZERO VELOCITY ACCUMULATOR
LISK R1.3 AVERAGING COUNTER
AVER LHL R7, PPR (R4) LOAD POWER HALFWORD
NHI R7, X'1FF' GET ONLY THE POWER
AR R8, R7 ADD TO ACCUMULATOR
AIS R4.2 INCREMENT ARRAY POINTER
LHL R7, PPR (R4) LOAD vel/VAR HALFWORD
NHI R7, X'FFFF' INCREMENT FOR NEXT HALFWORD
AIS R4.2 INCREMENT FOR NEXT HALFWORD
SIS R1.1 SUBTRACT FOR INNER LOOP
BNZ AVER BRANCH IF NOT 3 ADDED UP
DH R8, AVEN DIVIDE POWER
NHI R9, X'1FF' SLA R9.16 SHIFT POWER TO LEFT
SRA R10.8 DH R10, AVEN DIVIDE VELOCITY
SLA R11.8 NHI R11, X'FFFF' GET ONLY LOWER HALFWORD
AR R9, R11 PUT TOGETHER
AIS R12, 4 INCREMENT NEW ARRAY POINTER
ST R9, PPR (R12) PUT INTO ARRAY
CLHI R4, 3072 CLHI R4,3072
BP AVEEND
AVEEND

AVEEND
SVC 1, OUTBLK OUTPUT L A T AZIMUTH
SVC 1, WAITBLK WAIT FOR FIRST READ TO FINISH
SVC 1, PPR (BLK READ NEXT AZIMUTH
SVC 1, OUTBLK2 OUTPUT LAST AZIMUTH
SVC 1, WAITBLK WAIT FOR READ TO FINISH
LH F0, PPR (BLK + SVC 1, STA READ STATUS
BNZ ERROR 1F NOT ZERO, ERROR
LJ R0, PPR (BLK + SVC 1, STA
BNZ ERROR
LJ F0, OUTBLK + SVC 1, STA
BNZ ERROR
LJ F0, OUTBLK2 + SVC 1, STA
BNZ ERROR
LJ R6, NAZ
AIS R6, 2 ADD TWO TO AZIMUTH CTR
STH R6, NAZ
CHI R6, 1760 TOO MANY AZIMUTHS?
BLR R13 NO, KEEP GOING
LIS R6, 2 YES, INDCTR=2
STH R6, INDCTR
B DONE QUIT
ERROR SVC 2, ERR (CODE CONVERT ERROR CODE
SVC 2, ERR (BLOK OUTPUT MSG TO CONSOLE
SVC 2, PAUSE TASK PAUSED.
LM 0. R SAVE RESTORE FORTRAN REGISTERS

***$P2 IS FORTRAN STMT NO. 2
B $P2 START OVER
PAUSE EQU *
DB 0.1 PAUSE

ALIGN 4

ERRCODE EQU *
OPT DB 0
DB 6 CODE 6
DC H'0'
DC A (ERCD) DESTINATION
ERRBLOK EQU *
PRINT CONSOLE MSG
DB 0.7
PRINT 15 CHR$ CODE 7
DC H'15'
PRINT 15 CHRS
DC C'
ERROR CODE
DC C'/I/O ERROR/
ALIGN 4

WAITBLK EQU *
DB X'08'
WAIT ONLY
DB 10
DB 0.0
DSF 5
ALIGN 4

WAITREAD EQU *
DB X'59'
READ AND WAIT
DB 10
DB 0.0
DC A(PPRI)
DC A(PPRI)
DSF 3
ALIGN 4

PPRIBLK EQU *
DB X'51'
READ
DB 10
DB 0.0
DC A(PPRI2)
DC A(PPRI2)
DSF 3
ALIGN 4

OUTBLK EQU *
DB X'31'
WRITE
DB 9
DB 0.0
DC A(OUT2)
DC A(OUT2)
DSF 3
ALIGN 4

OUTBLK EQU *
DB X'31'
WRITE
DB 9
DB 0.0
DC A(OUT)
DC A(OUT)
DSF 3

RETURN
END
*07/19/79 12:49:01
***LISTING FOR ERT1:CRANE.FTN
$N
$ASSM
ERTRMAP PROG
$FORT
$TITL  FILE TSE RMAP - PRINT OUT DATA FIELDS-CHANGED FOR ERT READ BY CLB
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 NRC, VAR, STORE(10), PLACE, OLDATA
INTEGER*4 ZEE(1024), ANC(1024), IDTIME, IZF, ITZ, IZB, IZS
INTEGER*2 TWENTY, ELEVEN, DAY, HOUR, MINUTE, SECOND, TP, ELEVAT, AZ
INTEGER*2 AZIM, TC, TA, T
INTEGER*2 Y, I, THEtal, NRC1, II, THEtA, RH0, STOP, GRND
INTEGER*2 MEAN, POWER, SIGMA, TP2, TP3, SEGNO, Q, J, ZTH, MMU
INTEGER*2 BEGIN, SUM1, SUM2, JMIN, M, K, L, I
INTEGER*2 Y1, ZERO, TWO, BETA
INTEGER*4 RsAV(16), R1SAV
INTEGER*2 IREF(1024), IVEL(1024), IVAR(1024), IVELL(256)
+ , JVEL(256)
INTEGER*2 RE, HR, TL1, TL2, RQUANT
REAL RNN, RRAREA, AXX
REAL PCTMIN, AZT, ELEVAA, BGA, AZchk, ENDA
INTEGER*2 FLAG, IQ, IB, IE, IM
INTEGER*2 IOUT, DBB, MAXV, MAXS, SLOPE
INTEGER*4 NRCEAD
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, IMIN, ISEC
COMMON /CONST/ VMISW, DIV, VMAG, VMlSWM, ZDIV, ADR1, A2, A3, B1, B2, HDIV
COMMON /SWITCH/ IC1(44), IC2(44), TC(1080), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1980), IPTC(44),
+ NEMB, NEMC, NAC
COMMON/AZ2/SINA, COSA, DELTA2, ISCANF, NEL, RI, SA
COMMON/AZM/AZT, AZLAST, AZSTAR, NA, ELEVAA
COMMON/REFL/RE(1025), HR(258), NCL, NID, NIDP, INCL
X, IMX, IMM, TL1, TL2, RQUANT, JVEL(258)
COMMON /PNT15/ NCMX, NMV, NUMX, IELSN, NSCAN, IESNL, NVSCN, NT
COMMON /INTL/ MHSN, MNSN, HM, DNSN
COMMON/EXTRF/RHO, GRND, ZTH, BETA, K, RRAREA(288), RHO2(16), AXX(288),
1PRFM
COMMON/STORE/ANC
COMMON /ZLOOK/ IZOFF, ZARY(91)
COMMON /ECONST/ EARTH, VMK
COMMON /MAPPAR/ DAY, HOUR, MINUTE, SECOND, DBB, MAXV, MAXS, IOUT, SLOPE
COMMON /CST/ COSP, SINPHI, COSPH2
COMMON /DATA1/ ECL(224), NOFS, KOFST, ICLAD, NANI
COMMON /DATA2/ VCL(736), MXVC, NVC1
COMMON /DATA3/ VRC(192), MXVR, NVRI
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, ID, JYR, KTL
COMMON /FILTER/ TATRMN, AREAMN, DAZM
EQUIVALENCE(RE(2), IREF(1)), (IDVEL(2), JVEL(1))
EQUIVALENCE(ANC(5), ZEE(1))
DATA TWENTY/18/
DATA TWO/2/
DATA PCTMIN/ 05/, MMU/0/, SIGMA/0/, OLDATA/0/
DATA ZERO/0/

C C ********** INITIALIZE ARRAY
C
T=0
AZLAST=-999.
DO 901 J=1, MXVC
   981 VCL(J)=0.
   DO 902 J=1, MXVR
      982 VCL(J)=0.
VMISW = (VMISW - 1) / DIV
RE(i) = 0
JDVEL(1) = 0
JDVEL(256) = 0
ICC = 0

CALL CONMSG(7, 'TSERMAP')
TL1 = TL1 / RQUANT
TL2 = TL2 / RQUANT
REWIND 4
REWIND 8
DO 2 I = 1, 1024
    IREF(I) = 0
    IVEL(I) = 0
2
    DO 3 IX = 1, 91
        II = IX - 1200
        ZX = FLOAT(II) / 10.
3
    ZARY(IX) = 10. ** ZX
00001 CONTINUE
DB = 65. 28
FMT: FORMAT(IX, 'ENTER PRF, 0 = 768, 1 = 922, 2 = 1075, 3 = 1229')
RCKM = 0. 75
FLAG = 0
READ(7, 333) IOUT, DBB, MAXV, MAXS, SLOPE
333
    FORMAT(I3)
    READ(7, 333) OLDATA
01002 FORMAT('//', 'AZIM ELEV RAN ', B8< 'REF VEL STD')
    BITVEL = MAXV
    BITVAR = MAXS
    REWIND 9
    K = 0
10000 CONTINUE

$ASSM
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
STM R0, RSAVE
FREZE
COPY SVC1.
L R3, ANCBLK + SVC1. SAD, ANCBLK, START
AIS R3, 15, READ ANCILLARY ONLY
ST R3, ANCBLK + SVC1. EAD
SVC 1, ANCBLK, THE READ
LH R0, ANCBLK + SVC1. STA
BNZ ERROR
LIS R1, 0
STM R1, STOP
LIS R5, 0, WORKING REGISTER
LIS R9, 10, MULTIPLICAND
LHI R11, 100, MULTIPLICAND
L R8, ANC(R1), DATA
STBR R0, R5, HOUR 0.
NR R10.R5 1 HOUR
SRLS R5.4 10 HOUR
NHI R5.3
MHR R5.R9 (*10)
AHR R5.R10 TOTAL
STH R5.HOUR HOURS
EXBR R0.R0
STBR R0.R5 MINUTES
LIS R10.15 MASK
NR R10.R5 1 MINUTE
SRLS R5.4 10 MINUTE
NHI R5.7 MASK
MHR R5.R9 (*10)
AHR R5.R10 TOTAL MINUTES
STH R5.MINUTE
EXBR R0.R0
STBR R0.R5 DAYS
LIS R10.15 MASK
NR R10.R5 1 DAY
SRLS R5.4 10 DAY
MHR R5.R9 (*10)
AHR R5.R10 TOTAL DAYS
STH R5.DAY DAYS
AIS R1.4 NEW DATA
L R0.ANC(R1)
LIS R5.0 JUST TO BE SURE
STBR R0.R5 NRC AND TP
LIS R10.3 MASK
NR R10.R5 NRC
AIS R1.0.1
LHI R12.256 NRC=(NRC+1)*256
MHR R10.R12 LHI R10.256
STH R10.NRC1 STORE
SLHLS 10.2 MULTIPLY BY 4 FOR BYTE COUNT
AIS 10.15
A 10.NRCBLK+SVC1. SAD
ST 10.NRCBLK+SVC1. EAD STORE END ADDRESS FOR READ
ST 10.NRCEAD STORE AWAY FOR SECOND READ
NHI R5.X'CO' MASK TP
SRLS R5.6 STH R5.TP STORE(UNFIXED)
EXBR R0.R0
STBR R0.R5 SECONDS
LIS R10.15 MASK
NR R10.R5 1 SECOND
SRLS R5.4 NHI R5.7
MHR R5.R9 (+10)
AHR R5.R10 TOTAL SECONDS
STH R5.SECOND STORE
AIS R1.8 MORE DATA
LHL R0.ANC(R1) ELEVATION
NHI R0.X'FFF' ANGLE
STH R8.ELEVAT STORE ANGLE
LM R8.RSAVE

#FORT
NRC=NRC1
GRND=PCTMIN+1
NRC

REWORD 9
17 ELEVEN=8 D-10
DR=(2**TP)*RCKM
K=K+1
NRC1=NRC
10001, CONTINUE

*ASMM

STM R0,RSAVE
SVC 1,NRCBLK
LH R0,NRCBLK+SVC1 STA LOAD IN STATUS
BNZ ERROR IF NOT ZERO, I/O ERROR
LIS R1,0
LH R0,ANC(R1)
BM MINUS IF STOP=0
LIS R1,1
STH R1,STOP MINUS
LIS R1,0 GET AZIMUTH
LH R0,ANC(R1) ANGLE
NHI R0,'X'FFF' MASK
STH R0,AZ STORE

AIS R1,4
LHL R0,ANC(R1) ELEVATION
NHI R0,'X'FFF' ANGLE
STH R0,ELEVAT STORE IT
LHL R8,NRC1 MAX. NO. OF CELLS
LHI R3,32767 MAX. HALFWORD
LDAI R15,GRND
LHL R1,0(R15) CELL COUNTER
LIS R2,0 SUM 2R
LIS R4,7 FOR
LIS R5,0 SHIFTING
LIS R6,0 SUM 2RZ/128
LDAI R15,2TH
LHL R7,0(R15) MIN. POWER
LIS R9,0 CELLS PER SEG.
LDAI R15,RHO
LHL R10,0(R15) MIN. SEGMENT SIZE
STH R10,PLACE INITIALIZE PLACE
SIS R10,1
LHL R11,MMU MINIMUM MEAN
LHL R12,SIGMA MINIMUM VARIANCE

*MAIN LOOP

* LOOP AR R1,R10 NEXT CELL
   NLOOP AIS R1,1 IF DONE,
   CR R1,R8 LEAVE
   BNL REALLY
   POW LR R15,R1
   SLLS R15,2
   L R5,ZEE(R15) RAW DATA
   EXHR R0,R5 POWER
   LH R2,0LDATA
   BNZ NO9
   NHI R0,'X'1FF' STORE REFLECTIVITY
   STH R0,IREF(R1) PUT MEAN IN TOP OF R15
   CONT EXHR R15,R5
   SRA R15,24
   BNMS POSITIVE ABSOLUTE VALUE
   XHI R15,'X'FFF' COMPLEMENT
   AIS R15,1
   STH R15,IVEL(R1) STORE VELOCITY
   POSITIVE NHI R5,'X'FF' VARIANCE
   STH R5,IVAR(R1) STORE VARIANCE
   B NLOOP
   NO9 NHI R0,'X'FF'
DATA STORING ROUTINE

REALLY LM RS, RSAVE

$FORT

DO 101 I=1,256
  IF (IREF(I).LE.0) GO TO 111
  IREF(I)=IREF(I)*100./256. -DB
  IREF(I).LT.-39) IREF(I)=IREF(I)+100
  IVEL(I)=IVEL(I)*BITVEL/128.
  IDVEL(I)=IVEL(I)-IVELL(I)
  IVEL(I)=IVEL(I)
C F4WORD=IVAR(I)*BITVAR/256.
C IVRI)=SQRT(F4WORD)
GO TO 101
00111 IREF(I)=0
  IVEL(I)=0
  IVAR(I)=0
101 CONTINUE

00500 CONTINUE
  ICC=ICC+1
  ELEVAR=ELEVAR/11.37778
  IELSN=IFIX(ELEVAR)
  T=(DAY*24+HOUR)*60+MINUTE)*60+SECOND
  AZT=AZ/11.37778
  FLAG=FLAG+1
  A=AZT*0.01743
  AZCHK=AZT
  IF(K. NE.1) GO TO 105
  BGNA=AZT
  ENDA=AZT+359.
105 CONTINUE
  IF(K.LT.180) GO TO 106
  IF(AZT.GT.180) GO TO 106
  AZCHK=AZT+359.
106 CONTINUE
  IF(AZCHK.GT. ENDA) K=1
  SINA=SIN(A)
  COSA=COS(A)
  DELTAZ=0.0191987
  NA=K
  RE(258)=0
  IF(NA.EQ.1)NAC=1
  NAC=NAC+1
  IF(NA.GT.1)NAC=0
  IF(NA.NE.1 OR AZLAST.LT.-990.) GO TO 224
  IDAY=DAY
  IHR=HOUR
  IMIN=MINUTE
  ISEC=SECONI
  PHI=ELEVAR+DAZM
  COSPHI=COS(PHI)
  SINPHI=SIN(PHI)
  COSPH2=COSPHI*COSPHI*EARTH
  CALL TRACK
223 IF(NSCAN.EQ.4) GO TO 800
224 CALL CONTOP
AZLAST=AZT
AZLAST=AZT
000 CALL STRAK
STOP

SASSM ALIGN 4
ERROR SVC.2. ERRCODE DECODE ERROR BITS
SVC 2, ERRBLUK
SVC 2, PAUSE
LM 0, RSAVE

*FORT
IF(K.EQ.0) GO TO 10000
IF(K.EQ.1) GO TO 10001
GO TO 1
10003 CONTINUE

*ASIM
ALIGN 4
ERRCODE EQU *
DB 0
DB 6
DC 'H'0'
DC A(ERCD)
ALIGN 4
ERRBLOK EQU *
DB 0
DB 7
DC 'H'28'
ERCD DC 'I/O ERROR IN RTN RANGE'
ALIGN 4
PAUSE EQU *
DB 0, 1
ALIGN 4
ANCBLK DB 'X'59'
DB 9
DB 0, 0
DC A(ANC)
DC A(ANC)
DSF 3
NRCBLK DB 'X'59'
DB 9
DB 0, 0
DC A(ANC)
DSF 3

*FORT
END
***LISTING FOR ERTI: INPARM.FTN

$N

BLOCK DATA

FOR PROGRAM EXTRAD ERT NO. 162

VERSION 4.0 LEVEL 780301

JHW IBM370

IMPLICIT INTEGER*2 (I-N)

IMPLICIT INTEGER*2 W, TL1, TL2, HR, ROQUANT, T, TM

COMMON /CDRAYS/ IC(32,10), C(32,9), ID(32,10), D<32,9>, IM, JM

COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, R1, A2, A3, B1, B2, HDIV

COMMON /VPARM/ VX, VY

COMMON /PNTRS/ NCMX, NVMIN, NUMX, IELSN, NSCAN, IESNL, NVSCN, NT

COMMON /ECONST/ EAIRTH, VMK

COMMON /NLIS15/ NVARM, NCARM, NVO, ICO, IO, JO, JYR, KTL

COMMON /ZLOOK/ IZOFF, ZARY(91)

COMMON /INTL/ MHSN, MNSN, HM, FNSN

COMMON /DVAL/ DELA

COMMON /CNTRS/ IATR(5), MATR

COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2

COMMON /TMAX/ TM

COMMON /AZM/ AZMUTH, AZLAST, AZSTAR, NA, ELEVAT

COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NANI

COMMON /DATA2/ VCL(736), MXVC, NVC1

COMMON /DATA3/ VR(192), MXVR, NVR1

COMMON/FILTER/ TAM(A), AREAMN, DAZM

COMMON /PRSTOR/ NUP, TATR(1400), NMAX, IACT(70),
+ IDC(22), IPRNG(34), KAMAX, MXTR

*COMMON/REL/ W(1025), HR(256), NCL, NIDP, INCL,
X IMX, IMN, TL1, TL2, ROQUANT, IDVEL(258)

COMMON/THRESH/LDV

LOGICAL PRI, PR2, PRIN2

DATA ECL/224*0., VCL/736*0., VR/192*0., 0/

DATA TL3/36/, TL2/60/

DATA NOFST/16/, KOFST/7/, ICLAD/112/

DATA DAZM/0 0174533/, ISCANF/0/

DATA LI/8., SA/900., ROQUANT/1/

DATA LDV/3/, MXVR/192/, NVR1/6/, MXVC/736/, NVC1/23/

DATA NFM/2/, IEMAX/22/, NFC/2/

DATA KAMAX/45/, JKMDB/80/, IMAX/34/, IAMAX/2700/, IR/15/, JR/45/,
+ IMXJM/60/, NCL/258/, KAMAX/990/

DATA NID/250/, NUP/4/, NIDP/70/, NMAX/20/, MXTR/1400/

DATA AREAMN/0.4/, IMX/257/, IMN/2/

DATA IZOFF/18/, DELA/0 0087/, FNSN/0 009/

DATA IATR/9, 8.5, 0.0/

DATA ITYPE/1/, PR1/, TRUE/., PR2/., FALSE/., PRIN2/, FALSE/.

DATA NVARM/32/, MNSN/5/, MHSN/7/, HM/6/5/

DATA TM/0/, NCARM/16/, VMK/1 E-3/

DATA IM/32/, JM/9/, NVO/0/, ICO/0/, IO/0/, JO/0/

DATA VMISW/5/, DIV/2/, VMAG/01/, ZDIV/1/, HDIV/5/


DATA VX/0/, VY/0/, IESNL/0/, IESN/0/

DATA EARTH/6 4057 E-5/, NT/0/, NSCAN/0/, NVSCN/1/

END
SUBROUTINE CONTOR

C JHN AFGL SUDBURY RADAR SUBROUTINE
C VERSION 4.1 LEVEL 79117
C FIND EVENTS ALONG SINGLE RADIAL
C
C IMPLICIT INTEGER*2 (I-N)
INTEGER *2 W, TL1, TL2, HR, ROQUANT, TC
+
, T, IC21(22), IC22(22)
COMMON /SWITCH/ IC1(44), IC2(44), TC(1960), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1980), IPTC(44),
+ NEMB, NEMC, NAC
COMMON/AZM/ AZMUTH, AZLAST, AZSTAR, NA, ELEVAT
COMMON /AZM/ SINA, COSA, DELTAZ, ISCANF, NEL, RI, SA
COMMON/FILTER/TATRMN, AREAMN, DAZM
COMMON /FIXED/ NPA, IEMAX, NFC, IEM, JEM
COMMON /PRSTOR/ NUP, TATR(1400), NUMAX, IACT(70),
+ IDC(22), IPRNG(34), KAMAX, MXTR
COMMON/REFL/ W(1025), HR(2548), NCL, NID, NIDP, INCL
+ X, IMX, INN, TL1, TL2, ROQUANT, IDVEL(258)
COMMON /THRESH/ KMAX, T(88), JMXDB, JMAX, IAMAX, IR, JR, IMXJMX
COMMON /PRSTOR/ NUP, TATRMN, AREAMN, DAZM
COMMON /CONPIC/ NCEL, NMR

* DATA RPD/.017453/

CALL CONMSG6, 'CONTOR')

IF (IFLAG .EQ. 1) WRITE(3,7) IX=1, 102

FORMAT (IX, 2015)

IF (NA .NE. 1) GO TO 61

初期化。

TATRMN=AREAMN*1. E06/SA
NEMC=IEMAX
NCEL=1
JEM=0

61 CONTINUE
DO 101 K=1, IEMAX
IDC(K)=0
101 CONTINUE
DO 102 J=1, JMAX
IPRNG(J)=0
102 CONTINUE
NEMB=NEMC
NEMC=NAC*IEMAX
NEM1=NEMC+1
IEM=0
IEM2=0
IP=0
IPB=0

FIND EVENTS

DO 281 I=2, NCL**

IF (W(I), GT, TL) GO TO 2311

W(I)=A
2311 IF (RQUANT.GT W(I).GT. W(I))/RQUANT
2311 IF (W(I).GT. TL1) GO TO 131
2311 GO TO 241
131 IF (W(I-1).LE. TL1) GO TO 141
131 GO TO 151
141 IEM=IEM+1
141 IEM+1
141 IEM=NEMC
141 IF (IEM.LE. IEMAX) GO TO 141
141 WRITE(3,1412) IEMAX, K
141
1412 FORMAT(1X, 39HEVENT COUNTER EXCEEDED MAX VALUE, IMAX=.16, 5X, 14) IEM=IEMAX
1411 IC1(IEA)=I-1
1411 IC2(IEA)=0
C C
C C
151 IF (W(I)-W(I-1)) 171, 181. 161
161 IPB=I-1
161 GO TO 181
171 IF (IPB.EQ.0) GO TO 181
171 IF (IPB.LE. JMAX) GO TO 171
171 WRITE(3, 1713) IP, IEVENT
171
1713 FORMAT(1X, 17HN PEAKS EXCEEDED, 216) IP=JMAX
171 GO TO 181
1711 IF (IPB.EQ.0) GO TO 181
1711 IF (IPB.LE. JMAX) GO TO 1711
181 CONTINUE
181 GO TO 282
241 IF (W(I-1). LE. TL1) GO TO 281
241 IC2(IEA)=I-1
C C
C C
2412 IF (W(I)-W(I-1)) 2412
2412 IF (W(I-1).GT. TL2) GO TO 281
2412 IEM2=IEM2+1
2412 IF (IEM2.GT. IEMAX) IEM2=IEMAX
2412 IC21(IEM2)=I-1
2412 IC22(IEM2)=0
2412 GO TO 281
2412 IF (W(I-1).LE. TL2) GO TO 281
2412 IC22(IEM2)=I-1
281 CONTINUE
281 CALL PEAKD
C C
C C
C C
C C IF (IEM.NE. 1) GO TO 802
C C WRITE(6:1) AZMUTH, TL2, IEM2, (IC21(J), IC22(J), J=1, IEM2)
C C 1 FORMAT(8.21BF./11(216.3X)/11(216.3X))
C 801 WRITE(8) AZMUTH, TL1, IEM, (IC1(J), IC2(J), J=NEM1, IEA)
C JEM=IEM
C RETURN
C END
SUBROUTINE PEAKD

C
C VERSION 5.0  LEVEL 780616
C JW  AVCO  IBM360
C DETERMINES PEAK VALUES AND THEIR ATTRIBUTES.
C
C **********************************************************************
C IMPLICIT INTEGER*2 (I-N)
REAL  UP(6), TATR(1400), BUF(8)
INTEGER*4 RSAVE(16), TCVL, TBVL, TCVLB, TCVM, TATC, TATM
INTEGER*2 HB, IACT(70), IDC(22),
+ IFCRNG(34), TM
INTEGER*2 W, RQUANT
INTEGER*2 TL2, T, TC
COMMON /DATAl/ ECL(224), NOFST, KOFST, ICLADJ, NANI
* COMMON /SWITCH/ IC1(44), IC2(44), TC(1980), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1980), IPTC(44),
+ NEMB, NEMC, NAC
COMMON /PWORK/ KMAX, T(80), JMXDB, JMAX, IAMAX, IR, JR, IMXJM
COMMON/REFL/ W(1025), HB(258), NCL, NIDP, NIDP, INCL,
X IMX, IMN, TM, TL2, RQUANT, IDVEL(258)
COMMON /FIXED/ NPA, IEMAX, NFC, IEM, JEM
COMMON/AZM/ AZMUTH, AZLAST, AZSTAR, NA, ELEVAT
COMMON /AZ2/ SAZ, CAZ, DAZ, ISCANF, NEL, RI, SA
COMMON/FILTER/ TATRMN, AREAMN, DAZM
COMMON /THRESH/ LDB
COMMON /PRSTOR/ NUP, TATR(1400), NUMAX, IACT(70),
+ IDC(22), IFCRNG(34), KAMAX, MXTR
COMMON /CONPK/ NCELL, NMX
COMMON /FILTER/ TATRMN, AREAMN, DAZM
COMMON /THRESH/ LDB
COMMON /PRSTOR/ NUP, TATR(1400), NUMAX, IACT(70),
+ IDC(22), IFCRNG(34), KAMAX, MXTR
COMMON /CONPK/ NCELL, NMX
DATA DPR/57.296/, A/0.1/

C
C IEM IS NO. OF EVENTS IN C RADIAL.
C INITIALIZE AND GENERATE HC ARRAY

NBADR=NCADR
NCADR=IAMAX+NAC
NBKA=NCKA
NCKA=KAMAX+NAC
NAX=NA
ITY=0
IF(NA. NE. 1) GO TO 2109
LM=6
LMDF=LM*NIDP
NBADR=IAMAX+NAC
NBKA=KAMAX+NAC
NAN=0
NANI=0
LMM=LM-1
IDX=LM+1
NIMN=1
FQUANT=RQUANT
NCLM=NCL-1
LDBM=LDB-1
NUMP=2+LM*LDB
LDX=NUMP-1
NPDP=LDX*NIDP
ID2=1+(LDB-1)*LM
DO 17 1=1, KOFST
17 UP(I)=0.
DO 18 I=1,LIMT
   18 ECL(I)=0.
   DO 19 I=1,NCL
   19 IDVEL(I)=0

   NMx=1
   DO 2107 I=1,NIDP
       IACT(I)=0
   2107 DO 2108 J=1,MXTR
       TATR(J)=0.
   2108 K=1,KMAX
   DO 23 IPCNT(KAD)=0
   23 DO 2±07 II,NIDP
       IRCT(I)=S
       DO 2±08 J=MXTR
           TATR(J)=0.
   2±09 DO 23 K=1,KMAX
       IPCNT(KAD)+=0
   23 IF(<IEM.LE.0) GO TO 952
   DO 951 IE=1,IEM
       IER=1+NEMC
       KIE=(IE-1)*KMAX+NCKA
       KIEM=IE-KMAX
       IPTC(IER)=0
       ICESS=IC1(IER)
       ICESP=IC2(IER)
       IF(IE.EQ.0) GO TO 232
       DO 233 K=1,KMAX
           KA=K+KIE
           KB=K+KIEM
           DO 233 IPCNT(KA)=IPCNT(KB)
   232 IPL=0
       IF(IE.EQ.0) GO TO 951
       DO 233 IPL=0
           IF(IP.LE.IPL) GO TO 951
       IPL=IPL+1
       JE1=JE2=0

   FIND EVENTS ASSOCIATED WITH C EVENTS.
   IE IS NO. OF EVENTS IN PREVIOUS RADIAL.
   IF(JEM.EQ.0) GO TO 41
   DO 41 JE=1,JEM
       JEB=JE+NEMB
       IF(IC2(JEB).LT.ICES) GO TO 31
       IF(IC1(JEB).GT.ICESP) GO TO 41
       JE2=JE
       IF(JE1.LE.0) JE1=JE
   41 CONTINUE
   31 CONTINUE

   FIND THRESHOLDS FOR IE EVENT

   DO 51 J=1,WMXDB
       T(J)=0
       NTHRES=1
       DO 71 L=IPL,IP
           IF(L.GT.JMAX) GO TO 71
           IR1=IPCRNG(L)
           IF(IR1.LT.ICES) GO TO 71
           IF(IR1.GT.ICESP) GO TO 712
           DO 711 K=1,LDB
               IU=W(IR1)
               IT=IU-TH-K+1
               IF(IT.LE.0) GO TO 711
               IF(IT.GT.JMXDB) IT=JMXDB
               IF(T(IT).EQ.0) NTHRES=NTHRES+1
       711 CONTINUE
       71 CONTINUE
IF(NTHRES. GT. KMAX)WRITE(6, 7878)NA, NTHRES, KMAX, IE
7878 FORMAT(2X, 13HNUMBER OF THRESHOLDS EXCEEDS KMAX, 4I10)
IF(NTHRES. GT. KMAX)IPSRT=NTHRES-KMAX
IPT=1
DO 91 L=1, JMXDB
IF (T(L). LE. 0) GO TO 91
KA=IPT+KIE
TC(KA)=L+TM-1
IPSRT=IPSRT-1
IF(IPSRT. GT. 0)GO TO 91
IPT=IPT+1
91 CONTINUE
IF(IPT=0)GO TO 951
IF(IPT. GT. JR)WRITE(6, 7879)NA, IPT, JR, IE
7879 FORMAT(2X, 13HNUMBER OF THRESHOLDS EXCEEDS JR, 4I10)
IF(IPT. GE. JR)IPT=0
IPSRT=IPSRT-1
IF(IPSRT. LT. 0)GO TO 91
CONTINUE
IPT=IPT-1
IF(IPT. LT. JR)WRITE(6, 7879)NA, IPT, JR, IE
CONTINUE
LOOP ON RANGE IN IE EVENT TO FIND CONTOUR

IBGN=ICEST+1
IND=ICESP+1
DO 161 I=IBGN, IND

LOOP ON THRESHOLD
DO 131 K=1, IPT
IUt=K+KIE
IF (IU.GT.TC(KA)) GO TO 111
GO TO 141
111 IU=IU-1
IF (IU.LE.TC(KA)) GO TO 121
GO TO 131

START RANGE FOR SEGMENT (CONTOUR)

121 IPCNT(KA)=IPCNT(KA)+1
IF(IPCNT(KA). LE. IMXJMX)GO TO 1211
WRITE(6, 1212)ITY, K, IE
1212 FORMAT(2X, 13HNUMBER OF SEGMENTS EXCEEDS IMXJMX, 4I10)
IPCNT(KA)=IMXJMX

1211 IPE=IPCNT(KA)
IREG=1-1
IADDR=IPE+(K-1)*IMXJMX+NCDR
IPC(IADDR)=IREG
IPC2(IADDR)=0
131 CONTINUE
GO TO 161

END RANGE FOR SEGMENT

141 DO 151 KL=K, IPT
IF(W(I-1). EQ. -999) GO TO 161
IUt=K+KIE
IF (IU.LE.TC(KA)) GO TO 161
IPE=IPCNT(KA)
IREG=1-1
IADDR=IPE+(KL-1)*IMXJMX+NCDR
IPC2(IADDR)=IREG
151 CONTINUE
161 CONTINUE

ASSOCIATE CELLS  LOOP ON THRESHOLD HIGHEST TO LOWEST
DO 941 LC=1, IPT
KL=IPT- LC+1
KR=KC+ KIE
KZ=KC+ KIEM
IF(KC LE. 0) GO TO 941
TCVL=TC(KA)
TCVM=TCVL+1
TCVLB=TCVL+LDB
NPC= IPCNT(KA)
NPL=0
IF(KC GT. 1) NPL= IPCNT(KZ)
IF(NPC LE. NPL) GO TO 941
NPL= NPL+1
C
LOOP ON SEGMENTS
DO 931 IPE=NPL, NPC
JADDR=IPE+(KC-1)*IMXJMX+NCADR
IHBM=IPC1(JADDR)
IHD=IPC2(JADDR)
K=KC+1
KY=KA+1
KX=KZ+1
NPK=0
TATM=0
LPE=IPCNT(KY)
LPL=0
IF(IE GT. 1) LPL= IPCNT(KX)
LPL=LPL+1
IF(LPE LT. LPL OR K GT. IPT) GO TO 193
C
C
LOOP SEGMENTS, NEXT HIGHER THRESHOLD
C
192 DO 191 L=LPL, LPE
IADDR=L+(K-1)*IMXJMX+NCADR
IF(IPC2(JADDR), LT. IHBM) GO TO 191
IF(IPC1(JADDR), GT. IHD) GO TO 193
NPCEL=IPC3(IADDR)
IF(NPCEL LE. 0) GO TO 191
TATC=TATR(NPCEL)
IF(TATC LT. TATM) GO TO 231
TATM=TATC
NPK=NPCCEL
231 IF(TATC LT. 0) TATC=-TATC
C
NPCEL IS FOR NEXT HIGHER (ENCLOSED) THRESHOLD ON C RADIAL
191 CONTINUE
GO TO 193
932 NPK=-NPCEL
GO TO 193
1911 NPK=-(NIDF+1)
C
ASSOCIATE CELLS ON B RADIAL, TOP DOWN
C
193 MPK=0
IF(NA EQ. 1) GO TO 361
TATM=0
IF(JE2 EQ. 0) GO TO 371
DO 261 JE=JE1, JE2
JEB=JE+ NEMB
ITATM=0
IF(IC2(JEB), LT. IHBM) GO TO 261
IF(IC1(JEB), GT. IHM) GO TO 3661
C
271  IFB=IPTC(JEB)
    IF(IBF LE 0)GO TO 261
    DO 291 LB=1,IPB
    KB=IPB-LB+1
    KBB=(J-E-1)+KMAX+NBKA
    KBA=KB+KBB
    KBC=KB+KBB-KMAX
    TBVL=TC(KBA)
    JEOE=TBVL+1
    NP2=IPCNT(KBA)
    NP1=0
    IF(JE GT 1)NP1=IPCNT(KBC)
    IF(NP2 LE NP1)GO TO 291
    NP1=NP1+1
    DO 281 JPE=NP1,NP2
    IADDR=JPE+(KB-1)+IMXIMX+NBAE
    IF(IPC2(IADDR) LT IHEM)GO TO 261
    IF(IPC1(IADDR) GT IHM)GO TO 2911
    LPCEL=IPC3(IADDR)
    IF(LPCEL LE 0)GO TO 281
    IF(TCVL LE TBVL)GO TO 282
    IEQL=TATR(LPCEL)
    IF(JEQL LT IEQL)GO TO 261
    282 TATC=TATR(LPCEL)
    IF(TATC LT TATM)GO TO 281
    TATM=TATC
    ITATM=TATM
    MPK=LPCEL
    KBM=KB
    JBM=JE
    281 CONTINUE
    2911 IF(TATM NE 0)GO TO 261
    291 CONTINUE
    261 CONTINUE
    3661 IF(MPK EQ 0)GO TO 371
    IF(TATM GT TCVLB)MPK=-MPK
    GO TO 421
    371 DO 194 I=IHB,1HD
    IF(HB(I) EQ -999)GO TO 194
    IF(IABS(HB(I)) LE TC(KA))GO TO 194
    IF(NPK EQ 0)GO TO 931
    IF(NPK GT 0)GO TO 366
    GO TO 3662
    194 CONTINUE
    HAVE B COMPARE WITHIN RANGE
    361 CONTINUE
    IF(NPK EQ 0)GO TO 631
    MPK=0 AND NPK=0 - NO COMPARE
    MPK=0 AND NPK NE 0 - NO B COMPARE
    NPK=0 AND MPK NE 0 - B COMPARE
    HIGHEST THIS RADIAL
    IF(NPK LE 0 OR NPK GT NMX)GO TO 3662
    NO PRIOR RADIAL FOR COMPARISON, INCREMENT NPCEL
    NPCEL=NPK
    359 INDX=TATR(NPCEL)-TCYM
    IF(INDX GE LDS OR INDX LE 0)GO TO 366
    392 IN=NINDX+1
    INX=INDX+INDX+LM
IEQL-TATR(JN)
IF(IEQL NE 0 OR NA EQ 1) GO TO 3921
IEQL=NPCEL+(IN-LMM-1)+NIDP
IEQL=TATR(IEQL)
IF(IEQL LE 0) GO TO 366
NPC=NPCEL
NPCEL=IEQL
IF(NPC EQ NPCEL OR NPCEL GT NMX) GO TO 366
GO TO 359
3921 IPC3=JADDR=NPCEL
JN1=JN+NIDP
JN2=JN1+NIDP
JN3=JN2+NIDP
JN4=JN3+NIDP
JN5=JN4+NIDP
IF(TATR(JN1) EQ -999) GO TO 419
1ST=IHB
ISP=IHD
DO 411 1=1ST,ISP
R=RI+SA+(FLOAT(I-1)+5)
I=I+1
RU=R+FLOAT(IU)+DAZ
TATR(JN1)=TATR(JN1)+DAZ*R
TATR(JN2)=TATR(JN2)+RU
TATR(JN3)=TATR(JN3)+SAZ*R+RU
TATR(JN4)=TATR(JN4)+CAZ*R+RU
TATR(JN5)=TATR(JN5)+IDVEL(I)
411 CONTINUE
419 KN=NPCEL+(INX-1)+NIDP
TATR(KN)=NA
KN=NPCEL+NPD
TATR(KN)=IE
IF(1ST EQ 2 OR ISP EQ IMX) TATR(JN1)=-999
GO TO 366
3662 NPCEL=NPC
366 IF(NPCEL GT NMX OR NPCEL LF 0) GO TO 931
INDEX=TATR(NPCEL)-TCVM
C
C COMBINE NPCEL WITH NPCEL AT THIS LEVEL
C
C COMBINE BY SETTING AREA AS POINTER AND IDX TO NA = 0
C
C
C IF(NPC LF LPL OR A GT IPT) GO TO 931
DO 365 L=LPL,LPE
IADDR=L+(K-1)+INM,IMX
IF(IPC2<IADDR) LT IMB M GO TO 365
IF(IPC1<IADDR) LT IMH M GO TO 931
LPCE = IPC (IADDR)
IF(LPCE LE 0 OR LPCE GT NMX) GO TO 365
LNX=LPCE+LMDP
IF(TATR(LNX) EQ 0) GO TO 365
IF(NPCEL EQ LPCE+LMDP) GO TO 365
INDEX=TATR(LPCE)-TCVM
IF(INDEX GE LDB) GO TO 365
IF(INDEX LE 0) INDEX=0
IND=LPCE+(INDEX+1)*LMDP
IF(TATR(IND) EQ 0) GO TO 365
INDEX=INDEX+1
IPG=0
DO 3663 J=INDEX,LDN
INDEX=INDEX+1
IEQL=LPCE+J*LMDP
IEQL=TATR(IEQL)
IF(IEQL EQ NA) IPG=IPG+1
DO 3663 I=INDEX
IEQL=LPCE+(1+INDEX)*NIDP

IF(IPD EQ 0) OR IE LE 1 GO TO 3664
DO 3665 I=L,IE
IA=I+NEMC
IPTT=IPTC(I)
IF(IPTT LE 0') GO TO 3665
DO 3666 KT=1, IPTT
KTI=(T-1)*KMAX+NCA
KTH=KT+KTI
KTB=KT+KTI-KMAX
NPCT=IPCNT(KTA)
IEOL=TC(KTA)+1
NPCL=0
IF(I GT 1) NPCL*IPCNT(KTB)
IF(NPCT LE NPCL) GO TO 3666
NPCL=NPCL+1
DO 3667 LP=NPCL, NPCT
IADDR=LP*(AT-1)+IMX+INADDR
IF(LPCEL NE IPC3)IADDR) GO TO 3667
INDNT=TATR(NPCEL)-IEOL
IF(INDNT LT LDR'GO TO 3668
3669 IPC3(IADDR)=0
GO TO 3667
3668 IF(INDX GE LDB'GO TO 3669
3669 CONTINUE
3666 CONTINUE
3665 CONTINUE
TIP=0
3664 IF(INDX GE LDB'GO TO 365
IACT(LPCEL)=NPCEL
IEOL=LPCEL+1*INDX*LM+1*NIDP
TATR=IEOL-NPCEL
IF(INDX NE 0') GO TO 365
IACT(LPCEL)=NIDP-1
LN=LPCEL*NIDP
TATR=LN=0
365 CONTINUE
GO TO 931
C C C C C
COMBINE NPCEL AND LPCEL; PEAK VALUES EQUAL
C C C C C
COMBINE WITH F RAPID CELLS
C
421 IF(NPK LE 0') GO TO 422
IF(NPK LT 0' GO TO 3662
NGM=0
LPCEL=NPK
LMX=LPCEL+LMDF
IEOL=TATR(LMX)
KEM=KEM+1+JEM-1)*KMAX+NBA
IF(IEOL EQ NA AND NPK EQ 0 AND TCYL EQ 0+
+ TC(KEM) GO TO 485
INDEX=TATR(LPCEL)-TCVM
IMOX=INDEX
IF(NPK GT 0') IMOX=TATR(NPK) TCVM
IF(INDX NE INDEX) GO TO 4212
NGM=1
NPCEL=NPK
IND=INDEX
INDEX=IND
INDEX=IND
GO TO 4213
4212 IF(INDX LT 0') GO TO 481
NPCEL=LPCEL
COMBINE WITH B-RADIAL, C-LEVEL LOWER

4213 IF (INDEX .GE. LD8) GO TO 4221
IN=INDEX*LM
IN1=INDEX*LM*NIDP
INL=(INDEX+LM)*NIDP
IEQ=NPCEL+ILN
512 IF (TATR(IEQL).NE.0.) GO TO 5311
IEQ=NPCEL+IN1
JEQL=TATR(IEQL)
IF (JEQL .LE. 0. AND. NGM .EQ. 0) GO TO 4221
IF (NGM .NE. 1) GO TO 5312
5314 IM=INDEX*LM
IF (IM .LT. 0) GO TO 5311
IF (LPCEL .LE. 0. OR. LPCEL .GT. NMX) GO TO 422
IEQL=LPCEL+(INDEX+LM)*NIDP
5313 IF (TATR(IEQL).NE.0.) GO TO 5311
IEQL=LPCEL+INDEX*LM*NIDP
IF (TATR(IEQL).GT.0.) GO TO 5313
LPCEL=NPCEL
GO TO 4221
5312 IEQL=NPCEL+IN1
NPCEL=TATR(IEQL)
IF (NPCEL .LE. 0. OR. NPCEL .GT. NMX) GO TO 4221
INDX=TATR(NPCEL)-TCVM
GO TO 5314
5311 IPC3(JADDR)=NPCEL
IEQL=NPCEL+IN1
IF (TATR(IEQL).EQ.-999.) GO TO 8012
IST=1HB
ISP=1HD
IEQL2=IEQL+NIDP
IEQL3=IEQL2+NIDP
IEQL4=IEQL3+NIDP
IEQL5=IEQL4+NIDP
DO 531 I=IST, ISP
IU=W(I)
R=RI+SA*(FLOAT(I-1)-.5)
RU=R+FLOAT(IU)*DAZ
TATR(IEQL)=TATR(IEQL)+DAZ*R
TATR(IEQL2)=TATR(IEQL2)+RU
TATR(IEQL3)=TATR(IEQL3)+SAZ*R+RU
TATR(IEQL4)=TATR(IEQL4)+CAZ*R+RU
TATR(IEQL5)=TATR(IEQL5)+IDVEL(I
531 CONTINUE
8012 IEQL=NPCEL+ILN
TATR(IEQL)=NA
NMP=NPCEL+NPD
TATR(NMP)=IE
IEQL=NPCEL+IN1
IF (IST .EQ. 2 .OR. ISP .EQ. NMX) TATR(IEQL)=-999.
LPCEL=NPCEL
GO TO 4221

COMBINE WITH B-RADIAL, C-LEVEL HIGHER

IF FIRST COMBINE, AREA=0, IF SECOND OR HIGHER, AREA=-1.
TEST AREA TO ESTABLISH NEW NUMBERS

481 INDEX=-INDEX
IND=NUMP-1
INS=2
IPG=0
TATR(LPCEL)=TCVM
LMP=LPCEL*NIDP
TATR(LMP)=1E
IF(INDX GE LDB)GO TO 482
IND=LDB-INDEX
DO 4632 I=INDEX,LDB
IEQL=LPCEL+(I+1)*LMDP
JEQL=TATR(IEQL)
IF(JEQL EQ NA)IPG=IPG+1
4832 CONTINUE
DO 483 I=1,IND
DO 483 J=I,IND
IN=LPCEL*(I+(LNC-1)*LMDP
IM=LPCEL*(J+(LNC-1)*LMDP
483 TATR(IN)-TATR(IM)
IND=INDN+LMDP
482 DO 4833 I=1,LDB
IEQL=LPCEL+I*LMDP
JEQL=TATR(IEQL)
IF(JEQL EQ NA)IPG=IPG+1
4835 CONTINUE
DO 484 I=INS,IND
IN=LPCEL+(I-1)*NIDP
484 TATR(IN)=0
DO 4841 I=1,IND
IEQL=LPCEL+I*LMDP
4841 TATR(IEQL)=NA
IF(IPG EQ 0 OR IE LE 1)GO TO 488
DO 4831 I=1,IE
IA=I+NEMC
IPTT=IPTC(IA)
IF(IPTT LE 0)GO TO 4831
DO 4833 KT=1,IPTT
KTP=(I-1)*KMAX+KCA
KTA=KT+KTP
KTB=KT+KTP-KMAX
NPCT=IPCNT(KTA)
IEQL=TC(KTA)+1
NPCL=0
IF(I GT 1)NPCL=IPCNT(KTB)
IF(NPCT LE NPCL)GO TO 4833
NPCL=NPCL+1
DO 4834 LP=NPCL,NPCT
KADDR=LP+(KT-1)*INX+KMAX+KCA
IF(LPCEL NE IPC3(KADDR))GO TO 4834
INDXT=TATR(LPCEL)-IEQL
IF(INDXT LT LDB)GO TO 4834
IPC3(KADDR)=0
4834 CONTINUE
4833 CONTINUE
4831 CONTINUE
IPG=0
488 IN=0
IF(LPCEL LE 0 OR LPCEL GT NMX)GO TO 931
LND=LPCEL+LMDP
TATR(LND)=NA
IPC3(KADDR)=LPCEL
NPCEL=LPCEL
NGM=0
GO TO 512
485 DO 486 I=NMIN,NIDP
IF(IACT(I) EQ 0)GO TO 487
IIL, IN Ii
TATRN=TATR(JEOQL)
IF(TATRL. NE. 0. . AND. TATRN. NE. 0.) GO TO 8911
JEQL=LPCEL+LDA
TATRJ=TATR(JEOQL)
IF(TATRL. EQ. 0. . AND. TATRJ. LE. 0.) GO TO 851
IF(TATRL. GT. 0.) GO TO 8912
LPCEL=TATRJ
IF(LPCEL. LE. 0. OR. LPCEL. GT. NMX) GO TO 461
GO TO 4221

8912 IEQL=NPCEL+NDA
TATRJ=TATR(IEQL)
IF(TATRN. EQ. 0. . AND. TATRJ. LE. 0.) GO TO 8913
NPCEL=TATRJ
IF(NPCEL. LE. 0. OR. NPCEL. GT. NMX. OR. NPCEL. EQ. LPCEL) GO TO 461
IPC3(IADDR)=NPCEL
GO TO 502

8913 DO 8914 I=1,LM
IEQL=LPCEL+(LD+I-1)*NIDP
8914 TATR(IEQL)=0.
IPC3(IADDR)=0
GO TO 4221

8911 IBNDRY=0
IEQL=LPCEL+LDA
JEQL=NPCEL+NDA
IF(TATR(IEQL). EQ. -999. . OR. TATR(JEQL). EQ. -999. ) X IBNDRY=1
DO 891 I=1,LM
IEQL=LPCEL+(LD+I-1)*NIDP
JEQL=NPCEL+(ND+I-1)*NIDP
IF(IBNDRY. EQ. 0) TATR(IEQL)=TATR(JEQL)+TATR(IEQL)
TATR(JEQL)=0.

891 CONTINUE
IEQL=LPCEL+LDA
IF(IBNDRY. EQ. 1) TATR(IEQL)=-999
IEQL=NPCEL+(ND+LMM)*NIDP
TATR(IEQL)=0.
IEQL=NPCEL+NDA
TATR(IEQL)=LPCEL
IAC(j NPCEL)=-LPCEL
IPC3(IADDR)=LPCEL

461 CONTINUE
471 CONTINUE
441 CONTINUE
632 IF(NPK. LE. 0) GO TO 3662
NPCEL=LPCEL
GO TO 366

C C
C
631 DO 642 J=NIMN, NIDP
IF(IAC(J). EQ. 0) GO TO 643
642 CONTINUE
WRITE(6, 644)
644 FORMAT(5X, 1SH TOO MANY CELLS)
GO TO 931
643 NPCEL=J
NIMN=J
IF(NIMN. gt. NIDP) NIMN=NIDP
IF(NMX. LT. NIMN) NMX=NIMN
IAC(J)=1
IPC3(IADDR)=NPCEL
DO 671 I=1, NUMP
NPMP=NPCEL+(I-1)*NIDP
TATR(NMP)=0. 0
671 CONTINUE
NMN = NPCEL + NPDP
TATR(NMN) = IE
IST = IHB
ISP = IHD
NP2 = NPCEL + NIDP
NP3 = NP2 + NIDP
NP4 = NP3 + NIDP
NP5 = NP4 + NIDP
NP6 = NP5 + NIDP
DO 621 I = IST, ISP
R = RI + SA * (FLOAT(I - 1) - 5)
IU = W(I)
RU = R + FLOAT(IU) * DAZ
TATR(NP2) = DAZ * R + TATR(NP2)
TATR(NP3) = RU + TATR(NP3)
TATR(NP4) = SAZ * R + RU + TATR(NP4)
TATR(NP5) = TATR(NP5) + CA2 * R + RU
TATR(NP6) = TATR(NP6) + IDVEL(I)

621 CONTINUE
NIX = NPCEL + LMDP
TATR(NIX) = NA
IF (IST .EQ. 2 .OR. ISP .EQ. IMX) TATR(NP2) = -999.
931 CONTINUE
9411 CONTINUE
9511 CONTINUE

C CLEAN UP TATR AND IC ARRAYS - REMOVE IC POINTER C TO DELETED ARRAYS C

7010 DO 9512 I = 1, NMX
IF (IACT(I) .EQ. 0) GO TO 9512
IF (IACT(I) .GE. 0) GO TO 9611
DO 9613 IE = 1, IEM
IEA = IE + NEMC
KIE = (IE - 1) * KMAX + NCKA
KEM = KIE - KMAX
IPT = IPTC(IEA)
IF (IPT .LE. 0) GO TO 9613
DO 9618 IE = IPT + KEM
CA = CA + IE
KS = CA + KEM
NPC = IPCNT(KA)
TCVM = TC(KA) + 1
NPL = 0
IF (IE .GT. 1) NPL = IPCNT(KB)
IF (NPL .LT. NPC) GO TO 9618
NPL = NPL + 1
DO 9619 IE = NPL, NPC
IADDR = IPE + (KC - 1) * IMXJMX + NCADR
IF (I .NE. IPC3(IADDR)) GO TO 9619
IF (IACT(I) .LT. -NIDP) GO TO 9614
INDEX = TATR(I) - TCVM
IEQL = IE + INDEX * LMDP
JEQL = IE + INDEX * LM + 1 + NIDP
IF (TATR(JEQL) .NE. 0) GO TO 9619
JEQL = TATR(JEQL)
IF (JEQL .NE. -IACT(I)) GO TO 9614
IPC3(IADDR) = -IACT(I)
GO TO 9619
9614 IPC3(IADDR) = 0
9619 CONTINUE
9618 CONTINUE
9613 CONTINUE
IF (IACT(I) .GE. -NIDP) GO TO 9517
IACT(I) = 0
TATR(I)=0.
GO TO 9512
9517 DO 9513 J=1, LDB
KEQL=I+(LM*(J-1)+1)*NIDP
JEQL=TATR(KEQL)
IEQL=I+J*LMDP
IEQL=TATR(IEQL)
IF(JEQL .EQ. -I(CT(I).AND. IEQL .EQ 0).
+ GO TO 9514
9513 CONTINUE
GO TO 9611
9514 TATR(KEQL)=0
9611 DO 9612 K=2, LDB
IEQL=I+K*LMDP
JEQL=I+(K-1)*LM+1)*NIDP
IF(TATR(IEQL) .NE. 0 .AND. TATR(IEQL) .EQ. 0.)
+ TATR(IEQL)=0.
9612 CONTINUE
I(CT(I)=1
9512 CONTINUE
IF(NA .EQ. 1) GO TO 1030
C
END OF ASSOCIATION LOOPS
C
952 DO 991 I=1,NMX
IA=I+(LDX-1)*NIDP
IF(I(CT(I) .EQ. 0) GO TO 991
961 IF(TATR(IA) .EQ. 0) GO TO 9912
IEQL=TATR(IA)
IF(IEQL .LE. 0)IEQL=-IEQL
IF(IEQL .EQ. NAX-1) GO TO 971
952 CONTINUE
C
CHECK BACKGROUND COMING DOWN
C
971 INBR=0
ITERM=1
DO 9716 J=1, LDBM
IEQL=I+(J-1)*LM+1)*NIDP
JEQL=IEQL+NIDP
IF(TATR(IEQL) .LE. 0 .OR. TATR(IEQL) .EQ. 0) GO TO 9982
9716 CONTINUE
NMP=I+NPDP
IEQL=TATR(NMP)
J=1
IF(J .LT. 2) GO TO 968
DO 9711 J=1, JEM
968 JA=J+NEMB
IF(IEQL .NE. J) GO TO 9711
9712 IPB=IFTC(JA)
DO 9713 K=1, IPB
KA=(J-1)+KMAX+NKKA
KAP=K+KA
KAM=K+KA-KMAX
IEQL=TATR(I)
IEQL=IEQL-TC(KAP)
IF(IEQL .NE. LDB) GO TO 9713
NP=IPCNT(KAP)
NL=0
IF(J .GT. 1)NL=IPCNT(KAM)
NL=NL+1
DO 9713 N=NL, NP
IEQL=N+(K-1)*MXJMX+NBADR
IF(I(NE IPCNT(IEQL)) GO TO 9713
INBR=INBR+1
ITERM=0
IST=IPC1(IADDR)
ISP=IPC2(IADDR)+1
DO 9715 L=IST,ISP
IU=W(L)
IF(IU.EQ.-999)GO TO 9715
IF(ABS(IU).GT.TC(KAP))GO TO 9962
9715 CONTINUE
9711 CONTINUE
ITERM=3
IF(INBR.EQ.0)GO TO 9962
ITERM=4
IID=I+ID2*NIDP
IF(TATR(IID).LE.TATRMN)GO TO 9962
DO 981 J=1,LMM
IEGL=I+(ID2+J-1)*NIDP
981 UP(J)=TATR(IJ)
UP(2)=UP(2)/UP(1)
UP(6)=A*UP(2)+UP(5)
DO 985 M=1,NOFST
MG=6+(M-1)*KOFST+NAN1
IF(UP(M).GT.ECL(M))GO TO 9966
985 CONTINUE
GO TO 9982
986 NAM=NAN+1
IF(NAN.GT.1)NAN=0
NAB=NAN1
NAN1=NAN+ICLAD
LMT=NOFST-1
DO 988 J=1,LMT
KJ=(J-1)*KOFST
DO 987 K=1,LMM
JK=K+KJ
JK1=JK+NAB
JK2=JK+NAN1
IF(J.EQ.M)ECL(JK2)=UP(K)
IF(J.GE.M)JK2=JK2+KOFST
987 ECL(JK2)=ECL(JK1)
988 CONTINUE
ITERM=5
GO TO 9982
9912 DO 9913 J=1,LDB
INDP1=I+J*LMDP
IEGL=TATR(INDP1)
IF(IEGL.LT.0)IEGL=-IEGL
IF(IEQL.EQ.NA)GO TO 991
IF(TATR(INDP1).LT.0. AND. TATR(INDP1).NE.-999.)GO TO 991
9913 CONTINUE
ITERM=7
9982 CONTINUE
IF(I.LT.NIMN)NIMN=I
DO 982 J=1,NUMP
JA=I+(J-1)*NIDP
982 TATR(JA)=0.
IACT(J)=0
991 CONTINUE
1030 CONTINUE
1031 CONTINUE
C037 FORMAT(3I15)
1840 DO 1 J=1,1EM
JR=I+NEMC
ISTART=IC1(JR)
ISTOP=IC2(JR)
D-30
DO 2 I=ISTART, ISTOP
IF (MH=MH(I)) RETURN
HE(C(I)+1) = 0
HE(C(I)+1) = MH
CONTINUE
END
RETURN

SUBROUTINE TRACK
IMPLICIT INTEGER*2 (I-N), INTEGER*2 T, TA
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /NLIS/ NVARM, NCARM, NVO, ICO, IO, JO, JVR, KTL
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /PNTR/ NCMM, NVMIN, NVMX, IELS, NLSCAN, IESEN, NVSCN, NT
COMMON /DATA/ ECL(224), NOFST, KOFST, ICLAD, NANI
COMMON /ECONE/ EARTH, VMK
COMMON /CNT/ COSPHI, SINPHI, COSPH2
COMMON /A00/ SINA, COSA, DELTA, ISCANF, NEL, R1, SA
COMMON /KTA/ NV, NC, UCN

C C

UCN=0
VKM=COSPHI*VMK
VKM2=VKM*VKM
SAVM2=SA+VKM2
SAVM=SA+VKM
NAN2=NANI+1
DO 10 M=1,NOFST
MA=(NOFST-M)*KOFST+NAN2
IF(ECL(MA)>0) GO TO 22
10 CONTINUE
GO TO 41
22 NCMX=NOFST-M+1
DO 30 M=1,NCMX
M1=1+(M-1)*KOFST+NANI
M2=1+M1
M3=1+M2
M4=1+M3
M5=1+M4
M6=1+M5
M7=1+M6

C WRITE(6,50)M1, M2, M3, M4, M5, M6, COSPHI, VKM, ECL(M1), ECL(M2),
C + ECL(M3), ECL(M4), ECL(M5), ECL(M6), NANI
C 50 FORMAT(1X, 616, 4X, 3HCOS, F10.5, 4X, 3HVKM, F10.2, 4HECL-,
C + 6FB 2.4X, 4HNANI, 16)
ECL(M5)=ECL(M4)*SAVM
VKM=VKM/(ECL(M2)+ECL(M1))
ECL(M3)=ECL(M3)+VKME
ECL(M4)=ECL(M4)+VKME
ECL(M5)=ECL(M5)+SAVM2
R2=ECL(M3)*ECL(M3)+ECL(M4)*ECL(M4)
ECL(M7)=ECL(M6)
30 ECL(M6)=(SQRT(R2))*SINPHI*R2*COSPH2
FNSN=FNSN+1
IF(NSCAN.NE.1) GO TO 41
KTL=T
JDAY=JDAY
JHR=IHR
JMIN=MIN
JSEC=ISEC
DO 40 NC=1,NCMX
NC=NC
40 CALL ATRAK
40 CALL BTRAK
NVNX = NCMX
NVMIN = NVMX
GO TO 45

41 CONTINUE
   IF (IELSN .LT. IESNL) CALL STRAK
   IESNL = IELSN
   CALL COMPAR
45 CONTINUE
   RETURN
END
SUBROUTINE ATRAK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TA
LOGICAL PR1, PR2, PRIN2
INTEGER*4 IVCL(736)
COMMON /ZLOOK/ IZOFF, ZARY(91)
COMMON /TLIS/ T, TA, TDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JO, JR, KTL
COMMON /PNTRS/ NCMX, NVMIN, NVMX, IELS, NSCAN, IESNL, NVSCN, NTT
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2
COMMON /DVAL/ DELA
COMMON /DATA2/ VCL(736), MXVC, NVVC
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NAN1
COMMON /KTA/ NV, NC, DELW
EQUIVALENCE (VCL(1), IVCL(1))
IF(NC .LE. 0. OR. NV .GT. NCMX) GO TO 100
IF(NV .LE. 0. OR. NV .GT. NVARM) GO TO 100
NCEC=(NC-1)*KOF'+T+NLN
NVVC=(NV -1)*NVVC
NCA=2+NCEC
IZ=ECL(NCA)
NCA=NCA+1
X=ECL(NCA)
NCA=NCA+1
Y=ECL(NCA)
NCA=6+NCEC
H=ECL(NCA)
NVA=9+NVVC
IF(IVCL(NVA) .GE. 0) GO TO 10
DEFINE LOWEST ELEVATION VALUES
NVA=1+NVVC
VCL(NVA)=:;
NVA=NVA+1
VCL(NVA)=Y
NVA=NVA+1
IVCL(NVA)=IZ
NVA=NVA+1
NCA=NCEC+1
VCL(NVA)=ECL(NCA)
NVA=NVA+1
IVCL(NVA)=T
NVA=NVA+1
VCL(NVA)=H
NCA=3+NCEC
NCA1=4+NCEC
R2=ECL(NCA)*ECL(NCA)+ECL(NCA1)*ECL(NCA1)
R=SQRT(R2)
NVA=NVA+1
VCL(NVA)=R
NVA=NVA+1
IF(IVCL(NVA) .NE. 0) GO TO 10
NTT=NTT+1
IVCL(NVA)=NTT
INCREMENT ATTRIBUTE ARRAYS
10 IZL=IZL+1
   IZL=IZL+1
   IF(IZL.LT.1) IZL=1
   IF(IZL.GT.91) IZL=91
   Z=ZARY(IZL)
17 NVA=9+NVVC
   IVCL(NVA)=IVCL(NVA)+1
   NCA=NCEC+1
   NVA=23+NVVC
   VCL(NVA)=VCL(NVA)+ECL(NCA)
   NVA=10+NVVC
   VCL(NVA)=VCL(NVA)+Z
   NVA=NVA+1
   VCL(NVA)=VCL(NVA)+Z*X
   NVA=NVA+1
   VCL(NVA)=VCL(NVA)+Z*Y
   NVA=14+NVVC
   HL=VCL(NVA)
   IF(HL.LT.0 OR FNSN.LT.2) GO TO 11
   HL=H-DELA*R
11 NCA=NCEC+1
   NVA=13+NVVC
   VCL(NVA)=VCL(NVA)+Z*(H-HL)*ECL(NCA)
   SUMMIT VALUES
   NVA=NVA+1
   VCL(NVA)=H
   NVA=NVA+1
   IVCL(NVA)=IZ
   PEAK, BASE, AND TOP ARRAYS
   NVA=NVA+1
   IZP=IVCL(NVA)
   IF(IZP.GT.12) GO TO 99
   SET PEAK
   32 NVA=16+NVVC
       IVCL(NVA)=IZ
       NVA=NVA+1
       VCL(NVA)=H
   99 IF(.NOT.PRIN2) GO TO 100
       NCA=NCEC+1
       NVA=9+NVVC
100 RETURN
   END
SUBROUTINE BTRRK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TA
INTEGER *4 IVR(192)
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NANI
COMMON /DATA2/ VR(192), MXVR, NVR1
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JVR, KTL
COMMON /KTA/ NV, NC, UCN
EQUIVALENCE(VR(1), IVR(1))
IF(NV. LE. 0. OR. NV. GT. NVARM) GO TO 10
IF(NC. LE. 0. OR. NC. GT. NCARM) GO TO 10
C DEFINE LAST ELEVATION VALUES
C NCEC=(NC-1)*KOFST+NANI
NVVR=(NV-1)*NVR1
NCA=3+NCEC
NVA=1+NVVR
VR(NVA)=ECL(NCA)
NCA=NCA+1
NVA=NVA+1
VR(NVA)=ECL(NCA)
NCA=2+NCEC
NVA=NVA+1
VR(NVA)=ECL(NCA)
NCA=NCA-1
NVA=NVA+1
VR(NVA)=ECL(NCA)
NVA=NVA+1
IVR(NVA)=T
NVA=NVA+1
NCA=6+NCEC
VR(NVA)=ECL(NCA)
10 RETURN
END
* 07/19/79 13:02:10  
***LISTING FOR ERT1:COMPAR.FTN
$N

SUBROUTINE COMPAR
IMPLICIT INTEGER*2(I-N)
INTEGER*2 T, TA
LOGICAL PR1, PR2, PRIN2, ARAY
INTEGER*4 IVCL(736), IVR(192)
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NANI
COMMON /DATA2/ VCL(736), MXVC, NVCI
COMMON /DATA3/ VR(192), MXVR, NVR1
COMMON /CDRAYS/ IC(32,10), C(32,9), ID(32,10), D(32,9), IM, JM
COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2
COMMON /PARMS/ VX, VY
COMMON /PNTRS/ NCMX, NVMIN, NVMX, IELSIN, NSCAN, IESNL, NVSCN, NT
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, AIDIV, A1, A2, A3, B1, B2, HDIV
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLS/ NVARM, NARM, NVO, ICO, IO, JVR, KTL
COMMON /INTL/ COSPHI, SNEL, COSPI2
COMMON /KSR/ NV, NC, UCN
COMMON /RSLV/ IUV1(32), IUV2(32), IUC1(16), IUC2(16),
+ UV(32), UC(16), NCR
EQUIVALENCE(VCL(1), IVCL(1)), (VR(1), IVR(1))
IDV=1
ICV=1
IF(NCMX LE 0) RETURN
NVX=1
IF(NVMX GT 1) NVMX=NVMX
DO 3 I=1, NVMX
UV(I)=0.
IUV1(I)=0
3 IUV2(I)=0
DO 4 I=1, NCMX
UC(I)=0.
IUC1(I)=0
4 IUC2(I)=0
DO 5 I=1, IM
IC(I,1)=0
ID(I,1)=0
DO 7 J=1, JM
J1=J+1
IC(I, J1)=0
ID(I, J1)=0
C(I, J)=0.
D(I, J)=0.
7 CONTINUE
5 CONTINUE
DO 10 NC=1, NCMX
NCEC=(NC-1)+KOFST+NAMI
NC1=1+NCEC
NC2=NC1+1
NC3=NC2+1
NC4=NC3+1
NC5=NC4+2
NVC=0
DO 48 NV=1, NVMXP
NVVC=(NV-1)+NVCI
NVR=M(NV-1)+NVR1
MLAST=0
DELNL=0.
NLR=20+NVVC
+D-37
IF(IVCL(NLR) .LE. 0 .AND. IVCL(NLR1) .LE. 0) GO TO 40

DTR = T - IVR(NLR2)

RTEST = (VMAG + DTTA) + (VMAG + DTTA) + VMISWM

NRA = 1 + NVVR
NCA = 2 + NVVC
DELX = ECL(NC3) - VR(NRA) - VCL(NCA) * DTTA
DELX2 = DELX * DELX

IF(DELX2 .GT. RTEST) GO TO 20

NRA = 2 + NVVF
NCA = 2 + NVVC
DELV = ECL(NC4) - VR(NRA) - VCL(NCA) * DTTA
DELV2 = DELV * DELV

IF(DELV2 .GT. RTEST) GO TO 20

C ASSOCIATED, FIND BEST

NRA = 3 + NVVR
NRA1 = NRA + 1
NRA2 = NRA1 + 2

DELW = ABS(ECL(NC2) - VR(NRA)) * 2 * DIV + (DELX2 + DELV2) * 1

DIV = 1 + ABS(ECL(NC1) - VR(NRA1)) * ADIV

IF(DELW .GT. VMISW) MLAST = 1
GO TO 20

30 NV = NV + 1

IF(NV .GT. 1) GO TO 31

IUC1(NC) = NV
UC(NC) = DELW
GO TO 33

31 IF(IUC1(NC) .EQ. NV) GO TO 33

NVT = NV

IF(DELW .LT. UC(NC)) GO TO 32

35 IF(IUC2(NC) .LE. 0) GO TO 36

I = IUC2(NC)

IF(I .GT. IM) GO TO 361

39 J = ID(I, I) + 1
ID(I, I) = J

IF(J .LE. JM) GO TO 37

C PRINT 101, JM, NSCAN, NVT, NC, J

JO = JO + 1
J = JM

37 ID(I, J) = NVT
D(I, J) = DELW
GO TO 33

36 I = IDV
IDV = I + 1

IF(I .LE. IM) GO TO 38

C PRINT 102, IM, NSCAN, NVT, NC, I

361 IQ = IQ + 1
I = IM

38 IUC2(NC) = I
GO TO 39

32 DX = UC(NC)
UL(NC) = DELW
DELW = DX

NVT = IUC1(NC)
IUC1(NC) = NV
GO TO 35

33 IF(IUV1(NV) .NE. 0) GO TO 21

IUV1(NV) = NC
UV(NV) = DELW
GO TO 40

C CLUSTER
* 21 IF(IU1(NV) .EQ. NC) GO TO 40
  NCT=NC
  IF(DELW .LT. UV(NV)) GO TO 22
  25 IF(IU2(NV) .EQ. 0) GO TO 26
  I=IU2(NV)
  IF(I .GT. IM) GO TO 261
  29 J=IC(1,1)+1
  IC(1,1)=J
  IF(J .LE. JM) GO TO 27
  C PRINT 101, JM, NSCN, NV, NCT, J
  C 101 FORMAT(5, NO. OF CELLS IN CLUSTER EXCEEDS JM = 5110)
  JO=JO+1
  J=JM
  27 J1=J+1
  IC(1,J1)=NCT
  C(J,J)=DELW
  GO TO 40
  26 I=ICV
  ICV=I+1
  IF(I .LE. IM) GO TO 28
  C PRINT 102, IM, NSCN, NV, NCT, J
  C 102 FORMAT(5, NO. OF ENTRIES IN CLUSTER ARRAY EXCEEDS IM = 5110)
  261 IO=IO+1
  I=IM
  28 IU2(NV)=I
  GO TO 29
  22 DX=UV(NV)
  UV(NV)=DELW
  DELW=DX
  NCT=IU1(NV)
  IU1(NV)=NC
  GO TO 25
  C NO COMPAR V, TRY VCL
  C
  20 NCA=5+NVC
  DELT=T-IVCL(NCA)
  ATEST=VMISW+<VMAG+DELT>*<VMAG+DELT>
  NCA=1+NVC
  NCA1=21+NVC
  DELX=ECL(NC1)-VCL(NCA)-VCL(NCA1)*DELT
  DELX2=DELX*DELX
  IF(DELX2 .GT. ATEST) GO TO 40
  NCA=2+NVC
  NCA1=22+NVC
  DELY=ECL(NC2)-VCL(NCA)-VCL(NCA1)*DELT
  DELY2=DELY*DELY
  IF(DELY2 .GT. ATEST) GO TO 40
  NCA=3+NVC
  NCA1=NC1+1
  NCA1=NC1+1
  NCA2=NC1+2
  DELN=ABS(<ECL(NC2)-FLOAT(IVCL(NCA))>)*ZDIV+<DELX2
  *+DELV2)*DIV,+1 + ABS<ECL(NC1)-VCL(NCA1)>+ADIV
  2 +ABS<ECL(NC0)-VCL(NCA2)>+ADIV
  IF(MLAST .NE. 0 AND DELWL .LT. DELW) DELW=DELWL
  IF(DELY .LE. VMISW) GO TO 10
  40 CONTINUE
  IF(NVC .GT. 0) GO TO 10

  C ISOLATED CELL, NO COMPAR
  C
  IF(NVMX .LT. NVMIN) GO TO 501
  DO 50 NV=NVMIN, NVMX
  NVC=(NV-1)*NVC
  50 CONTINUE
NLH=IVCL(NCA)
NCMA=9+IVNC

IF(IVCL(NCA).EQ.0 AND IVCL(NCA1).EQ.0) GO TO 55

50 CONTINUE

501 NV=NVMX+1
IF(NV.LE.NVARM) GO TO 51

C PRINT 103:NVARM,NV
C 103 FORMAT(13,NV)
C 12110)

NVO=NVO+1
NV=NVARM

51 NVMX=NV

55 NVMIN=NV
IF(NVMX.LE.0) NVMX=NV

UCN=UC(NV)
CALL ATRAK
CALL BTRAK
IUV1(NV)=-NC
IUC1(NC)=-NV
UV(NV)=0.0
UC(NC)=0.0

10 CONTINUE

C HAVE LIST OF COMPARISONS, NOW RESOLVE CONFLICTS
C

DO 60 NV=1,NVMXP
IF(IUV1(NV).LT.0) GO TO 60
IF(IUV1(NV).EQ.0) GO TO 61
NC=IUV1(NV)
IF(NC.GT.NCMX) GO TO 61
IF(IUC1(NC).LE.0) GO TO 61
IF(IUV2(NV).EQ.0 AND IUC2(NC).EQ.0) GO TO 70
NC=NC
CALL RESOLV
NC=NCR
GO TO 60

70 UCN=UC(NC)
CALL ATRAK
CALL BTRAK
IUV1(NV)=-IUV1(NV)
IUC1(NC)=-IUC1(NC)
UV(NV)=0.0
UC(NC)=0.0
GO TO 60

C NO NC COMPAR, FIX HEIGHT STATISTICS
61 IF(FNSN.LT.1.1) GO TO 60
NVCA=7+(NV-1)*NVC1
CA=VCL(NVCA)
HTC=CA*SINEL+CA*CA*COSPI2

60 CONTINUE
RETURN
END
SUBROUTINE RESOLV
LOGICAL PR1, PR2, PR3
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 IVCL(736), NVT, IVS, IYT, KV, J, IDIJ, JJ, NCT, ICS, ICT,
+KC, ICIJ
DIMENSION V(384)
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JYR, KTL
COMMON /COMB/ IV(32, 7), IVMX
COMMON /PNTR3/ NCMX, NVMIN, NVMX, IELSIN, NSCAN, IESNL, NVSCN, NT
COMMON /CDRAYS/ IC(32, 10), C(32, 9), ID(32, 10), D(32, 9), IM, JM
COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2
COMMON /DATA2/ VCL(736), MXVC, NVC1
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, A1, A2, A3, B1, B2, HDIV
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /KTRA/ NV, NC, UCN
COMMON /RSLV/ IUV1(32), IUV2(32), IUC1(16), IUC2(16),
+ UV(32), UC(16), NCA
EQUIVALENCE (VCL(1), IVCL(1))
IVMX=NVARM

C HAVE CLUSTER, ORDER LISTS

C

DO 4 I=1, 6
  4 IV(:, I)=0
IVT=4
IVS=5
ICT=1
ICS=2
JV=1
JC=1
LV=0
LC=0
KV=1
KC=1
NCT=NCA
KOF2=128
KOF3=256

IF(NCT.LT.0. OR. NCT.GT. NCMX) GO TO 100
C PROCESS NCT

65 IF(IUC1(NCT).LE.0. OR. IUC1(NCT).GT. NVMX) GO TO 66
IF(UUC(NCT).LE.0.) GO TO 66
NVT=IUC1(NCT)
IF(UUC(NCT).GT.0.) UUC(NCT)=-UUC(NCT)
CALL COMB(NVT, IVS, IYT, KV, J)
IF(IUC2(NCT).LE.0. OR. IUC2(NCT).GT. IM) GO TO 62
I=IUC2(NCT)
JX=ID(I, 1)
IF(JX.LE.0) GO TO 62
IF(JX.GT. JM) JX=JM
ID(I, 1)=-ID(I, 1)
JX1=JX+1

DO 611 J=2, JX1
  611 IDIJ=ID(I, J)
  CALL COMB(INDIJ, IVS, IYT, KV, JJ)

611 CONTINUE
C PROCESS NVT

62 IF(IUV1(NVT).LE.0. OR. IUV1(NVT).GT. NCMX) GO TO 63

C
IF(\(\text{UV}(\text{NVT}) \leq 0\)) GO TO 63  
NCT=\(\text{UVV}(\text{NVT})\)  
IF(\(\text{UV}(\text{NVT}) \leq 0\)) \(\text{UV}(\text{NVT}) = -\text{UV}(\text{NVT})\)  
CALL \text{COMBIN}(\text{NCT}, \text{ICS}, \text{ICT}, \text{KC}, \text{J})  
IF(\(\text{UVV}(\text{NVT}) \leq 0\) \text{ OR } \text{UV}(\text{NVT}) \leq 0\)) GO TO 63  
I=\(\text{UVV}(\text{NVT})\)  
JX=IC(I, 1)  
IF(JX \leq 0\) GO TO 63  
IF(JX \geq JM) JX=JM  
IC(I, 1)=-IC(I, 1)  
JX1=JX+1  
DO 621 J=2, JX1  
ICIJ=IC(I, J)  
CALL \text{COMBIN}(\text{ICIJ}, \text{ICS}, \text{ICT}, \text{KC}, \text{JJ})  
621 \text{ CONTINUE}  
  
RUN \text{COMPAR} \text{ LIST} TO \text{ FLUSH OUT FULL SET}  
  
63 DO 631 K=JV, KV  
NVT=IC(K, IVS)  
IF(\(\text{NVT} \leq 0\) \text{ OR } \text{NVT} > \text{NVT}) \text{NVT} \text{ AND } \text{UVV}(\text{NVT}) \leq 0\)) GO TO 63  
IF(\(\text{UV}(\text{NVT}) \leq 0\)) GO TO 63  
IF(\(\text{UVV}(\text{NVT}) \leq 0\) \text{ AND } \text{UV}(\text{NVT}) \leq \text{NCM}) \text{ GO TO } 64  
631 \text{ CONTINUE}  
GO TO 66  
  
64 JV=K  
LC=LC+1  
GO TO 62  
  
66 DO 661 K=JC, KC  
NCT=IC(K, ICS)  
IF(\(\text{NCT} \leq 0\) \text{ OR } \text{NCT} > \text{NCT}) \text{NCT} \text{ AND } \text{UV}(\text{NCT}) \leq 0\)) GO TO 661  
IF(\(\text{UC}(\text{NCT}) \leq 0\)) GO TO 661  
IF(\(\text{ICUC}(\text{NCT}) \leq 0\) \text{ AND } \text{ICUC}(\text{NCT}) \leq \text{NCM}) \text{ GO TO } 67  
661 \text{ CONTINUE}  
GO TO 68  
  
67 JC=K  
LV=LV+1  
GO TO 65  
  
68 IF(\(\text{LC} \leq 0\)) GO TO 69  
LC=0  
JV=1  
JC=1  
LV=0  
GO TO 63  
  
69 IF(\(\text{LV} \leq 0\)) GO TO 70  
LV=0  
JV=1  
LC=0  
JC=1  
GO TO 66  
  
HAVE ORDERED LIST, NOW FIND \text{BEST MATCH}  
  
70 IF(\(\text{KC} \leq 1\) \text{ OR } \text{KV} \leq 1\)) GO TO 100  
KV=KV-1  
IF(\(\text{KV} \geq \text{IVMX}) \text{ GO TO } 100\)  
KC=KC-1  
IF(\(\text{KC} \geq \text{IVMX}) \text{ GO TO } 100\)  
IMSM=0  
DO 701 K=1, KV  
NV=IC(K, IVS)  
IF(\(\text{NV} \leq 0\) \text{ OR } \text{NV} \geq \text{NVMX}) \text{GO TO } 701\)  
IF(\(\text{UV}(\text{NV}) \leq 0\) \text{ OR } \(\text{UV}(\text{NV}) = -\text{UV}(\text{NV})\) \text{GO TO } 701\)  
701 \text{ CONTINUE}  
DO 71 K=1, KC  
KA2=K+KOF2
VCIC)=0.
VCIC2)=0.
VCIC3)=0.
IVCK, ICT)=0
IVCK = 0
IVT)=6
IVCK , 3)=0
NC-IVCK, ICS)
IFCNV. LE. a OR. NV. GT. NVMX) GO TO 724
IFCUVCNV). GT. a)GO TO 725
IVCIC, 7)=NV
724 I=IUC2(NC)
IF(I. LE. 0. OR. I. GT. IM) GO TO 721
JX=ID(I, 1)
IF(JX. LT. 0)JX=-JX
IF(JX. LE. 0. OR. JX. GT. JM) GO TO 721
NV=0
DWT=999.
DO 723 J=1, JX
J1=J+1
NVT=ID(I, J1)
IF(NVT. LE. 0. OR. NVT. GT. NVMX) GO TO 723
IF(UV(NVT) . LE. 0. 1) GO TO 723
DELW=D(I, J)
IF(DELW. LE. 0. 1) GO TO 723
IF(DLT. LT. DWT)DWT=DELW
IF(DWT . EQ. DELW) NV=NVT
723 CONTINUE
IF(NV. LE. 0. OR. NV. GT. NVMX. OR. DWT. GT. VMISW. OR. DWT. LT. 1) 1 GO TO 721
GO TO 726
725 DWT=UC(NC)
726 KA2=K+KOFZ
V(KAR2)=DWT
IF(UV(NV) . GT. 0)UV(NV)=-UV(NV)
IF(UC(NC) . GT. 0)UC(NC)=-UC(NC)
IV(K, ICT)=NV
GO TO 72
721 KNC=KNC+1
CInv = K
72 CONTINUE
IF(KNV .LE. 0, AND. KNC .LE. 0) GO TO 75
IF(KNC .EQ. 0, OR. KNC .GT. KC) GO TO 80

C
C CASCADE REORDER OF COMPAR LIST
C
J=0
731 J=J+1
IF(J .GT. KNC) GO TO 80
K=IV(J, IVT)
IF(K .LE. 0, OR. K .GT. KC) GO TO 731
NC=IV(K, ICS)
IF(NC .LE. 0, OR. NC .GT. NCMX) GO TO 739
NV=IUC1(NC)
IF(NV .LE. 0, OR. NV .GT. NVMX) GO TO 739
DO 738 L=1, KC
IF(IV(L, 1) .EQ. NV) GO TO 7381
IF(IV(L, 1CT) .EQ. NV) GO TO 7382
738 CONTINUE
GO TO 739
7381 NCT=IV(L, ICS)
DELT = VM ISW + V(L)
GO TO 7383
7382 NCT=IV(L, ICS)
KA2=L+KOF2
DELT = VM ISW + V(KA2)
7383 KT=L
IF(NCT .LE. 0, OR. NCT .GT. NCMX) GO TO 739
IF(IUC1(NCT) .LE. 0 OR. IUC2(NCT). LE. 0) GO TO 739
DELT=UC(NC)
IF(DELT LT 0) DELT=-DELT
I=IUC2(NCT)
IF(I .LT. 0) I=-1
IF(I .LE. 0 OR. I .GT. IM) GO TO 739
JX=ID(I, 1)
IF(JX LT 0) JX=-JX
IF(JX .LE. 0 OR. JX .GT. JM) GO TO 739
DWT=999.
NVB=0
NV1=0
DO 732 L=1, JX
IF(D(I, L) .LE. 0) GO TO 732
IF(D(I, L) LT DWT) DWT=D(I, L)
L1=L+1
NV1=IABS(ID(I, L1))
IF(DWT .EQ. D(I, L)) NVB=NV1
IF(NVB .LE. 0 OR. NVT1 .GT. NVMX) GO TO 732
IF(UV(NVB) .GT. 0 AND D(I, L). LT. DWT1) DWT1=D(I, L)
IF(DWT1 .EQ. D(I, L)) NV1=NV1
732 CONTINUE
IF(NVB .LE. 0 OR. NVB .GT. NVMX OR. DWT LE. 1 OR. DWT .GT. VMISW)
1 GO TO 739
734 IF(NVB .LE. 0 OR. NVB .GT. NVMX OR. DWT1 .GT. VMISW OR. DWT1. LE. 1)
1 GO TO 735
DELT1=DELT+DWT1
DELT2=DELT+DWT
IF(DELT1 GT DELT) GO TO 735
IV(K, 6)=NV
KA3=K+KOF:
V(KA3)=DELT
IV(KA3, 6)=NV1
KA3=KT+KA3
V(KA3)=DWT1
IF(DELW2 .GE. DELW1) GO TO 739
IV(KT, 7) = NVB
GO TO 739

735 DO 736 I = 1, KC
IF(IV(I, 3).EQ. NVB) GO TO 739
IF(IV(I, ICT).EQ. NVB) GO TO 737

736 CONTINUE
GO TO 739

737 KA2 = I + KOF2
DELT = DELT + V(KA2)
DELV2 = DELW1 + DWT + YMISH
IF(DELV2.GT. DELT) GO TO 739
IV(K, 6) = NVB
KA2 = K + KOF3
V(KA3) = DELW1
IV(KT, 6) = NVB
KA3 = KT + KOF3
KA2 = I + KOF3
V(KA3) = DWT
IF(I, ICT) = 0
V(KA2) = 0.

739 IV(K, 7) = 0
KNV = KNV - 1
GO TO 731

EXCHANGE PAIRS FOR MIN MEASURE

80 IF(KNV .LE. 0 OR. KNV .GT. KC) GO TO 75
DO 801 K = 1, KC
NVB = IV(K, 7)
IF(NVB .LE. 0 OR. NVB .GT. NVMX) GO TO 801
IF(IVU2(NVB).LE. 0) GO TO 801
NC = IV(K, ICS)
IF(NC .LE. 0 OR. NC .GT. NC MX) GO TO 801
NV = IV(K, 2)
DO 802 L = 1, KC
IF(NVB .EQ. IVL, ICT) OR. NVB .EQ. IV(L, 3)) GO TO 803

802 CONTINUE
GO TO 801

803 NCB = IV(L, ICS)
I = IUV2(NVB)
IF(I .LT. 0) I = -1
JX = IC(I, 1)
IF(JX .LE. 0 OR. JX .GT. JM) GO TO 801
DO 807 J = 1, JX
J1 = J + 1
IF(IC(I, J1), EQ. NCB) GO TO 808

807 CONTINUE
GO TO 801

808 DSET = C(I, J)
DUC = UC(NC)
IF(DUC .LT. 0) DUC = -DUC
DELBW = DSET + DUC
KA2 = K + KOF2
KA3 = K + KOF3
DELV1 = V(KA3)
KA = KA - KOFST
IF(DELV1.LE. 1) DELW1 = V(KA2)
IF(DELV1.LE. 1) DELW1 = V(K)
KA2 = L + KOF2
KA3 = L + KOF3
DELV2 = V(KA3)
IF(DELV2.LE. 1) DELW2 = V(KA2)
IF(DELV2.LE. 1) DELW2 = V(L)
DELV = DELW1 + DELW2
IF(DELV .GE. DELWR) GO TO 804
UPDATE ATTRIBUTES

DO 78 K=1, KC
  KA3=K+KOF3
  KA2=K+KOF2
  IF(PRINT2)
    1  WRITE(6,788) IV(K,ICS), IV(K,6), IV(K, ICT),
    2  IV(K,3), V(KA3), V(KA2), V(K)
  788 FORMAT(1X, 415, 3F8.2)
  NC=IV(K,ICS)
  IF(NC LE. 0. OR. NC. GT. NCMX) GO TO 78
  IF(IUC1(NC). LE. 0) GO TO 78
  NV=IV(K,6)
  IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO S10
  DWT=V(KA3)
  GO TO 820
S10  NV=IV(K, ICT)
  IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 811
  DWT=V(KA2)
  GO TO 820
S11  NV=IV(K,3)
  IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 79
  DWT=V(K)
  820 IF(DWT LE. 0. OR. DWT. GT. VMISW) GO TO 79
  IF(IUVI(NV). EQ. 0) GO TO 79
  UCN=DWT
  CALL ATRA:
  CALL BTRA:
  GO TO 77
S9  IF(NVNX. LT. NVMIN) GO TO 7911
  DO 791 I=NVMIN, NVMX
    NVVC=-(I-1)*NVCC
    IVA=9+NVVC
    IVA1=IVA+11
    IF(IVCL(IVA1). EQ. 0) AND. IVCL(IVA). EQ. 0) GO TO 792
  791 CONTINUE
S911  I=NVMX+1
  IF(I. LE. NVARM) GO TO 7921
  PRINT 103, NVARM, I
  C 103 FORMAT(2,R) NO. OF ACTIVE TRACKS EXCEEDS ARRAY MAXIMUM NVMX = ' ,
  C 1 - 2110)
  NV0=NV0+1
  I=NVARM
  GO TO 77
S921  NV=1
S92  NV=1
  NVMIN=I
  UCN=0
  CALL ATRA:
  CALL BTRA:
  77  IUVI(NV)=--NC
  IUC1(NC)=--NV
  UVC(NV)=0.
  UCN(NC)=0.
  78 CONTINUE
DO 99 K=1, NV
NV=IV(K, IVS)
IF(NV, LE. 0. OR. NV. GT. NVMX) GO TO 99
IF(IUV1(NV). LE. 0) GO TO 99
IUV1(NV)=-IUV1(NV)
IF(FNSN. LT. 1.1) GO TO 99
NVA=7+(NV-1)*NVCl
VA=VCL(NVA)
HTC=VA*SINEL+VA*VA+COSPI2
99 CONTINUE
100 RETURN
END
SUBROUTINE COMBIN(N, IS, IT, K, J)
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 N, IS, IT, K, J
COMMON /COMB/ IV(32, 7), IVMX

* INSERT N INTO ORDERED ARRAY IV(K, IT)
* RETURN NEW ARRAY AS IV(K, IS)

I=IS
IS=IT
IT=I
L=0
DO 10 J=1, K
IVJ=IV(J, IT)
IF(IVJ .LT. 0) IVJ=-IVJ
IF(IVJ .EQ. 0) GO TO 40
10 IV(J, IS)=IV(J, IT)
J=K
40 L=1
IV(J, IS)=N
30 DO 50 I=J, K
IL=I+L
50 IV(IL, IS)=IV(I, IT)
K=K+L
IF(K .GE. IVMX) GO TO 70
IV(K, IS)=0
GO TO 80
70 WRITE(6, 100) K, IVMX
100 FORMAT(' ERROR IN COMBIN, 13, K, IVMX', 2X, 2I10)
K=IVMX-1
80 RETURN
END

*
SUBROUTINE STRAK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TM, TA
INTEGER*4 IVCL(736), ITCL(21), IVR(192)
DIMENSION ATST(62), NUM(62)
LOGICAL PRI, PR2, PRIN2
DIMENSION TCL(21), DUM(6)
COMMON /TMAX/ TM
COMMON /FLGS/ I TYPE, PRI, PR2, PRIN2
COMMON /PNTRS/ NCMX, NVMIN, NVMX, IELS, NSCAN, IESNL, NVMCN, NT
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLS/ NVARM, NCAK, NVO, ICO, IO, JO, JYR, KTL
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, A1, A2, A3, B1, B2, HDIV
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /VPARM/ VX, VY
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NAN1
COMMON /DATA2/ VCL(736), MXVC, NVC1
COMMON /DATA3/ VR(192), MXVR, NVR1
EQUIVALENCE(TCL(±), ITCL(l))
EQUIVALENCE'(VCL(±)..IVCL(±)).(VR(i).IVR(i))
DATA IZERO/0/, IOUT/31/, LAN/62/
NVOUT=NVMX-1
DO 19 N=1, LAN
ATST(N)=0

19 NUM(N)=0

VXC=0
VVC=0
NSN=0
NACT=0
NFN=FNSN-1.
IF(NFN LE. 0) GO TO 55
WRITE(2) KTL, NVMX, JDAY, JHR, JMIN, JSEC
       1 , NVN, NSCAN, NT, NVMX, JYR, NVO
       2 , JO, IZERO, VX, VY, DUM
IF(PRI OR PR2)
WRITE(6,1000) JDAY, JHR, JMIN, JSEC, NVSCN
1000 FORMAT(1HL, 3X, 'PROGRAM TRACK 4 (781207)', 5X, 'TRACK TIME', 414,
15X, 'VOL SCAN', 16//, 1X, 'TRK LOCATION DBZ CELL PEAK',
2' SURFACE SUMMIT VELOCITY A'/, 1X
3' ID EAST NORTH AV VOL DBZ HT DBZ HT',
4' EAST NORTH L'/, 1X,
5' KM KM KM KM KM KM',
6' M/S M/S L'
DO 20 NV=1, NVOUT
       NV23=NV+1+NVA
       R=VCL(NV23)
       DO 21 N=1, IOUT
       NN=N+MAN
       IF(A.GT.ATST(NN)) GO TO 22

21 CONTINUE
       NN=NN+1

22 LAN=MAN
       MAN=MAN+IOUT
       IF(MAN GT. IOUT) MAN=0
       IFNL=0
DO 20 N=1, IOUT
   NN=M+MAM
   ML=M+LAN
   IF(ML .NE. NN OR. IFML .EQ. 1) GO TO 24
   ATST(MM)=A
   NUM(MM)=NV
   LAN=LAN-1
   IFML=1
   GO TO 23
24 ATST(MM)=ATST(ML)
   NUM(MM)=NUM(ML)
23 CONTINUE
20 CONTINUE
50 DO 100 N=1, IOUT
   NN=M+MAM
   NV=NUM(NN)
   IF(NV .EQ. 0) GO TO 100
   NVV=(NV-1)*NV1
   NV1=1+NVV
   NV2=NV1+1
   NV3=NV2+1
   NV4=NV3+1
   NV5=NV4+1
   NV6=NV5+1
   NV7=NV6+1
   NV8=NV7+1
   NV9=NV8+1
   NV10=NV9+1
   NV11=NV10+1
   NV12=NV11+1
   NV13=NV12+1
   NV14=NV13+1
   NV15=NV14+1
   NV16=NV15+1
   NV17=NV16+1
   NV18=NV17+1
   NV19=NV18+1
   NV20=NV19+1
   NV21=NV20+1
   NV22=NV21+1
   NV23=NV22+1
   NR1=1+(NV-1)*NVR1
   NR2=NVR1+1
   NR3=NVR2+1
   NR4=NVR3+1
   NR5=NVR4+1
   NR6=NVR5+1
   IF(NVCL(NV9) .LE. 0) GO TO 200
   ZZDIV=1.0/VCL(NV10)
   ITCL(1)=IVCL(NV5)
   TCL(2)=VCL(NV11)*ZZDIV
   TCL(3)=VCL(NV12)*ZZDIV
   TCL(4)=10.*ALOG10(VCL(NV10)/FLOAT(IVCL(NV9)))
   TCL(5)=VCL(NV13)*ZZDIV
   ITCL(7)=IVCL(NV16)
   TCL(8)=VCL(NV17)
   ITCL(10)=IVCL(NV3)
   TCL(11)=VCL(NV4)
   TCL(12)=VCL(NV6)
   IM3=IVCL(NV16)-3
   30 ITCL(15)=IVCL(NV15)
   TCL(16)=VCL(NV14)
   IVL8=IVCL(NV8)
   IF(IVL8 .LT. 0) IVL8=-IVL8
   ITCL(17)=IVL8
   ITCL(18)=0
ITCL(20) = 0
ITCL(21) = 0
VXT = VCL(NV21)
VYT = VCL(NV22)
ITIM = IVCL(NV20)
JTIM = ITCL(1)
IF (ITIM. EQ. 0) OR. ITIM. EQ. JTIM) GO TO 40
DELTM = JTIM - ITIM
DELTM = 1.0 / DELTM
TCL(20) = (TCL(2) - VCL(NV19)) * DELTM
TCL(21) = (TCL(3) - VCL(NV19)) * DELTM
VXT = TCL(20)
VYT = TCL(21)
40 IVCL(NV20) = ITCL(1)
VCL(NV19) = TCL(3)
VCL(NV18) = TCL(2)
VCL(NV21) = A1*VXT + A2*VCL(NV21) + A3*VX
VCL(NV22) = A1*VYT + A2*VCL(NV22) + A3*VY
VR(NR1) = TCL(2)
VR(NR2) = TCL(3)
VR(NR3) = TCL(4)
VR(NR4) = VCL(NV4)
IVR(NR5) = ITCL(1)
VR(NR6) = TCL(6)
IZVAL = VR(NR3)
IDTC = FLOAT(VCL(NV9)) / (FNSN - 1.0) * 10. + 5
NSN = NSN + 1
VXC = VX + TCL(NV20)
VYC = VY + TCL(NV21)
VXP = VCL(NV21) * 1000.
VYP = VCL(NV22) * 1000.
NACT = NACT + 1
WRITE(2) TCL, VCL(NV39), VCL(NV40), IDTC
IF (PR1) GO TO 59
IF (NOT. PR2) GO TO 59
IF (IDTC. GT. MNSN. OR. IZVAL. GT. 35). AND. IVCL(NV8). GT. 0)
1 IVCL(NV8) = IVCL(NV8)
IF (IVCL(NV8). GE. 0) GO TO 59
WRITE(6, 1005) ITCL(17), VR(NR1), VR(NR2), IZVAL, TCL(5), ITCL(7),
ITCL(8), ITCL(10), TCL(12), ITCL(15), TCL(16),
2VXP, VYP, IDTC
1005 FORMAT(IX, 14, 2F5.0, 13, F6.1, 13, F5.1, 14, + 3F5.1, 2X, 12)
59 DO 41 I = 9, 14
IA = I + NVR
41 IVCL(IA) = 0
IVCL(NV23) = 0
GO TO 100
200 IF (IVCL(NV20). LE. 0) GO TO 102
IF (<T-IVCL(NV20)). LE. TM) GO TO 100
DO 101 I = 1, 20
IA = I + NVR
101 IVCL(IA) = 0
IA = IA + NVR
VCL(IA) = 0
IF (NV. LT. NVMIN) NVMIN = NV
202 IVCL(NV23) = 0
100 CONTINUE
IF (NSN. EQ. 0) GO TO 56
VN = NSN
VX = B1*VXC/VN + B2*VX
VY = B1*VYC/VN + B2*VY
DO 43 JI = 1, IOUT
II = II + 1
I = NUM(I1)
IF (I. EQ. 0) GO TO 43
IF(IVCL(NVA).GT.0) GO TO 43
NA=21+(I-1)*NVC1
VCL(NA)=VX
NA=NA+1
VCL(NA)=VY
43 CONTINUE
VXP=VX*1000.
VYP=VY*1000.
56 IF(NVSCN.NE.1) GO TO 55
WRITE(6,1003)
1003 FORMAT(///IX. VOL START TIME NO EL LAST TRACK'
1 ' OVERFLOW AVG VELOCITY'///, 1X',
2 'SCAN DAY HMMM SS SCANS SCAN TOTAL ACT.'///,
3 ' NV IC I J EAST NORTH')
55 WRITE(6,1004) NVSCN, JDAY, JHR, JMIN, JSEC, FNSN, NSCAN, NT, NACT
1004 FORMAT(1X, I5, I5, I5, I5, I5, I5, I3, I8, 2F6 1)
NVSCN=NVSCN+1
NVO=0
ICO=0
IO=0
JO=0
JDAY=IDAY
JHR=IHR
JMIN=MIN
JSEC=ISEC
FNSN=1.089
KTL=T
RETURN
END
APPENDIX I

SAMPLE OUTPUT
SAMPLE OUTPUT FILE

Two types of output are produced following each volume scan:

1. A list of attributes describing each of the 31* most significant cells detected and tracked within the volume scan
   a. ITCL (17) - the ID number of this cell track
   b. VR (1) - cell location to east [(-)west] of radar (km)
   c. VR (2) - cell location to north [(-)south] of radar (km)
   d. IZVAL = VR (3) - average reflectivity of cell (dBZ)
   e. TCL (5) - cell volume (km\(^3\))
   f. ITCL (7) - peak reflectivity of cell (dBZ)
   g. TCL (8) - height of peak reflectivity (km)
   h. ITCL (10) - reflectivity at base of cell (dBZ)
   i. TCL (12) - height of cell base (km)
   j. ITCL (15) - reflectivity at summit of cell (dBZ)
   k. TCL (16) - height of cell summit (km)
   l. VXP = VCL (39)*10\(^3\) - cell velocity toward east [(-)west] (m/s)
   m. VY = VCL (40)*10\(^3\) - cell velocity toward north [(-)south] (m/s)
   n. ID'C = IVCL (9)/(FNSN-l)*10 - number of scans cell detected/number of scans per volume scan percent of elevation scans cell detected at

2. A summary list of statistics for the entire volume scan
   a. NV;CN - volume scan number
   b. JD:Y, JHR, JMIN, JSEC - start time of volume scan
   c. NFN = FNSN-1 - number of elevation scans processed in this volume scan
   d. NSCAN - number of last elevation scan in the volume scan
   e. NT - number of cell tracks updated this scan
   f. NACT - number of possible cell tracks stored from current and previous scans
   g. NVO - number of significant cells detected but not outputted
   h. ICO - number of internally paired and clustered cells over the dimensions of CLUST array in subroutine RESOLV
   i. IO - number of detected cells over array dimensions of cluster array in subroutine COMPAR
   j. JO - number of detected cells over array dimensions of IC array in subroutine COMPAR
   k. VXP - an estimate of the average velocity east [(-)west] any new cells will have (m/s), set to 0 on first scan
   l. VY - an estimate of the average velocity north [(-)south] any new cells will have (m/s), set to 0 on first scan

*see Appendix F - Option to Increase Number of Significant Cells
<table>
<thead>
<tr>
<th>TRK ID</th>
<th>EAST</th>
<th>NORTH</th>
<th>DBZ</th>
<th>PEAK</th>
<th>SURFACE</th>
<th>SUMMIT</th>
<th>VELOCITY</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 900</td>
<td>40 45</td>
<td>1.5 45</td>
<td>4.3 45</td>
<td>4.3 45</td>
<td>4.3 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>1 900</td>
<td>63 34</td>
<td>2.7 34</td>
<td>1.8 34</td>
<td>1.8 34</td>
<td>1.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>2 900</td>
<td>37 41</td>
<td>4.0 41</td>
<td>2.3 41</td>
<td>2.3 41</td>
<td>2.3 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>18 890</td>
<td>32 47</td>
<td>2.7 47</td>
<td>4.1 47</td>
<td>4.1 47</td>
<td>4.1 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>1 900</td>
<td>67 42</td>
<td>7.8 42</td>
<td>6.7 42</td>
<td>6.7 42</td>
<td>6.7 0 0</td>
<td>0 0</td>
<td>7 0</td>
<td></td>
</tr>
<tr>
<td>19 800</td>
<td>36 36</td>
<td>5.4 36</td>
<td>6.8 36</td>
<td>6.8 36</td>
<td>6.8 0 0</td>
<td>0 0</td>
<td>7 0</td>
<td></td>
</tr>
<tr>
<td>20 800</td>
<td>36 37</td>
<td>2.6 36</td>
<td>3.9 36</td>
<td>3.9 36</td>
<td>3.9 0 0</td>
<td>0 0</td>
<td>7 0</td>
<td></td>
</tr>
<tr>
<td>21 900</td>
<td>33 39</td>
<td>1.8 39</td>
<td>4.8 39</td>
<td>4.8 39</td>
<td>4.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>22 900</td>
<td>36 36</td>
<td>2.4 36</td>
<td>1.7 36</td>
<td>1.7 36</td>
<td>1.7 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>23 900</td>
<td>33 39</td>
<td>1.8 39</td>
<td>4.8 39</td>
<td>4.8 39</td>
<td>4.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>24 900</td>
<td>36 36</td>
<td>2.4 36</td>
<td>1.7 36</td>
<td>1.7 36</td>
<td>1.7 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>25 900</td>
<td>33 39</td>
<td>1.8 39</td>
<td>4.8 39</td>
<td>4.8 39</td>
<td>4.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>26 900</td>
<td>36 36</td>
<td>2.4 36</td>
<td>1.7 36</td>
<td>1.7 36</td>
<td>1.7 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>27 900</td>
<td>33 39</td>
<td>1.8 39</td>
<td>4.8 39</td>
<td>4.8 39</td>
<td>4.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>28 900</td>
<td>36 36</td>
<td>2.4 36</td>
<td>1.7 36</td>
<td>1.7 36</td>
<td>1.7 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
<tr>
<td>29 900</td>
<td>33 39</td>
<td>1.8 39</td>
<td>4.8 39</td>
<td>4.8 39</td>
<td>4.8 0 0</td>
<td>0 0</td>
<td>3 0</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX F

Options to Increase:
Number of Significant Cells
Number of Active Tracks
Option: Increase Number of Significant Cells Processed Each Scan

Both the real-time program and the post-mission program are set up to process 16 significant cells each scan. To increase the number of cells the dimensions of several arrays and address offsets must be increased.

In block common DATA1 - for each additional cell
   increase dimension of ECL by 14
   increase NOFST* by 1
   increase IC1AB* by 7

In block common RSLV - for each additional cell
   increase dimension of: IUC1, IUC2GC by 1

In block common NVLIS - for each additional cell
   increase NCARM* by 1

In subroutine RESOLV - for each additional cell
   increase dimension of V by 24
   increase KOE2KOF3 by 8

Note: to process more than 32 cells each scan the number of active tracks must be increased to at least that number (see option to increase active tracks).

*variable set in block DATA
Option: Increase Number of Active Tracks Updated Each Scan and Outputted Each Volume Scan

Both the real-time program and the post-mission program are set up to update and output 31 active tracks at any one scan. To increase the number of tracks, the dimensions of several arrays and address offsets must be increased.

In block common DATA2 - for each additional track
  increase dimension of VCL and IVCL by 23
  increase MXVC* by 23

In block common DATA3 - for each additional track
  increase dimension of VR and IVR by 6
  increase MXVR* by 7

In block common RSLV - for each additional track
  increase dimension of: IHV1, IHV2&IHV by 1

In block common CDRAYS - for each additional track
  increase the first dimension of: IC, IC1&D by 1
  increase IM* by 1

In block common NVL/1S - for each additional track
  increase NVARM* by 1

In block common COMB - for each additional track
  increase the first dimension of IV by 1

In subroutine STRAK - for each additional track
  increase dimension of ATST & NUM by 2
  increase IOUT by 1
  increase IAN by 2

*variable set in block DATA