NOAA Technical Memorandum ERL ESG-10

AN AUTOMATED METHOD FOR THE DOCUMENTATION OF CLOUD-TOP CHARACTERISTICS OF MESOSCALE CONVECTIVE SYSTEMS

John A. Augustine

Environmental Sciences Group
Boulder, Colorado
February 1985

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL ESG-10

AN AUTOMATED METHOD FOR THE DOCUMENTATION OF CLOUD-TOP CHARACTERISTICS OF MESOSCALE CONVECTIVE SYSTEMS

John A. Augustine
Weather Research Program

Environmental Sciences Group
Boulder, Colorado
February 1985

UNITED STATES
DEPARTMENT OF COMMERCE

Malcolm Baldrige,
Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION

Environmental Research
Laboratories

Vernon E. Derr,
Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA Environmental Research Laboratories. Use for publicity advertising purposes of information from this publication concerning proprietary products or tests of such products is not authorized.

CONTENTS

AN AUTOMATED METHOD FOR THE DOCUMENTATION OF CLOUD-TOP CHARACTERISTICS OF MESOSCALE CONVECTIVE SYSTEMS

John A. Augustine

ABSTRACT. Measurements of cloud-top area, centroid, and eccentricity for mesoscale convective systems (MCS's) have been automated. The algorithm resides on the PROFS-2 VAX 11/780 computer and utilizes digital infrared imagery received at the local satellite ground station. The software is user-activated and processes one image per run. All operations are performed within the bounds of a user-defined image subsector. Cloud tops are isolated at a threshold of -52°C; if the area at this threshold exceeds 10,000 km^2, the storm is documented. Documentation includes area and centroid measurements at -52°C, -58°C, -64°C, -70°C, and -76°C. Also, the eccentricity of the storm is estimated by fitting an ellipse to the -52°C envelope and computing the ratio of the minor to major axis. Results are written to file for later printing, and to a RAMTEK monitor. On the monitor, the processed sector is displayed with a geographical map background, cloud tops are color-coded according to the five threshold temperatures, and the major and minor axes' endpoints of best-fit ellipses to documented storms are highlighted. Tests have shown that the algorithm provides acceptable results as long as storm cloud tops are isolated from one another and are well defined. However, on occasions when storms are merged at the cloud definition threshold (-52°C), the algorithm has no way to objectively separate them, and thus erroneously documents the merged storms as one large MCS.

I. INTRODUCTION

A. Background

Mesoscale Convective Systems (MCS's) are convective storms that have a lateral dimension of 250 - 2500 km. A specialized subset of this category of storm is the Mesoscale Convective Complex (MCC). MCC's, identified by Maddox (1980), are huge nocturnal, circular convective storms that occur over the central United States during the convective season (April through September). A typical example of an MCC is shown in Fig. 1. Characterization

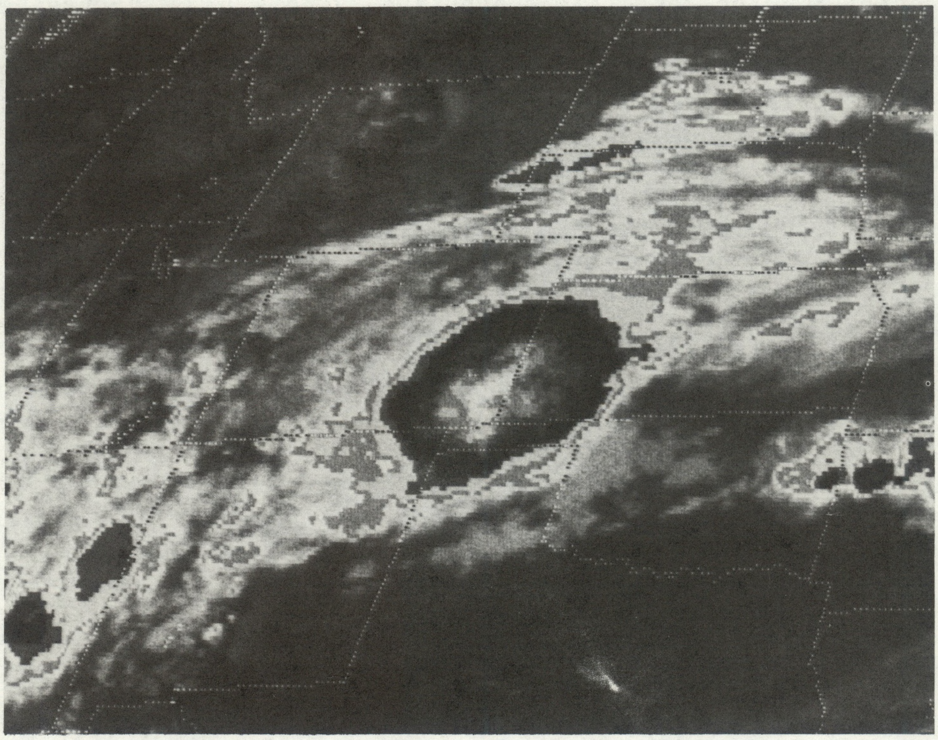


Figure 1. Thermal infrared imagery of a typical MCC enhanced by the MB-enhancement curve.

is based on features observable on MB-enhanced hardcopy infrared imagery from the Geostationary Operational Environmental Satellites (GOES). The MB-enhancement is described in Fig. 2. According to Maddox (1980), the scale of MCC's is two orders of magnitude greater than that of individual thunderstorms. Specifically, these criteria define MCC's:

- The continuous cloud shield ⩽ -32°C must be greater than 100,000 km^2
- The interior cloud area ⩽ -52°C must be greater than 50,000 km^2
- The above area thresholds must be maintained for at least 6 hours
- The storm must be circular in shape, that is, the eccentricity, defined as the ratio of the minor to the major axis at the time of the -32°C area's maximum extent, must exceed .7.

The physical considerations that form the basis of these criteria are given in Maddox (1980).

Each year since 1981, an annual summary of all MCC's that occurred over the United States has been prepared (Maddox et al., 1982, Rodgers et al., 1983). Duration, time of initiation, time of maximum extent, time of termination, cloud shield area at both -32°C and -52°C thresholds, and storm tracks are included for each storm. In addition, a 5-year climatology of all MCS's from 1979 through 1983 has been prepared by Bartels et al. (1984).

In the past, availability of GOES hardcopy imagery only made this job time consuming and cumbersome. Threshold areas were measured manually with a planimeter and area-weighted centroids for storm tracking were approximated. The eccentricity measurement required transfer of the perimeter of the -32°C area to a basemap from which accurate distances could be computed. The time

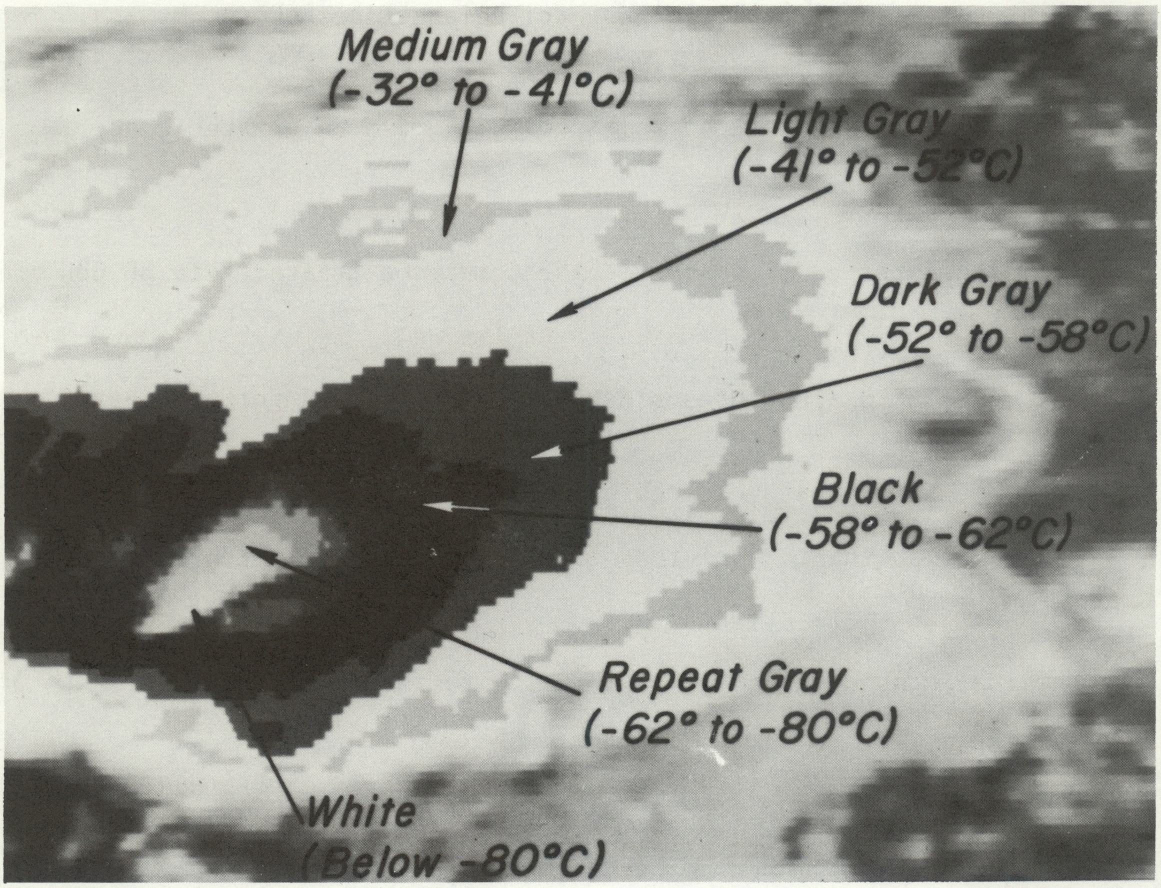


Figure 2. Graphic depiction of the MB enhancement (from GOES User's Guide, 1983).

required to complete documentation of a full year's MCS's and MCC's was prohibitive, and consequently only MCC's were considered.

With the introduction of real-time digital GOES imagery at the Program for Regional Observing and Forecasting Services (PROFS) at the Environmental Research Laboratories (ERL) in Boulder, this situation has changed. The availability of infrared digital imagery from the NOAA-NCAR operated satellite ground station has enabled much of the MCS documentation procedure to be automated. Moreover, no longer are the measurements tied to only those temperatures dictated by the MB-enhancement curve. Mathematical procedures to compute accurate cloud-top areas, centroids, and eccentricity have been developed and coded into a working software package. This report describes these procedures and corresponding software, which will be used for MCC and MCS classification from the 1985 convective season onward.

B. Overview of the automated method

The new system, although not totally automatic, introduces efficiency into the MCC documentation procedure. Hardcopy imagery must still be examined prior to running the software to determine which digital images are to be processed. Execution is initiated by the user, and only one digital image is processed per run. A description of the data to be processed is entered by user response to a few questions that appear on the terminal CRT. Two products are generated: (1) an output file detailing threshold areas, centroids, and eccentricity information for every storm in the image whose -52°C area is greater than 10,000 km^2, and (2) a color-coded video display of the processed image.

The automated method of documenting MCS's differs from the manual method in that the -32°C area is not considered. All measurements -- area, centroid, and eccentricity -- are calculated at -52°C. Areas and centroids for colder thresholds at -58°C, -64°C, -70°C, and -76°C are also computed if any part of the cloud top exceeds those thresholds. The decision to ignore the -32°C area was made after several months of testing revealed that the -32°C area unique to a particular storm could not always be determined objectively. On several occasions more than one mesoscale event would be encompassed by a single -32°C envelope (Fig. 3). These situations could be handled subjectively, but not objectively. In contrast, the -52°C area appeared to delineate individual storms well most of the time. On this observation, the decision was made to document these storms at the colder threshold.

The shift of the cloud definition threshold area from -32°C to -52°C may not affect the annual summaries to a great extent. The -52°C area provides at least one of the original MCC area measurement criteria. With this area as a benchmark, the -32°C area, if needed, could be approximated from the hardcopy image or the video display product. Eccentricity, originally measured for the -32°C envelope, is also measured from the -52°C area in the automated scheme. However, because eccentricity represents the ratio of the minor to major axis, that for the -52°C area should be similar to that for the -32°C area, at least for a symmetric storm where the -32°C and -52°C areas are the same shape. If not, a subjective adjustment could be made easily.

II. DATA

A. Satellite characteristics

In order to appreciate the mathematical procedures and idiosyncrasies of

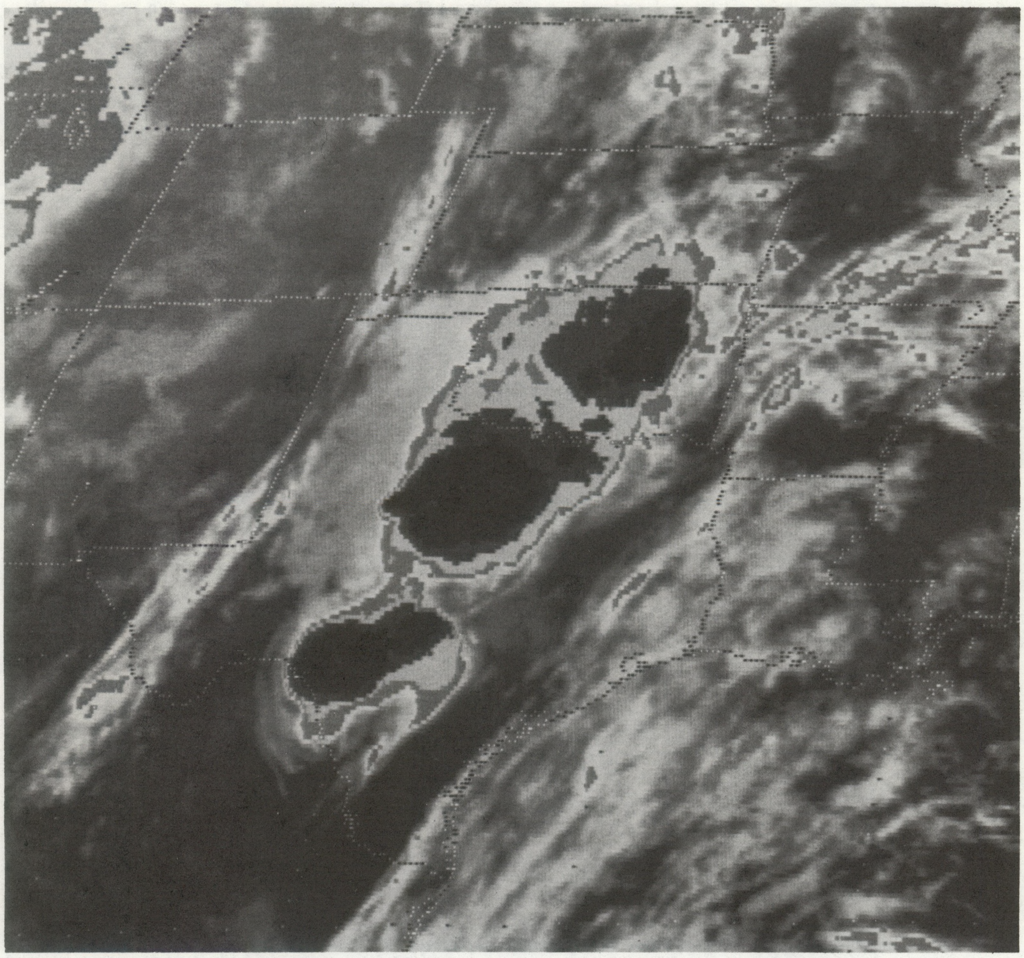


Figure 3. Example of three distinct MCS's encompassed by one -32°C envelope.

the calculations applied to the digital imagery in the automated procedure, it is important to understand the properties of GOES spacecrafts and the data they provide. By virtue of being geostationary, these satellites are locked into equatorial orbits at a nominal altitude of approximately 35,800 km. Imaging is carried out by the Visible and Infrared Spin Scan Radiometer (VISSR). Visible radiance is recorded in the 0.55 to 0.75 μm range of the electro-magnetic spectrum, and infrared radiance is sensed in the infrared window between 10.5 and 12.6 μm . The VISSR instrument is fixed on the side of the satellite, and scan lines are produced as the spinning spacecraft sweeps these sensors across the Earth from west to east. One pass of the VISSR across the Earth during a particular rotation results in one scan line. Along each scan, data are sampled every 84 microradians. This translates to an east-west subpoint pixel dimension of 3.0 km. After each complete satellite rotation the mirror that directs incoming radiation into the VISSR is stepped down by 192 microradians, which translates to a north-south subpoint pixel dimension of 6.9 km. The areal extent of the subpoint infrared pixel is therefore 20.70 km^2. One full disk infrared image is made up of 1821 scan lines, each containing 3822 elements. With the satellite spinning at a rate of 100 rpm, the full disk image requires 18.21 minutes to complete. This, combined with the time to back up the mirror to the top of the Earth, accounts for about 30 minutes per image.

Nominally, two GOES platforms provide coverage of the United States; they reside over the Equator at 75°W (GOES-East) and 135°W (GOES-West). Their respective coverages are shown in Fig. 4. With the exception of special cases when high temporal frequency imagery is requested, GOES-East provides an image on the hour and 30 minutes past the hour; GOES-West's imagery is staggered between those times at 15 and 45 minutes past the hour. However, owing to the

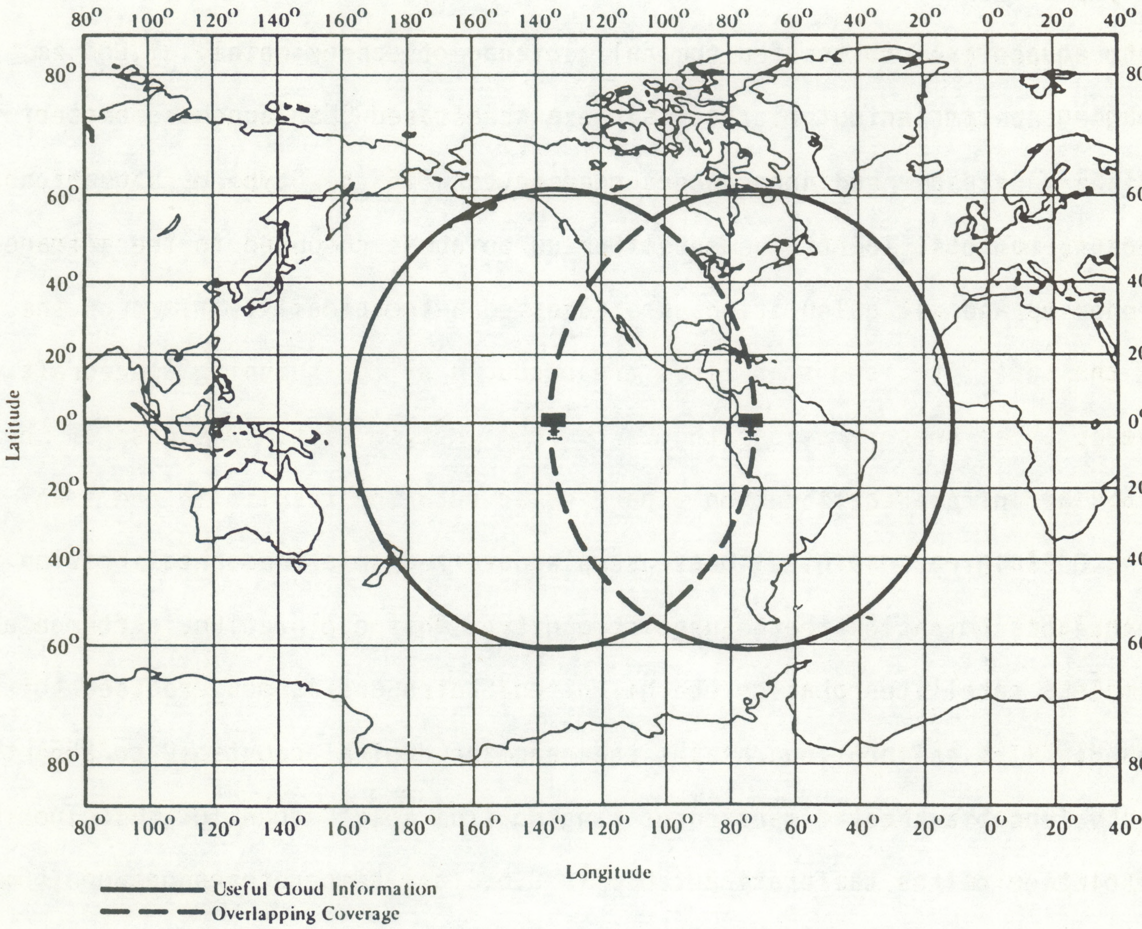


Figure 4. GOES-East and GOES-West coverage areas (from GOES User's Guide, 1983).

failure of GOES-East in the summer of 1984, only GOES-West is operational at the present time. To accommodate this loss, GOES-West has been moved eastward to approximately 98°W and has assumed the time schedule of GOES-East. This central position allows for the full coverage of the continental United States in the interim period prior to the next scheduled GOES launch. Users of this software package need not concern themselves with this type of situation because the position of the satellite subpoint is computed for each image processed and all calculations are adjusted automatically.

B. The infrared calibration

GOES infrared digital data are standardized to a common calibration table before dissemination to the user community. This calibration is common among all GOES satellites that cover the United States and is not expected to change. The calibration relates the range of digital counts (0 to 255) to equivalent blackbody temperatures ranging from 56.8°C to -110.2°C. The resolution of the calibration table is 0.5°C for temperatures between 56.8°C and -31.2°C, and 1.0°C for the range -31.2°C through -110.2°C. This table is given in Smith and Vonder Haar (1976). Digital counts corresponding to the five temperature thresholds used for MCS documentation (-52°C, -58°C, -64°C, -70°C, and -76°C) are 197, 203, 209, 215, and 221, respectively. These thresholds are fixed in the software and cannot be changed by the user unless the source code is modified.

C. The satellite data base

The Boulder ground station consists of three antennae and two processors dedicated to satellite data ingest. Consequently, data from two satellites may be ingested simultaneously. However no firm operational schedule governs the ground station. At times, data from both satellites (when they are both functional) is available, and at other times, imagery from only one is ingested. With respect to MCS documentation algorithm, GOES-East imagery is preferable to that from GOES-West, because, as is evident in Fig. 4, the eastern satellite's subpoint is more central to the region where MCS's typically occur (east of the Rocky Mountains), than GOES-West's. At the present time, there is no such choice because of the failure of GOES-East. However, the new position of GOES-West at 98°W offers a better perspective of the central United States than did GOES-East and thus is optimal for MCS documentation.

At PROFS, ingested digital imagery is sectorized and written to disk files in the SATDAT directory on a disk logically associated with the name DATA_DEV. As new imagery are ingested and written to file, the oldest data files are deleted. Residence time of an individual day of satellite data is typically 48 hours, although the temporal resolution is reduced as the data age. The latest 24 hours are usually available at half-hour intervals; after that, the temporal frequency is reduced to 3 hours. Therefore, to ensure that a complete set of digital imagery is available for a particular MCS, it is advisable that the documentation software be run within 24 hours of a particular event. In special cases, image files may be spooled to tape for later analysis.

The "infrared large sector" accessed by the MCS documentation software is 360 infrared lines by 785 infrared elements. Pixels in the PROFS infrared sectors are twice as large as raw GOES infrared pixels. This is a consequence of the sectorizing routine used at PROFS, which averages pairs of adjacent pixels as the images are sectorized. For example, if the subpoint pixel were included in one of the sectors, it would have dimensions of 6.0 X 6.9 km rather than 3.0 X 6.9 km. When the two GOES platforms are operating in their nominal positions the GOES-East infrared large sector is centered on 36°N and 93°W, and that for GOES-West data is centered on 36°N and 100°W. (Since the failure of GOES-East and subsequent movement of GOES-West, the current sector has its center at 36°N and 98°W.) Geographic coverage of this large sector varies by a small amount depending on which satellite is accessed, but it essentially covers the entire continental United States, southern Canada, and northern Mexico.

III. PROGRAM DESCRIPTION

A. Background information

Source code for the MCS documentation software is written in VAX-11 FORTRAN, a superset of FORTRAN 77. It was developed on a Digital Equipment Corporation (DEC) VAX 11/780 computer and makes use of many conventions unique to the DEC VMS operating system (i.e., variable word sizes, logical names, VMS intrinsic functions, etc). Also, many software libraries unique to the PROFS computer system are accessed, especially those routines that deal with retrieving specialized data sets; these reside in PROFS' Meteorological Data Access Library (MDAL). Owing to these constraints, and the liberal use of virtual memory, the MCS documentation package is executable only on the PROFS network of DEC computers, or on a PROFS-compatible system.

A listing of the source code is presented in APPENDIX C. This listing does not include any library subroutines, system functions, or DEC systemm routines that are accessed. Such routines are listed or described in VAX manuals or the library documentation files that reside on the PROFS computer network. Although lengthy, the program is modular and well documented.

B. The documentation software

The MCS documentation software is designed to process one image per run. This is accomplished in five sequential subroutine calls in the main program. The functions of these five steps are to (1) initialize, (2) produce a subsector of the PROFS infrared large image, (3) produce a pixel area correction table, (4) locate and document all significant cloud tops colder than -52°C in the sector, and (5) display the products. These steps are executed sequentially in one complete, undisturbed run. A simplified flow diagram of the software is given in Fig. 5, where subroutine calls of the main program are listed in the left column in the order that they are called. The following discussions of this software emphasize the data sector, and the cloud top area, centroid, and eccentricity calculations.

1. The subsector

The infrared large sector accessible from the PROFS satellite data base provides adequate coverage of the geographical area over which MCS's occur climatologically, but it is too large for efficient operation of the MCS documentation package. Moreover, on a particular day, MCS's may occur only over relatively small regions where meteorological conditions support the development and subsistence of these specialized convective storms, while other areas within the large data sector are relatively inactive. Therefore,

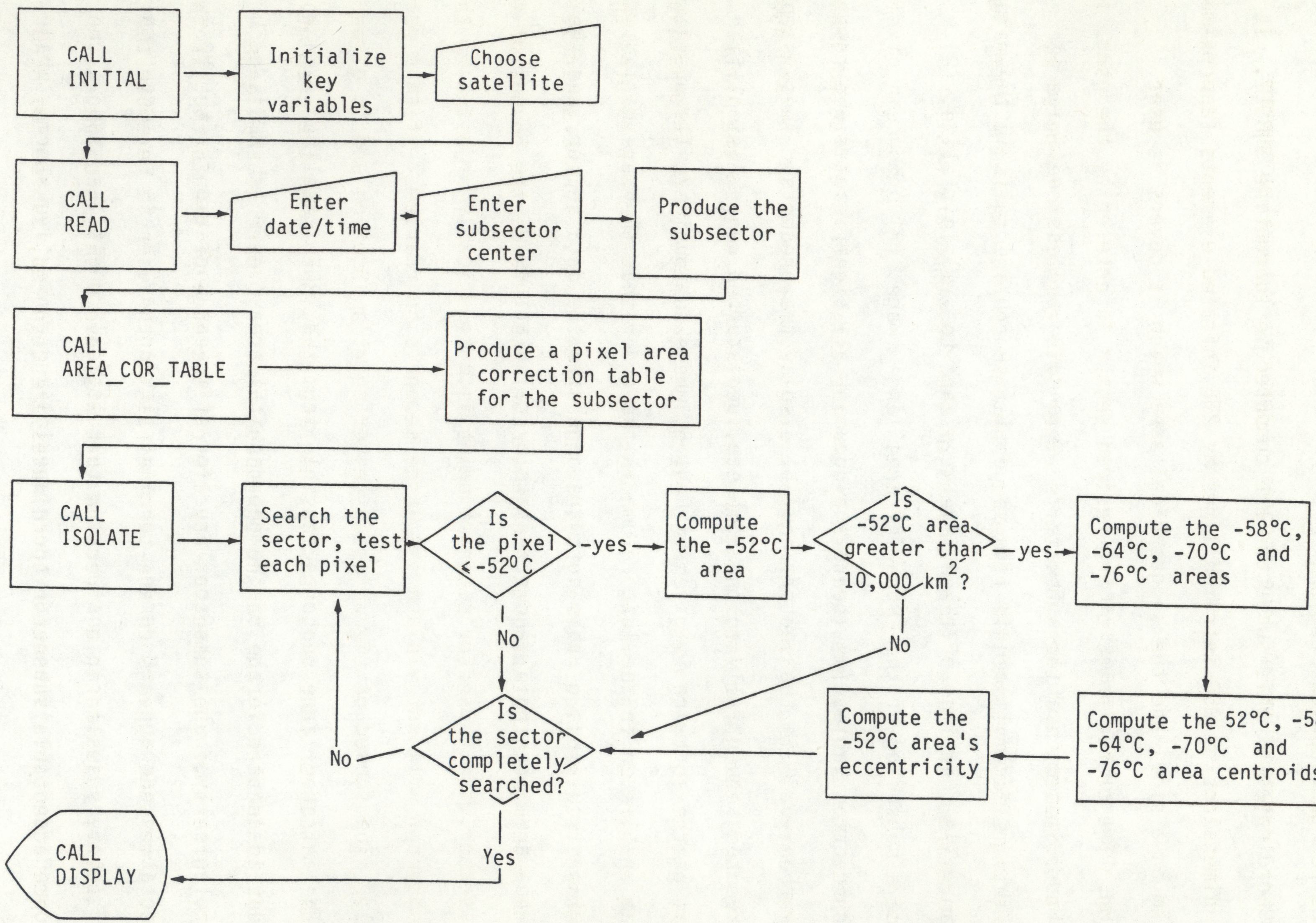


Figure 5. Simplified flow chart for the MCS documentation software. The five subroutine calls of the main program are in the left column.

the first function of this package is to extract a data sector that includes the desired geographical region.

Coordinates of this subsector are computed in subroutine SUBSET. It has fixed dimensions of 250 infrared lines by 256 infrared elements (approximately 2000 km on a side) and the geographical area which it covers is user defined. The center point of the desired sector is entered by the user, in Earth coordinates, and the subsector's upper left corner is computed by backspacing 125 lines and 128 elements from that point. With the upper left corner defined, the other three corners of the subsector are easily computed.

These coordinates are verified before the subsector is extracted from the infrared large image. First, the user-entered center point is checked against the bounds of the large image. If the center point is within the large sector, execution continues; if not, execution terminates. An informative message appears on the terminal in either case. The next check ensures that the subsector area lies entirely within the large image. If the sectorizing routine determines that part of the subsector lies outside the bounds of the large sector, the subsector's center point is automatically adjusted so that the subsector will not violate the large sector's boundaries. If this occurs, the user is informed of the adjusted center point's location and is given the options to stop or continue. Lastly, on rare occasions, when the area of interest is located in the far northwest or northeast part of the large sector, corners of the subsector may lie off the edge of the Earth. If this occurs, a message appears on the terminal asking the user to re-enter the center point, adjusted in a direction that will avoid this problem.

Once acceptable subsector coordinates are computed, subroutine READ selectively extracts the subsector from the infrared large image and loads it into an array named ISEC. Values of individual elements in ISEC are represented by digital counts ranging from 0 to 255 and are related to blackbody temperature by the standard GOES infrared calibration. Finally, the border values of ISEC are set to zero to ensure that all cloud tops at, or colder than, the -52°C threshold have definite boundaries.

2. MCS documentation

After the subsector has been loaded into ISEC, each -52°C cloud top area within the subsector is processed individually. This is the focal point of the software package and is carried out by the multifunctional subroutine ISOLATE, and its many auxiliary subroutines. The first function of ISOLATE is to locate the first pixel in the subsector at, or colder than, the -52°C threshold. The ISEC array is searched by row, beginning with the first row. When the first qualified pixel is found, its digital count value is stored in the first element of the one-dimensional array IVALUE, and its location is saved in the IANDJ array. In the example in Fig. 6, the first pixel at the -52°C threshold (digital count 197) of the schematic cloud top was found at location (3,2). Beginning at the initial pixel, a search commences to locate all contiguous pixels belonging to that -52°C cloud top. As pixels at or above the threshold are located, their digital count values and locations are stored in IVALUE and IANDJ sequentially. Also, their positions in the ISEC array are set to zero to ensure that they are not relocated. In Fig. 6a, pixels belonging to the schematic -52°C cloud top are numbered in the order that the search routine would have found them. Examples of the structures of the IVALUE and IANDJ arrays for that cloud top are given in Fig. 6b.

After the entire cloud top has been isolated and stored, the search

SUBSECTOR ARRAY (ISEC)

(a)

x →

y ↓	1	2	3	4	5	6	7	8	9	•••	256
1	00	00	00	00	00	00	00	00	00	•••	00
2	00	195	[1] 197	[2] 198	[4] 197	195	193	191	191	•••	00
3	00	192	195	[3] 197	[5] 199	[6] 197	195	192	192	•••	00
4	00	190	194	195	[7] 198	[8] 197	195	194	193	•••	00
5	00	192	196	[9] 205	[10] 217	[11] 218	[12] 205	196	194	•••	00
6	00	192	195	196	[14] 198	[13] 197	196	194	192	•••	00
7	00	190	194	196	[15] 197	196	194	193	191	•••	00
8	00	189	192	194	195	194	193	192	190	•••	00
9	00	188	190	191	192	191	190	190	190	•••	00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
250	00	00	00	00	00	00	00	00	00	•••	00

CLOUD TOP STORAGE ARRAYS

(b)

IVALUE (unsorted)	IANDJ	IVALUE (sorted)	IANDJ
197	3,2	218	6,5
198	4,2	217	5,5
197	4,3	205	4,5
197	5,2	205	7,5
199	5,3	199	5,3
197	6,3	198	4,2
198	5,4	198	5,4
197	6,4	198	5,6
205	4,5	197	3,2
217	5,5	197	4,3
218	6,5	197	5,2
205	7,5	197	6,3
197	6,6	197	6,4
198	5,6	197	6,6
197	5,7	197	5,7

Figure 6. (a) Schematic depiction of a -52°C (197 digital count) cloud top in the ISEC array; pixels are numbered in the order that they were located. (b) Pixel digital count values (IVALUE) and their corresponding (x,y) locations (IANDJ) listed in the order that they were located (left pair). The sorted IVALUE and corresponding IANDJ arrays for the isolated cloud top are also shown (right pair).

terminates and subroutine DOCUMENT is called. There, the -52°C area is computed first; if it is greater than 10,000 km^2, DOCUMENT goes on to fully document the cloud top. If the -52°C area is not greater than the 10,000 km^2, that cloud top is ignored. After an MCS has been documented completely, control is returned to ISOLATE where the process of isolating and documenting qualified cloud tops continues until the entire subsector has been searched. At that point, control is returned to the main program where only the display of the products remains.

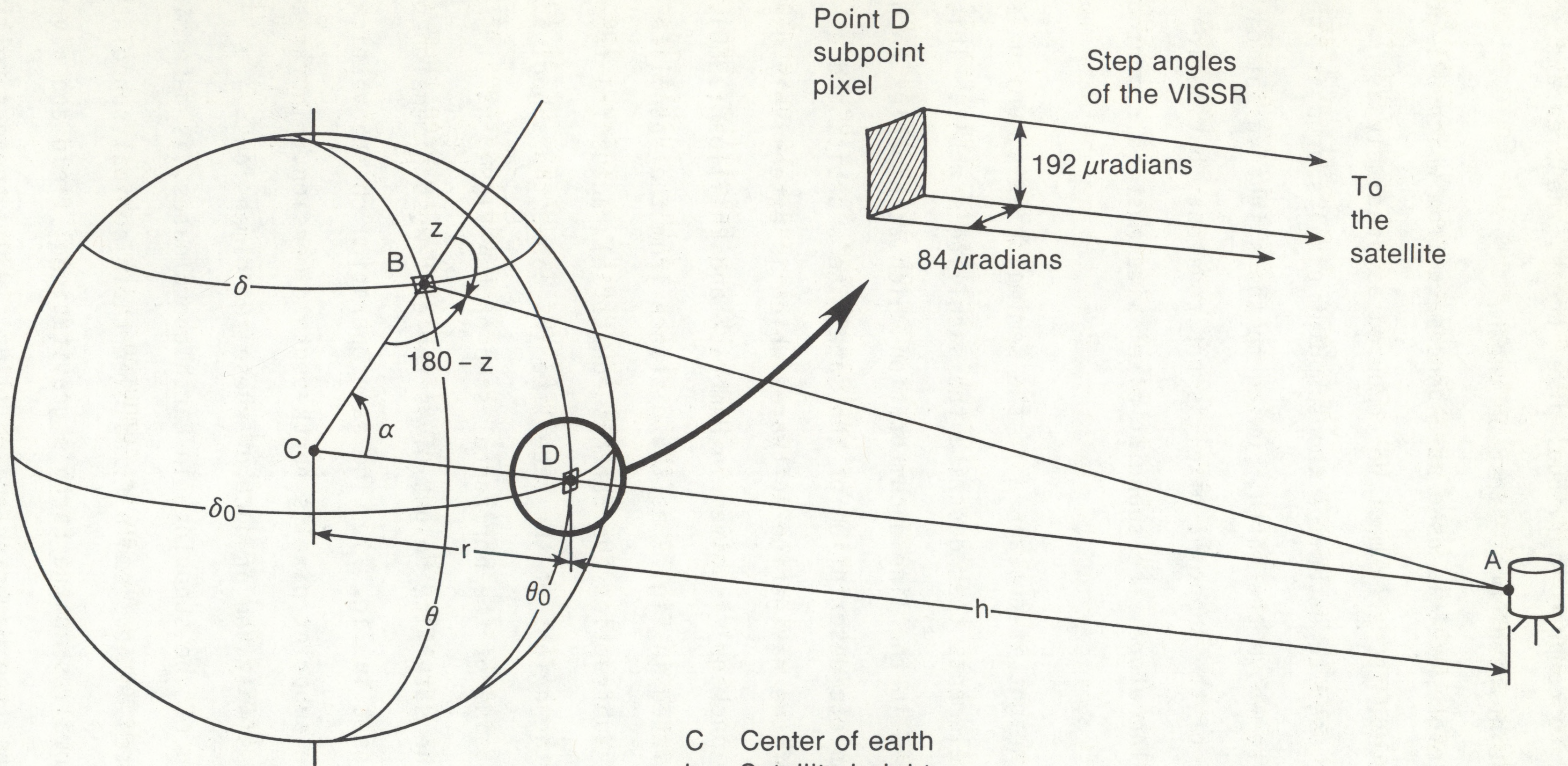
a. Cloud top area

In subroutine DOCUMENT, up to five areal measurements are made for a particular cloud top, the -52°C area, and the areas within the -52°C envelope as cold as, or colder than -58°C, -64°C, -70°C, and -76°C; these latter areas may or may not be contiguous. These measurements are made by pixel summation. Accurate cloud top area measurements by this method require that the areas of individual pixels that make up the cloud top be known or computed precisely. Unfortunately, pixels on a GOES image are not uniform in size. Rather, they increase in area radially from the satellite subpoint. This variation is a consequence of the "satellite projection" that results from the satellite's scanning technique, in which fixed VISSR step angles are used in scanning the spherical Earth. The resultant projection of the Earth on a full disk image can be described as a partial sphere (centered at the satellite subpoint) projected onto a tangent plane, which makes contact with the Earth only at the satellite subpoint. Thus the subpoint pixel area is the standard, and the transformation function describing the satellite projection represents the normalizing factor from which the areas of all other pixels can be computed. Therefore, to compute individual pixel areas, the area and location of the satellite subpoint pixel must first be known.

Because the subpoint pixel is used as a benchmark from which all other pixel areas are computed, its area must be computed as precisely as possible. One factor used to determine the subpoint pixel's area is the height of the satellite above the Earth. Owing to the slight variation in this height, the size of the subpoint pixel is not constant, but varies slightly with time. Therefore, the subpoint pixel size must be computed for each image processed.

The satellite's height is contained in the semimajor axis, which is one of the six Keplerian elements that describe the satellite's orbit. This information is supplied in the line documentation which accompanies each digital image. A complete description of the satellite's orbit, and a detailed discussion of the navigation software (which is extensively used in the MCS documentation package) is given in Hambrick and Phillips (1980). The semimajor axis is depicted in Fig. 7 as the distance from the satellite to the center of the Earth. Subtracting the Earth's equatorial radius (r) from the semimajor axis gives the height of the satellite (h) above the subpoint. Multiplication of the sine of the horizontal step of the VISSR (84 microradians) times the satellite height gives the east-west dimension of the subpoint pixel (see inset in Fig. 7). The same operation on the vertical step of the VISSR gives the subpoint pixel's north-south dimension. The area of the subpoint pixel is simply the product of these two dimensions.

Next the location of the subpoint, in Earth coordinates, is determined. Geostationary satellites do not remain fixed over their nominal subpoints. Rather, because of anomalies in the Earth's gravitational field and a variety of other considerations (see Hambrick and Phillips, 1980), these satellites



C Center of earth
h Satellite height
r Earth radius
D Satellite subpoint
δ_0 Subpoint latitude
θ_0 Subpoint longitude
δ Point B latitude
θ Point B longitude
z Point B zenith angle
α Angle BCD

Figure 7. Geometric configuration used to determine the area correction factor for the pixel at point B. Points A, D, and C represent the positions of the satellite, subpoint pixel, and center of the Earth, respectively.

wobble on their spin axes and also drift. This movement results in a variable subpoint location, which traces a figure-8 pattern about the Equator over a 24-hour period. The amplitude of the figure 8 ($\sim \pm 2°$ lat.) is significant and must also be considered when cloud top area and other measurements are obtained from digital satellite imagery. Therefore, the location (latitude and longitude) of the satellite subpoint (δ_o, θ_o) must also be determined for each image processed. In the MCS documentation software, the location of the subpoint is found simply by calling the navigation routines for the center point of the full-disk scan, which is always represented by the same line and element, regardless of the satellite.

When the subpoint pixel's area and location are known, the area of any other pixel on the image may be computed. The area of an individual pixel is a function of its zenith angle, which is defined as the angle between that pixel's zenith and the satellite (z in Fig. 7). This angle is determined by solving triangle ABC in Fig. 7, which is formed by the satellite, the pixel whose area is being measured, and the center of the Earth. Initially, only the length of side AC, the semimajor axis of the orbit, is known. The first step is to determine the latitude (δ) and longitude (θ) of the pixel (point B in Fig. 7) by calling the navigation routines. After the locations of both the pixel and subpoint are defined, the angular distance between them (α) may be computed:

$$\alpha = \arccos[\cos(\theta-\theta_o)\ \cos(\delta-\delta_o)]\ . \qquad (1)$$

When α is known, the distance $\overline{AB}$ can be computed:

$$\overline{AB} = \sqrt{r^2 + (r+h)^2 - 2r(r+h)\cos\alpha} \; . \qquad (2)$$

Because sin z = sin (180° - z), the law of sines can be used to solve for the zenith angle of the pixel:

$$z = \arcsin \frac{(r + h)\sin\alpha}{\overline{AB}} \; . \qquad (3)$$

Multiplication of the secant of the zenith angle of a particular pixel times the subpoint pixel area gives the area of that pixel.

Application of this procedure to all pixels as cold as, or colder than, -52°C in the sector would be computationally time consuming. To avoid this, a pixel area correction look-up table is created prior to any cloud top area calculations. This is the function of subroutine AREA_COR_TABLE. The product of this routine is the array COR(256,250) which has a one-to-one correspondence with the data sector (ISEC). Every pixel in ISEC has a corresponding area correction factor in COR. Therefore, multiplication of the subpoint pixel area times COR(I,J) gives the area of the pixel at ISEC (I,J).

In producing the area correction table, the navigation routines are not called for every pixel in ISEC. Rather, computing time is saved by navigating every 10th element of every 10th row of ISEC. Linear interpolation is then applied to compute the Earth coordinates for the pixels in between the navigated points. With those Earth coordinates, zenith angles of all pixels in the subsector are easily computed using Eqs. (1), (2) and (3).

A sensitivity test was carried out to determine the effect of errors introduced by this navigation/linear interpolation scheme on the area calculation. The area correction table was produced under two conditions, one in which the navigation/interpolation scheme described above was applied, and the other in which all pixels in the data array were navigated individually. The latter method produced precise area correction factors for all pixels in the ISEC array and thus gave true cloud-top areas, but required 25 times more computing time than the former method. Computed cloud-top areas resulting from the use of these two methods differed by only 0.03%, thus justifying the use of the navigation/interpolation scheme in the creation of the area correction table.

Cloud-top areas encompassed by the five threshold temperatures are computed from the IVALUE and IANDJ arrays that hold the digital count values and locations of all pixels making up the cloud top. In order to simplify the processing of these arrays the IVALUE array is sorted from high to low digital count, and the IANDJ array is rearranged accordingly. These sorted arrays for the schematic cloud top in Fig. 6a are also shown in Fig. 6b. Because -52°C is the cloud definition threshold, the -52°C area is computed simply by summing the individual areas of all pixels represented in the IVALUE and IANDJ arrays. Area correction factors for pixels in the IVALUE array are indexed in the area correction table by the coordinates held in IANDJ. The -58°C area is computed by considering only those pixels in IVALUE whose digital count value exceeds 202. In the example in Fig. 6b, only the first four elements of the sorted IVALUE and IANDJ arrays are considered for the -58°C area. This procedure goes on until all five threshold areas have been measured.

b. Centroids

Area-weighted centroids of the five threshold cloud top areas are also computed from the sorted IVALUE and IANDJ arrays. Centroid locations are computed from the x,y coordinates of the pixels stored in the IANDJ array;

information in the IVALUE array is used only to delineate the start points for each of the five areas. The x coordinate of the -52°C centroid is the median of all x values in the IANDJ array, and its y coordinate is the median of all y values. Coordinates of the -58°C area centroid are the medians of the x and y values of all pixels whose digital count value exceeds 202. In Fig. 6b, only the first four elements of the sorted IANDJ array would be considered in computing the -58°C area's centroid. Centroids for the colder threshold areas are computed in the same manner.

In computing the x and y medians for a particular cloud top area, the x and y values are removed from IANDJ array and stored in two independent arrays. These arrays are then sorted from high to low value; the medians are simply the midpoints of the sorted arrays. The unsorted and sorted x and y arrays for both the -52°C and -58°C areas of the schematic cloud top in Fig. 6a are given in Figs. 8a and 8b, respectively. In that example, the -58°C area's centroid is at location (5.5, 5), and the -52°C area's centroid is at (5, 4). For the sake of simplicity, depictions of the -64°C, -70°C areas' centroids are not shown.

c. Eccentricity

As given in the MCC definition on page 3, the difference between a linear convective system (squall line) and a circular convective system (MCC), as viewed from space, is defined in terms of eccentricity. The last function of subroutine DOCUMENT is to compute the eccentricity of the -52°C envelope. As defined by Maddox, the eccentricity of a convective system is the ratio of the minor axis length (b) to the major axis length (a) [it should be noted that this differs from the classic definition of eccentricity, which is

UNSORTED POSITION ARRAYS

-58°C AREA X	-58°C AREA Y	-52°C AREA X	-52°C AREA Y
6	5	6	5
5	5	5	5
4	5	4	5
7	5	7	5
		5	3
		4	2
		5	4
		5	6
		3	2
		4	3
		5	2
		6	3
		6	4
		6	6
		5	7

(a)

SORTED POSITION ARRAYS

-58°C AREA X	-58°C AREA Y	-52°C AREA X	-52°C AREA Y
7	5	7	7
→ 6	5	6	6
5	5	6	6
4	5	6	5
		6	5
		5	5
		5	5
		→ 5	4
		5	4
		5	3
		5	3
		4	3
		4	2
		4	2
		3	2

(b)

Figure 8. (a) Position arrays describing x and y coordinates for pixels belonging to the schematic cloud top in Fig. 6; (b) Position arrays sorted from high to low value. Arrows on the sorted arrays point to the midpoints or medians of those arrays. (For simplicity, only the position arrays for the -52°C and -58°C areas are presented.)

$(\sqrt{a^2 - b^2}/a)$]. Subjective depictions of a and b are shown for a typical MCC in Fig. 9. If this ratio exceeds .7, the storm is considered circular; otherwise, it is considered linear.

Measurement of this ratio directly from GOES hardcopy imagery is difficult because of problems introduced by the satellite projection. Therefore, before computation of the cloud top eccentricity, the satellite imagery must be transformed to a standard projection. Only then can distances be computed properly. Manually, this involves physically transferring the perimeter of the -52°C area from the hardcopy image to a base map, determining a best-fit ellipse, and measuring its major and minor axes in standard units.

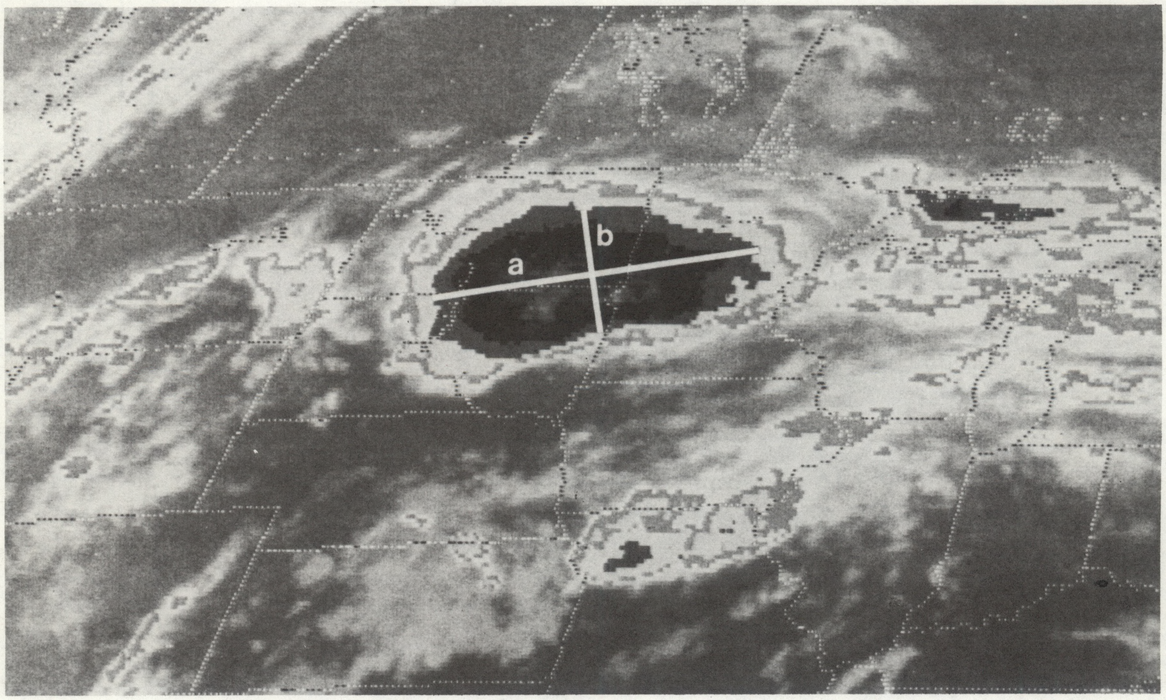


Figure 9. Subjectively determined major (a) and minor (b) axes for the MCC's -52°C area.

This procedure has been automated and is carried out in subroutine ELLIPSE, which is called from subroutine DOCUMENT.

The method adapted is one developed by Östlund (1974) and expanded by Zittel (1976) for automated radar echo tracking. It is described in APPENDIX B. According to this method, harmonic analysis is applied to a succession of x,y locations defining the outer edge of an entity, and the first harmonic describes the best-fit ellipse to that entity. Both Zittel and Östlund operated in a Cartesian coordinate system; however, in the present application, the ellipse is fit on a spherical surface.

The first step in the automated procedure is to describe the path around the -52°C envelope. This is accomplished by the specialized contouring subroutine PATHLENGTH which defines the outer edge of the -52°C area in quasi-regularly spaced points. The way in which the locations of these points are determined is given in APPENDIX A. Navigation software is used to transform the coordinates of each point of the perimeter to Earth coordinates (latitude and longitude), so that distances between successive points may be computed accurately. The distance between two points is computed on a tangent plane true at the midpoint between them using Eq. (4):

$$\Delta s_i = \sqrt{[(x_i - x_{i-1}) \cos(\delta)]^2 + [y_i - y_{i-1}]^2} \qquad (4)$$

where δ is the mean latitude between y_i and y_{i-1}. The $\cos(\delta)$ term is required for this application because the locations of these points are not defined in a Cartesian coordinate system, but in the Earth coordinate system; i.e., the absolute length of 1 degree of longitude varies as the cosine of the latitude. Also, owing to the way in which points describing the pathlength are determined, Δs is not constant, but somewhat variable. Because harmonic analysis assumes that Δs is constant, errors are introduced into the results. However, these errors should affect higher order harmonics more than the first harmonic. Three arrays describe the perimeter: (1) a latitude array YS (whose contents are referred to as y), (2) a longitude array XS (whose contents are referred to as x) and (3) an array S, which describes the path length cumulatively in kilometers from the beginning to end. Table 1 lists these arrays for the example -52°C cloud top in Fig. 10a.

Fig. 10b shows a plot of the x coordinate of points defining the -52°C perimeter versus the cumulative path length s. Fig. 10c shows the same for the y coordinates. These plots are roughly sinusoidal and thus well approximated by the fundamental frequency (first harmonic) of the harmonic analysis. Combined, the expressions describing the first harmonic of the x and y versus s plots correspond to the best-fit ellipse of the -52°C envelope; the mean x and y values (the zeroth harmonic) represent the center of the best-fit ellipse. The mathematics of this technique is given in APPENDIX B. The end result is an equation of the form

$$\frac{x'^2}{1/A'} + \frac{y'^2}{1/C'} = 1 \, . \qquad (5)$$

It describes the best-fit ellipse in a rotated (x',y') reference frame, which is tilted through Θ degrees with respect to the original (x,y) coordinate system (see Fig. 11). Also, because the mean x and y values are subtracted out, the ellipse represented by the first harmonic has been translated to the origin. In this case it was translated to the center of the Earth coordinate system (0°, 0°). A' and C' are constants related to Θ and other constants

Table 1. -- Perimeter coordinates for example in Fig. 10a

x (longitude)	y (latitude)	s (km)
-99.00	36.00	00.0
-98.75	37.00	114.5
-98.20	38.00	241.3
-98.00	38.20	272.4
-97.00	38.60	392.5
-96.00	39.00	512.6
-95.00	39.10	623.8
-94.20	39.00	713.8
-93.25	38.00	867.3
-93.20	37.00	978.5
-93.25	36.40	1045.2
-93.90	36.00	1130.8
-94.40	35.50	1208.6
-95.00	35.40	1276.5
-96.00	35.25	1388.8
-97.00	35.50	1503.3
-98.00	35.50	1614.5
-99.00	36.00	1739.0

derived in the ellipse-fitting mathematics (see Eqs. B34 and B35). By definition, the major and minor axes of the ellipse represented by Eq. (5) are aligned along the x' and y' axes, respectively, and the center is at (0',0'), which is collocated with the origin of the non-rotated coordinate system (see Fig. 11). According to Eq. (5), the ellipse axes' lengths in the rotated (primed) reference frame are $2\sqrt{1/A'}$ and $2\sqrt{1/C'}$.

In a Cartesian coordinate system these axis lengths would be sufficient. However, because the ellipse was fit in the Earth coordinate system, where the unit length in the x direction is a function of y (latitude), the ellipse must be translated from the origin to its proper location on the surface of the Earth before its axes' lengths are measured. Instead of translating the entire ellipse, only the axes' endpoints are moved,

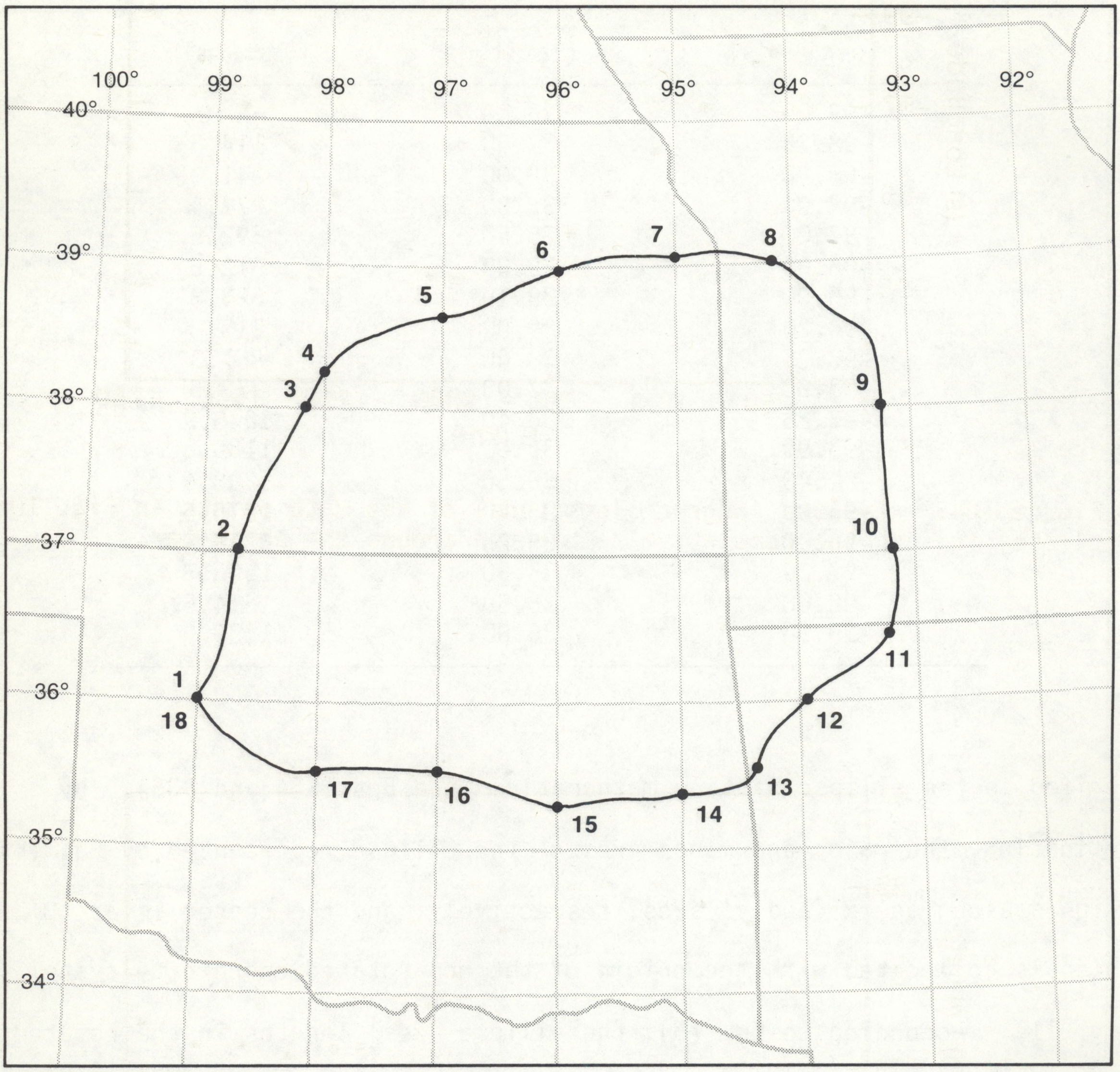


Figure 10a. Example of the -52°C envelope of an MCC. The points along the perimeter represent locations of the data points from which the best-fit ellipse is determined.

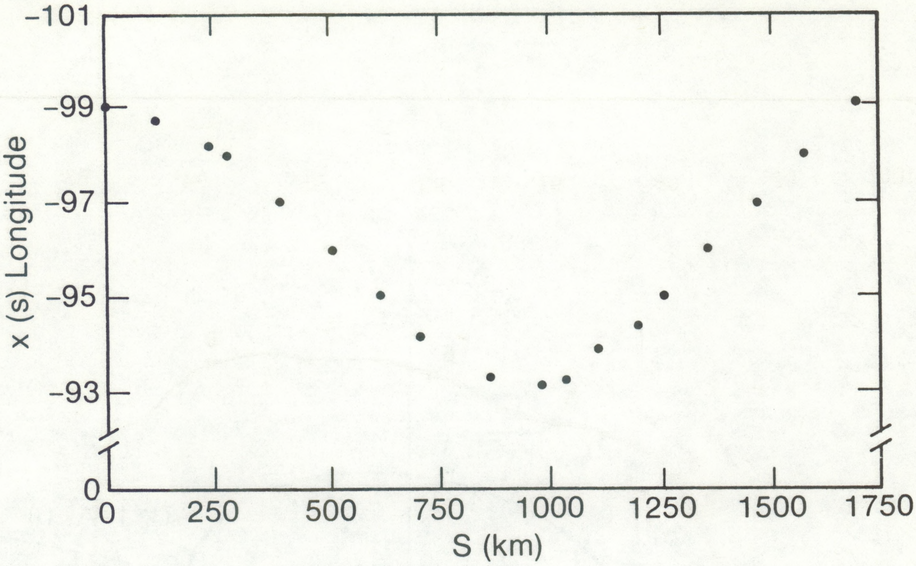


Figure 10b. x values (degrees longitude) of the data points in Fig. 10a plotted against the cumulative pathlength around the perimeter.

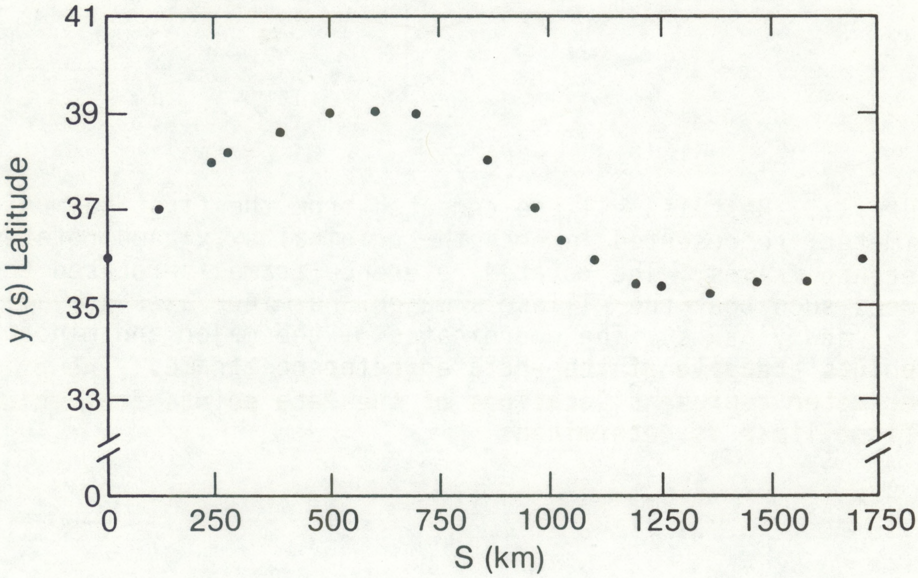


Figure 10c. y values (degrees latitude) of the data points in Fig. 10a plotted against the cumulative pathlength around the perimeter.

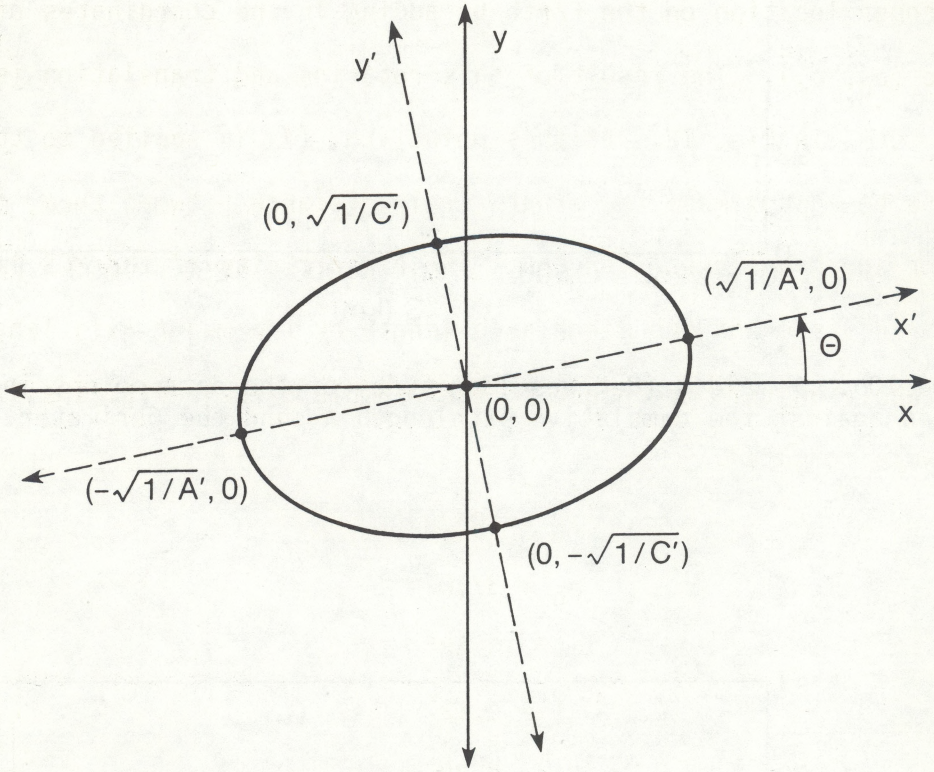


Figure 11. Best-fit ellipse computed from the first harmonic of the perimeter, represented in both the original (x,y) and rotated (x',y') reference frames. The rotated reference frame is rotated through Θ degrees such that the ellipse's major and minor axes are aligned along the x' and y' axes. The coordinates of the major and minor axes endpoints are given in the rotated reference frame.

which are computed by alternately setting x' and y' in Eq. (5) equal to 0 (see Fig. 11). The transfer of these points to their proper location on the Earth is a two-step process. First they are rotated through Θ degrees into the original non-rotated coordinate system. The endpoints are then translated to their proper location on the Earth by adding in the coordinates of the zeroth harmonic (a_0, c_0). The result of this rotation and translation is given, in general form, in Fig. 12. At this point, Eq. (4) is applied to these endpoints to approximate the great-circle distance between them, thus giving the major and minor axes' lengths. The eccentricity of the MCS is computed simply by division of the minor axis length by the major axis length.

In applying this technique to the example data in Fig. 10a, the following values were computed:

$$a_0 = -95.97°$$

$$c_0 = 37.07°$$

$$\Theta = 10.48°$$

$$A' = 0.12$$

$$C' = 0.28$$

The ellipse that they represent is shown in Fig. 13 overlaid on the original perimeter data from which it was computed. From these values, coordinates of the endpoints of the major and minor axes of the best-fit ellipse were computed and are listed in Table 2. Applying Eq. (4) to these coordinates gives a major axis length of 4.63° latitude (515.35 km), and a minor axis length of 3.74° latitude (415.74 km). The eccentricity is .81, which qualifies as a circular convective storm according to the Maddox (1980) classification.

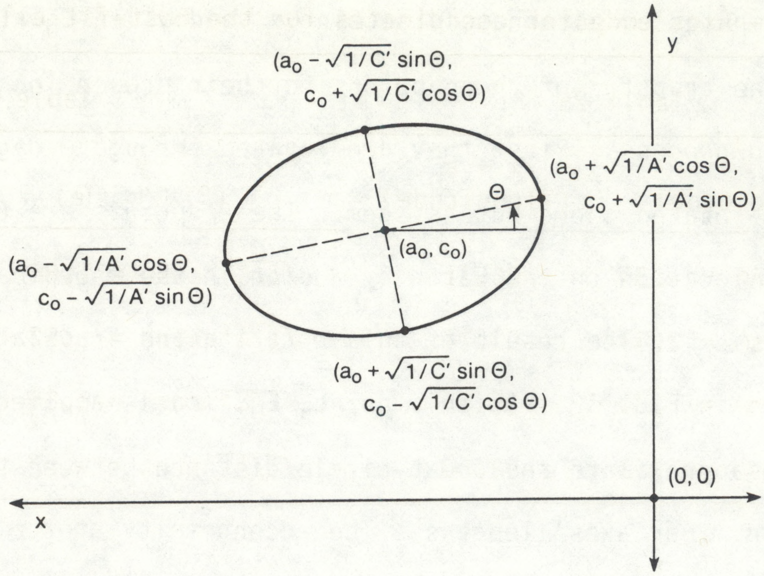


Figure 12. The best-fit ellipse in the x,y reference frame. The ellipse's major and minor axes' endpoints are labeled in (x,y) coordinates, and the center point has been translated to its true location (a_o,c_o).

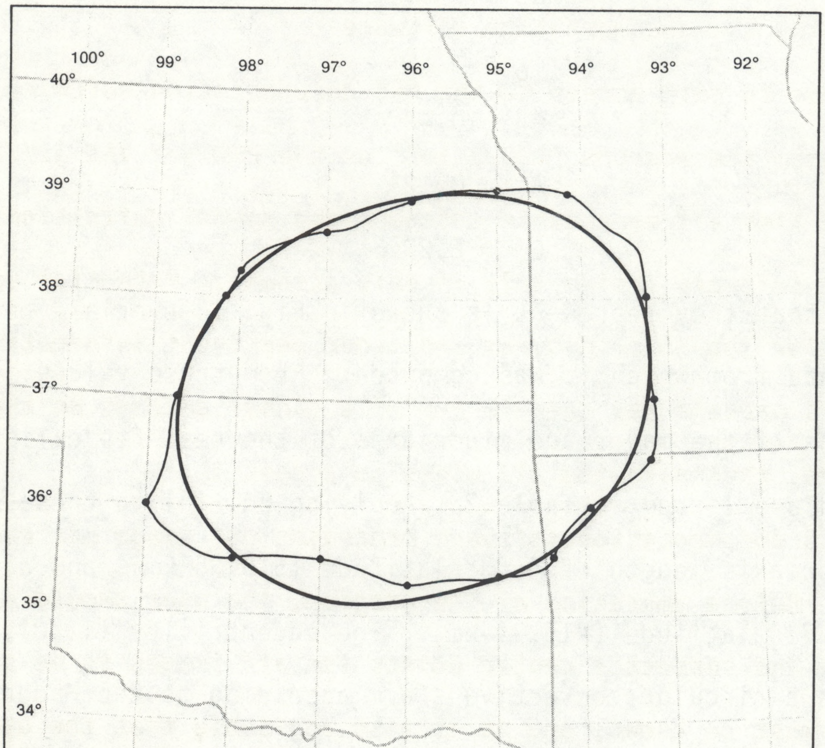


Figure 13. Best-fit ellipse for the data in Fig. 10a, overlaid on the original data.

Table 2. -- Axes endpoint coordinates for the best-fit ellipse in Fig. 13

Table 2a		Table 2b	
x (°longitude)	$x + a_o$ (°longitude)	y (°latitude)	$y + c_o$ (°latitude)
$-\sqrt{1/C'}\sin\Theta = -.34$	-96.31	$\sqrt{1/C'}\cos\Theta = 1.85$	38.92
$\sqrt{1/A'}\cos\Theta = 2.83$	-93.14	$\sqrt{1/A'}\sin\Theta = .52$	37.59
$\sqrt{1/C'}\sin\Theta = .34$	-95.63	$-\sqrt{1/C'}\cos\Theta = -1.85$	35.22
$-\sqrt{1/A'}\cos\Theta = -2.83$	-98.80	$-\sqrt{1/A'}\sin\Theta = -.52$	36.55

IV. OPERATION AND PRODUCTS

A. Operating the documentation software

In order to gain access to the MCS documentation software, the user must be logged onto the account "AUGUSTINE" on the PROFS-2 VAX 11/780 computer. At the present time all executable software and symbol definitions that access and run the software reside only in this account. Potential users must see the author for log-in procedures and other pertinent information. In the future, this package may operate out of a global account on the PROFS network of computers, thus allowing all users to access it.

The MCS documentation software processes one image per run, and is user activated. Before operation of the program, the user should compile a list of times (GMT) and subsector center points for all images to be processed. The Julian day must be known, and it is also advisable that the user check the SATDAT directory on the data device (logically associated with the name DATA_DEV) to ensure that all image files to be processed are available. A listing of all appropriate data files in SATDAT (those with .ILE or .ILW extensions) may be displayed by issuance of the following symbol:

SATDATA

The response is a listing on the user's CRT as shown in the example given in Table 3. The image file-naming convention at PROFS is YYDDDHHMM.ext, where YY represents the two least significant digits of the year (e.g., 85 for 1985), DDD represents the Julian day, HHMM is the time of the image in GMT, and the extension (.ext) identifies the sector. (The ";1" appended to each file name is that file's version number which is unimportant.) The extensions that identify the GOES-East and GOES-West infrared large files are .ILE and .ILW, respectively.

Table 3. -- Example response to the symbol "SATDATA"

843071730.ILE;1	843071800.ILE;1	843071830.ILE;1	843071900.ILE;1
843071930.ILE;1	843072000.ILE;1	843072030.ILE;1	843072100.ILE;1
843072130.ILE;1	843072200.ILE;1	843072230.ILE;1	843072300.ILE;1
843072330.ILE;1			

After the user has determined that the image files to be processed do exist, the documentation software may be initiated by issuing the following symbol:

MCC

The user is then prompted to enter information which describes the data file and geographic area to be processed. First, the user identifies the desired satellite, by response to the following request:

ENTER E FOR GOES-EAST OR W FOR GOES WEST

Any response other than E or W (upper or lower case) is unrecognized and the user is requested to re-enter the satellite identifier. (Currently, GOES-West resides at 98°W, but is still identified by "W".) Next, the user specifies the date and time of the image to be processed by responding to this next request:

ENTER THE DATE/TIME OF THE DESIRED IMAGE (YYDDDHHMM)

The required format of the user's response is shown in parentheses, which is in the same format as that for the file names shown in Table 3. If the requested image does not reside in SATDAT, the image file representing the next closest time is accessed. If a substitute image is connected, the user is informed and given options to terminate or continue by the following message:

```
--WARNING-- REQUESTED IMAGE NOT AVAILABLE,
            IMAGE REPRESENTING YYDDDHHMM
            WAS SUBSTITUTED.

  DO YOU WANT TO CONTINUE?
  (Y OR N)
```

The process is terminated if a negative response is entered. Once an image is accepted, the date and time of the accessed image is immediately displayed on the user's CRT in conventional notation, MM/DD/YY HHMM (GMT).

The third and final user input is the center point of the desired subsector. The user is prompted by the following command:

```
ENTER THE LATITUDE AND LONGITUDE OF THE DESIRED SUBSECTOR'S CENTER
(E.G., 40.5, 90.0)
```

The expected order of entry is latitude, then longitude, separated by a comma or blank. Because these are read under FORTRAN's free format, decimal points are optional if whole degrees are entered. No sign convention for these Earth coordinates is used because the software assumes that all events to be documented are located in the Northern and Western hemispheres. This assumption is harmless because the PROFS infrared large sector lies entirely in those hemispheres. From this point, no further user input is required and the program goes on to process the requested image. If it completes without problems, the following message appears on the CRT:

```
RUN COMPLETED, RESULTS ON YYDDDHHMM.DAT
READY FOR PRINTING
```

where "YYDDDHHMM" represents the date and time of the processed image.

The output file, identified by the extension ".DAT", contains a tabulation of all documentation calculations for each storm processed. Each time the program is executed, an output file is created with a unique file name (YYDDDHHMM.DAT). These files accumulate in the user's directory and may be printed at any time. To print a single file, the user issues the following command:

```
PRINT YYDDDHHMM.DAT
```

If the user wishes to preview this output file on the CRT before printing it, the following command may be entered:

```
TYPE YYDDDHHMM.DAT
```

After a series of images has been processed, the user may print all accumulated output files by issuing the following command:

```
PRINT YY*.DAT
```

The result is that all output files are concatenated and printed in chronological order.

If a problem occurs during execution, a message is usually printed and, depending on the severity of the problem, the program may or may not continue. Program termination is identified by the return of the operating system prompt ($) in the left column of the CRT screen. For certain situations, messages giving the source of the problem are printed on both the CRT and output file. For example, if the navigation information is not recoverable from the satellite line documentation, the following message will appear on the CRT screen:

```
PROBLEMS WITH GETSATDDC_SA, STOP EXECUTION
FORTRAN STOP
$
```

In this case, execution is terminated because many of the pending calculations are dependent on the accurate navigation of the satellite imagery. Many other such anticipated errors are recognized by this software; however, at times the program may fail on some unforeseen condition. In such cases, a small memory dump appears on the CRT informing the user where in the software the error occurred. Again, depending on the severity of the problem, the program may or may not terminate. When it does terminate prematurely, it is suggested that the user attempt to rerun the same data file. If the program again fails, one could safely assume that the requested image cannot be analyzed, and if possible, another image should be processed in its place. Finally, if, for some reason, the user wishes to terminate execution of the program, simultaneously depressing the "ctrl" and "y" keys aborts the program during any phase of operation.

B. Products

Two products are generated: an output file containing all pertinent documentation information, and a color-coded video display of the processed subsector with a geographical map overlay which appears on a RAMTEK monitor. The color code of the video display is given in Table 4. Example products for the image in Fig. 14a are shown in Figs. 14b and 14c.

Table 4. -- Color code for video display

Temperature Range	Color
>-32°C	White
-32°C to -51°C	Blue
-52°C to -57°C	Green
-58°C to -63°C	Light Green
-64°C to -69°C	Yellow
-70°C to -75°C	Orange
≤-76°C	Red

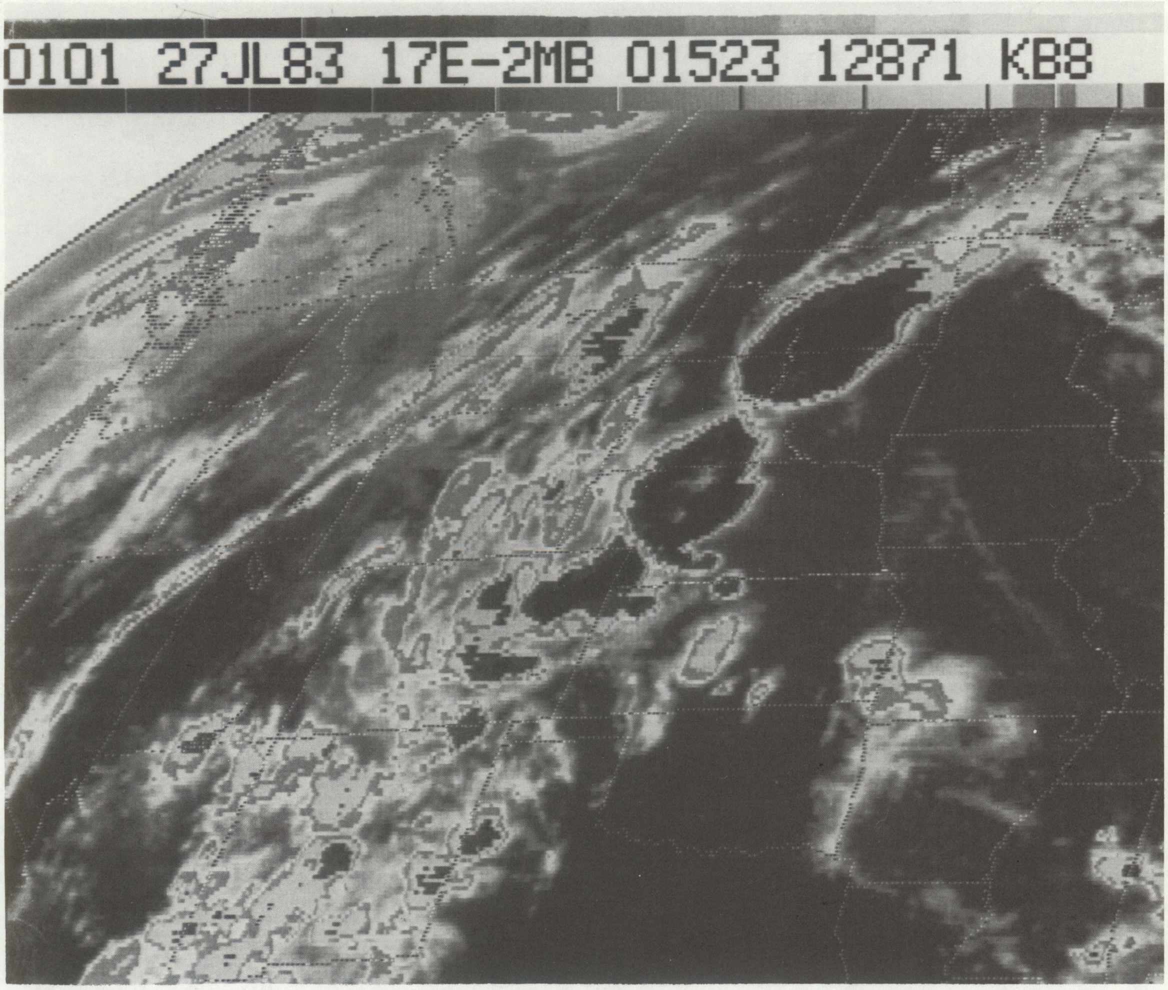


Figure 14a. GOES-East infrared image for 0100 GMT on July 27, 1983.

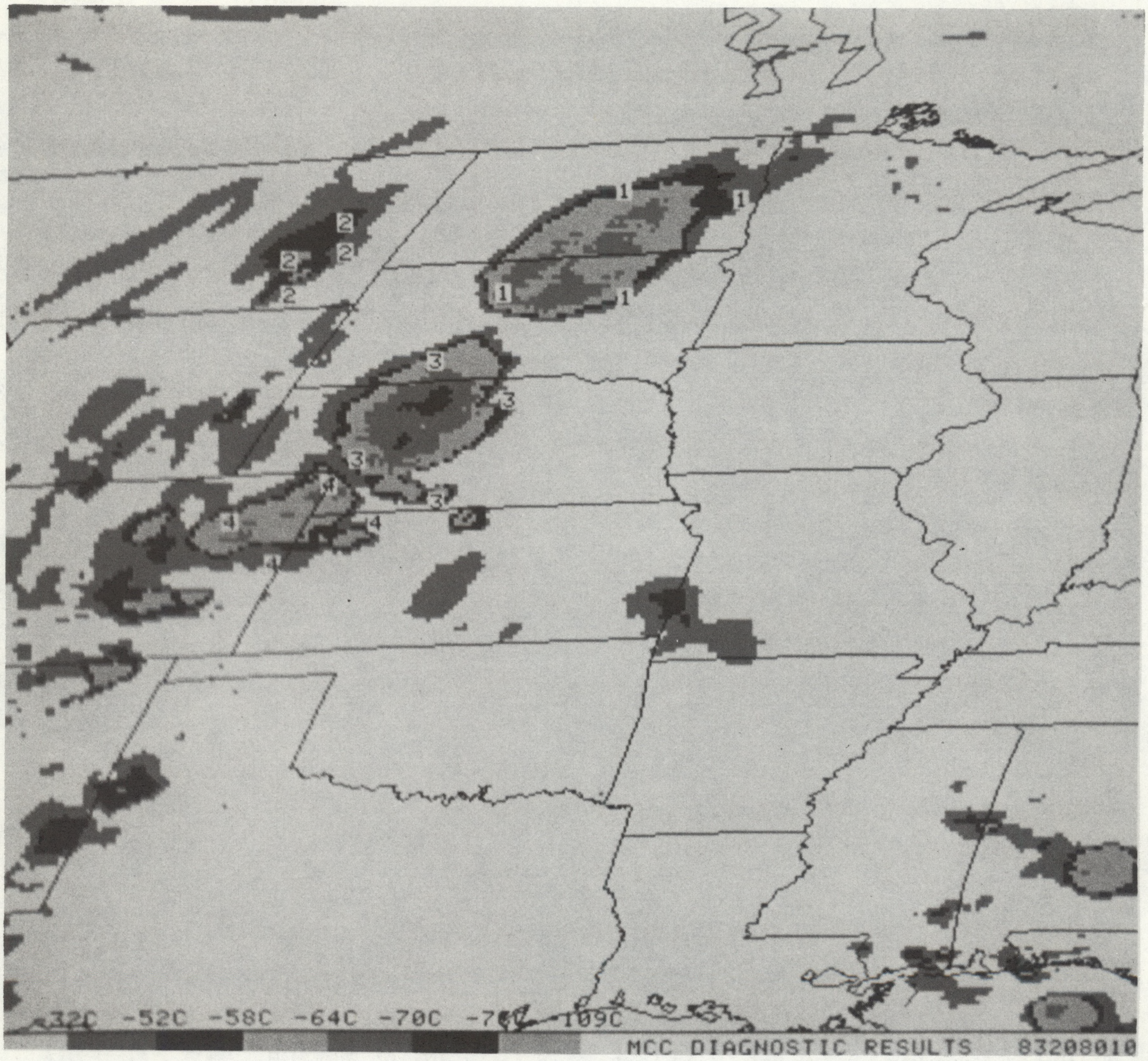


Figure 14b. Video display product of the processed 0100 GMT image in Fig. 14a.

07/27/83 STORM 1 DESCRIPTION 0111 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	1076	0.811320E+05 KM**2	46.21	-99.58
-58C	862	0.648598E+05 KM**2	46.11	-99.72
-64C	292	0.218452E+05 KM**2	45.70	-99.96

THE ECCENTRICITY OF THE -52C AREA IS 0.891

07/27/83 STORM 2 DESCRIPTION 0111 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	181	0.150449E+05 KM**2	46.57	-106.14
-58C	17	0.139527E+04 KM**2	46.24	-106.10

THE ECCENTRICITY OF THE -52C AREA IS 0.455

07/27/83 STORM 3 DESCRIPTION 0111 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	883	0.619409E+05 KM**2	42.19	-101.08
-58C	747	0.523894E+05 KM**2	42.19	-101.08
-64C	358	0.251141E+05 KM**2	42.19	-101.17
-70C	46	0.323556E+04 KM**2	42.48	-101.05

THE ECCENTRICITY OF THE -52C AREA IS 0.588

07/27/83 STORM 4 DESCRIPTION 0111 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	467	0.318848E+05 KM**2	39.87	-102.26
-58C	346	0.236606E+05 KM**2	39.98	-102.49
-64C	40	0.274331E+04 KM**2	40.05	-102.18

THE ECCENTRICITY OF THE -52C AREA IS 0.791

Figure 14c. Output file 832080100.DAT for labeled storms in Fig. 14b.

Table 5. -- Gray scale for video display

Temperature Range	Gray shade
>-32°C	White
-32°C to -51°C	Medium gray
-52°C to -57°C	Black
-58°C to -63°C	Light gray
-64°C to -69°C	Medium gray
-70°C to -75°C	Black
≤-76°C	Light gray

For the RAMTEK display in Fig. 14b, a gray scale is substituted for the color code given in Table 4 because publication in color is not feasible. This substitute gray scale is presented in Table 5. In the output file (Fig. 14c), storm descriptions are numbered and listed in the order that they were located and processed. Each storm description gives (1) the date and time of the image, (2) the storm number, (3) the number of pixels, cloud top area (in square kilometers) and area weighted centroids for the -52°C and colder threshold areas, and (4) the eccentricity of the -52°C envelope. The corresponding color-coded video display, which appears automatically on a RAMTEK monitor, shows the subsector that was processed; cloud top areas are highlighted according to the temperature ranges listed in Table 4. Although the -32° area is not numerically analyzed, it is color coded in blue (medium gray in Fig. 14b) because of consideration given it in the original MCC documentation by Maddox (1980). Another feature of this display is highlighted endpoints of the major and minor axes defining the best-fit ellipses to the -52°C envelopes of various storms in the sector. The two sets of axis endpoints for each storm are labeled by "storm number." Labels of "1" at the four axes endpoints of a particular storm associates it with the first storm processed, which is identified in the output file by the heading:

STORM 1 DESCRIPTION.

Endpoint labels of "2" represent the second storm processed and is identified with the output file entry:

STORM 2 DESCRIPTION,

and so on. For any given image, the upper limit on the number of storms that may be processed is 20. Axis labeling serves two purposes, the most significant of which is to illustrate the credibility of the eccentricity calculation. If the endpoints identify an ellipse that approximates the -52°C (green) area well, then the eccentricity value in the output file may be used with confidence; if not, the eccentricity may need a subjective adjustment. Of secondary importance are the numerical values of the axis labels themselves, which allow for quick association of storms on the video display with their corresponding documentation statistics in the output file.

Figures 15 - 23 show the images and products that chronologically follow the image in Fig. 14a, for the convective events that occurred over the north-central United States on July 27, 1983. The format is the same as that in Fig. 14; the first part (a) shows the hard copy image from which MCS's were initially identified, the second part (b) shows the video display (in a gray scale format), and the last part (c) shows the corresponding output file associated with the video display.

In the first processed image of this sequence (Fig. 14a) four storms were documented, the first in the Dakotas, the second in eastern Montana, the third in western Nebraska and the fourth in northeast Colorado. By virtue of the fact that they were documented, these were the only storms in the sector whose -52°C cloud shield areas exceeded 10,000 km^2. Although the Dakotas' storm (storm 1 in Figs. 14b and 14c) was the largest, the one in western Nebraska (storm 3 in Figs. 14b and 14c) was the most vigorous with over 3200 km^2 of the cloud shield colder than -70°C.

An hour later at 0200 GMT (Fig. 15a), the -52°C canopy of the Dakotas' storm had expanded by approximately 25%, the eastern Montana storm had begun to decay and the Nebraska and Colorado storms were merged at the -52°C level. Because the algorithm's cloud top isolation scheme links all contiguous pixels as cold as, or colder than -52°C, the merged storms were documented as one. As a result, the areas and centroids computed at various temperature thresholds and eccentricity measurement for the merged storm complex were practically meaningless. Since the algorithm considered the Nebraska-Colorado complex as one storm, it was the largest storm in the sector with a -52°C canopy exceeding 121,000 km^2. Although its eccentricity was computed at .718, which just exceeds the MCC criterion for that quantity, it is clear from Fig. 15b that the shape of the merged complex was not elliptical, as the major and minor axes endpoints depicted a the best-fit ellipse that did not approximate the storm's -52°C envelope well. This example illustrates the importance of the RAMTEK display. According to the printed output in Fig. 15c, the merged storm complex qualified as an MCC; however, its shape at -52°C on the video display (Fig. 15b) showed a large but disorganized convective complex.

By 0300 GMT (Fig. 16), the small MCS in eastern Montana had decayed, and the Dakota storm and Nebraska complex had merged at the -52°C level. Just as in the previous image, the algorithm documented the merged complex as one very large extensive storm. Between 0300 and 0500 GMT (Fig. 17), the part of the

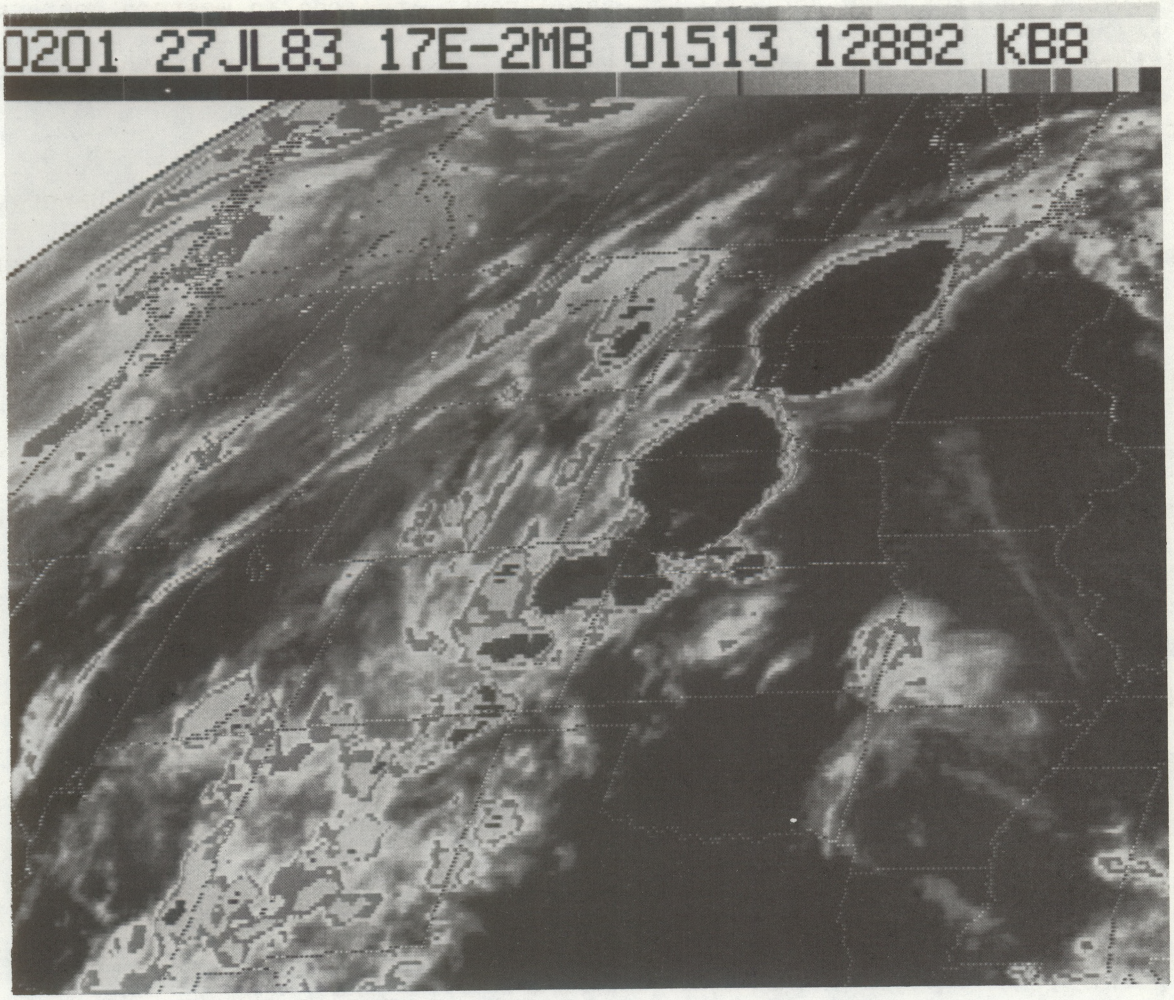


Figure 15a. GOES-East infrared image for 0200 GMT on July 27, 1983.

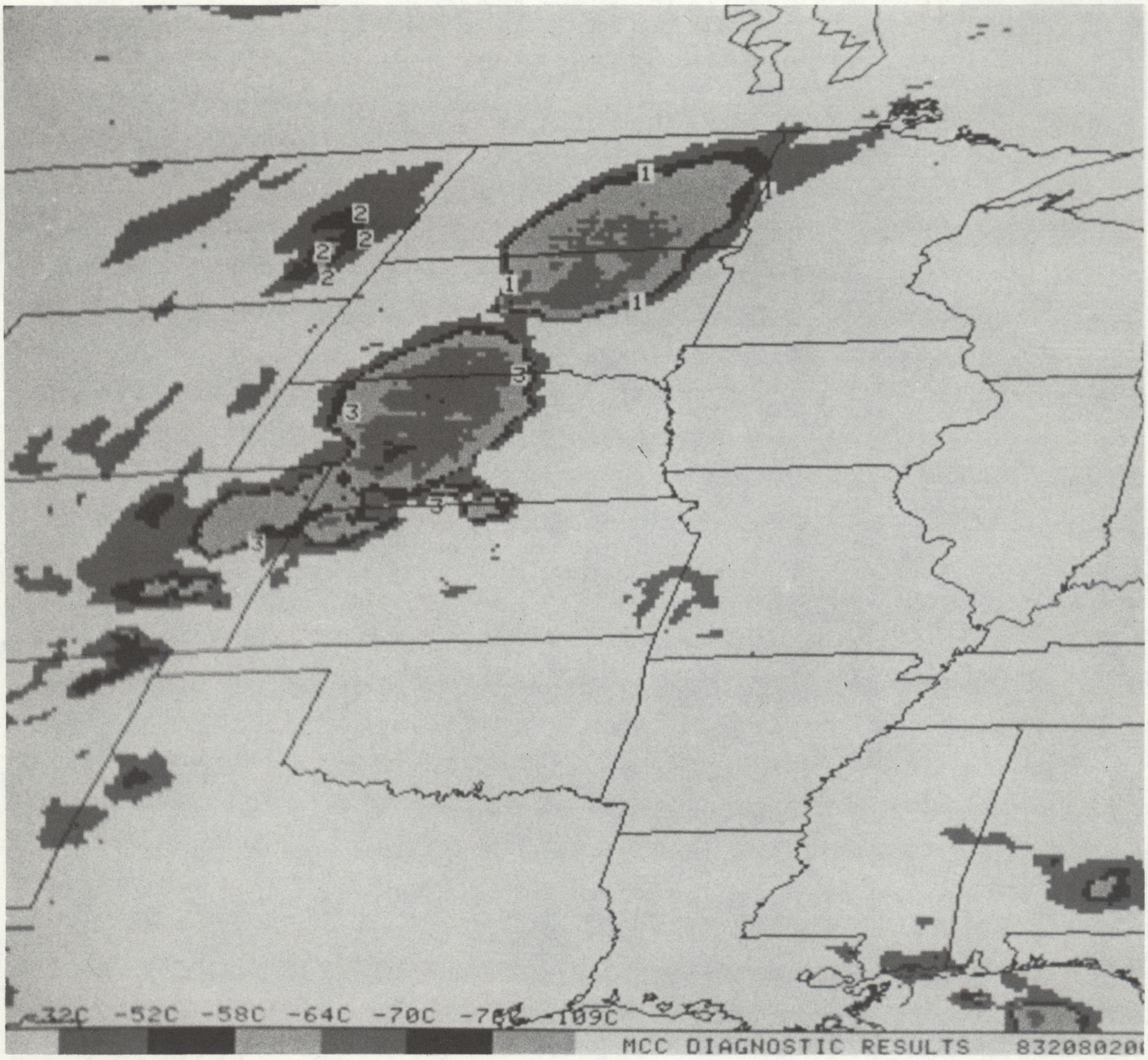


Figure 15b. Video display product of the processed 0200 GMT image in Fig. 15a.

07/27/83	STORM 1 DESCRIPTION			0206 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	1361	0.102406E+06 KM**2	46.41	-99.22
-58C	1152	0.864843E+05 KM**2	46.31	-99.35
-64C	380	0.281728E+05 KM**2	45.77	-99.43
-70C	1	0.736713E+02 KM**2	44.79	-100.66

THE ECCENTRICITY OF THE -52C AREA IS 0.989

07/27/83	STORM 2 DESCRIPTION			0206 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	124	0.101965E+05 KM**2	46.51	-105.47
-58C	19	0.154563E+04 KM**2	46.19	-105.44

THE ECCENTRICITY OF THE -52C AREA IS 0.368

07/27/83	STORM 3 DESCRIPTION			0206 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	1745	0.121073E+06 KM**2	41.70	-101.17
-58C	1426	0.991205E+05 KM**2	41.79	-101.04
-64C	635	0.443357E+05 KM**2	42.27	-100.84
-70C	21	0.144637E+04 KM**2	41.31	-101.06

THE ECCENTRICITY OF THE -52C AREA IS 0.718

Figure 15c. Output file 832080200.DAT for labeled storms in Fig.15b.

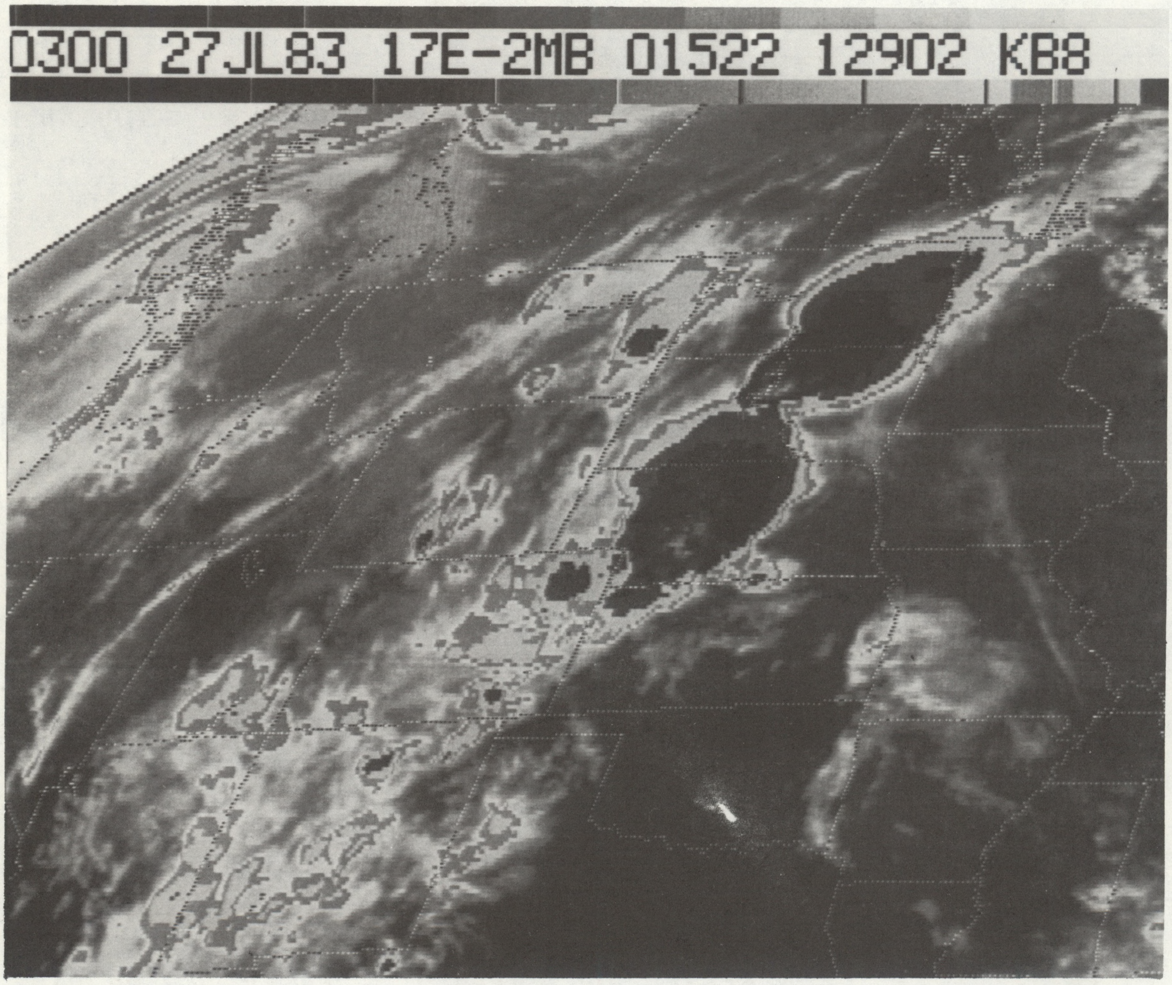


Figure 16a. GOES-East infrared image for 0300 GMT on July 27, 1983. Note that the two storms from 0200 merged at the -52°C threshold in central South Dakota. Because this occurred at the cloud definition threshold, the technique recognizes only one storm.

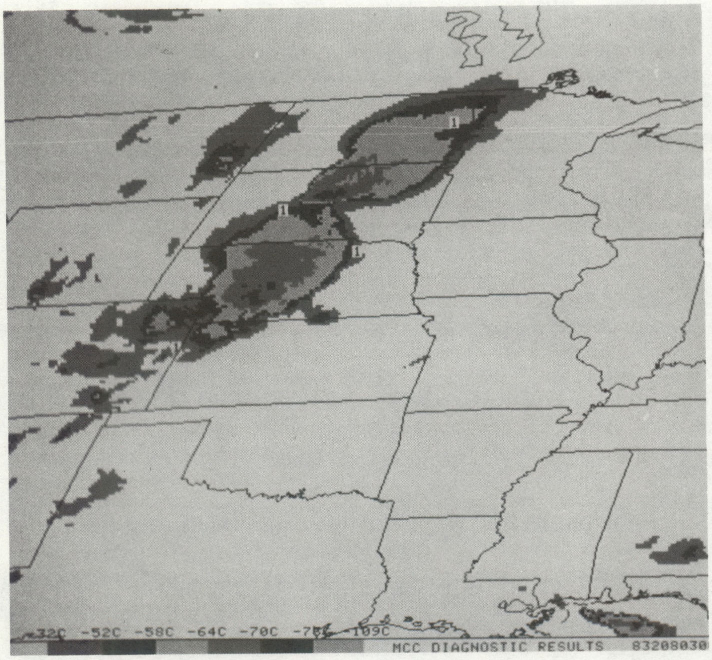


Figure 16b. Video display product of the processed 0300 GMT image in Fig. 16a.

```
THE IMAGE DATE/TIME IS  7/27/83  AT0306 GMT

07/27/83                STORM  1 DESCRIPTION                 0306 GMT

        N. PIXELS            AREA                      CENTROID
                                                   LAT          LONG
-52C      3474        0.250485E+06 KM**2          43.60        -99.91

-58C      2495        0.179574E+06 KM**2          43.39        -99.80

-64C       854        0.604599E+05 KM**2          42.42       -100.38

-70C        67        0.472288E+04 KM**2          41.35       -100.36

THE ECCENTRICITY OF THE -52C AREA IS 0.332
```

Figure 16c. Output file 832080300.DAT for the labeled storm in Fig. 16b.

merged complex in central Nebraska intensified, as evidenced by an increased -70°C area, and its counterpart in the Dakotas weakened. After 0500, the Dakota storm dissipated, and by the next available image at 0730 GMT (Fig. 18a) the Nebraska storm had matured into a well defined MCC. From this image onward, the algorithm worked well because the only storm in the sector was isolated and well defined. Its -52°C and -70°C areas at 0730 were 146,153 km^2 and 14,380 km^2, respectively, and according to the eccentricity calculation, it was nearly circular with a minor to major axis ratio of .973. Unlike previous images, the video display in Fig. 18b confirmed that the best fit ellipse was well representative of the -52°C envelope. An hour later (Fig. 19), the storm's -52°C envelope reached maximum extent, however, it appeared to have lost strength as the -70°C area diminished in size.

The four figures that follow (Figs. 20, 21, 22, and 23), representing 0930, 1000, 1100 and 1200 GMT, show this storm gradually decay as it progresses slowly eastward. Through this 3 1/2 hour period, the storm maintained the area and shape criteria which met the MCC definition, however, its canopy warmed, and its -52°C area became increasingly emaciated.

The presented sequence is a good illustration of the capabilities and deficiencies of this objective MCS documentation algorithm. Figs. 14 and 18-23 are examples of how well the algorithm may operate. When individual mesoscale convective systems are well defined at the cloud definition threshold the program appears to document their cloud top characteristics adequately. On the other hand, Figs. 15-17 show that when storms merge at the cloud definition threshold, the algorithm may give misleading results. For example, the printed output of these times (Figs. 15c, 16c and 17c) gave area and eccentricity measurements which qualified merged complexes as MCC's, however, in reality, they were erratically shaped and disorganized.

It is important that the user recognize situations when the objective MCS documentation algorithm may not produce the desired results. When these situations occur, documentation results for the affected storms should be ignored, and their cloud-top characteristics, if desired, should be measured by manual techniques.

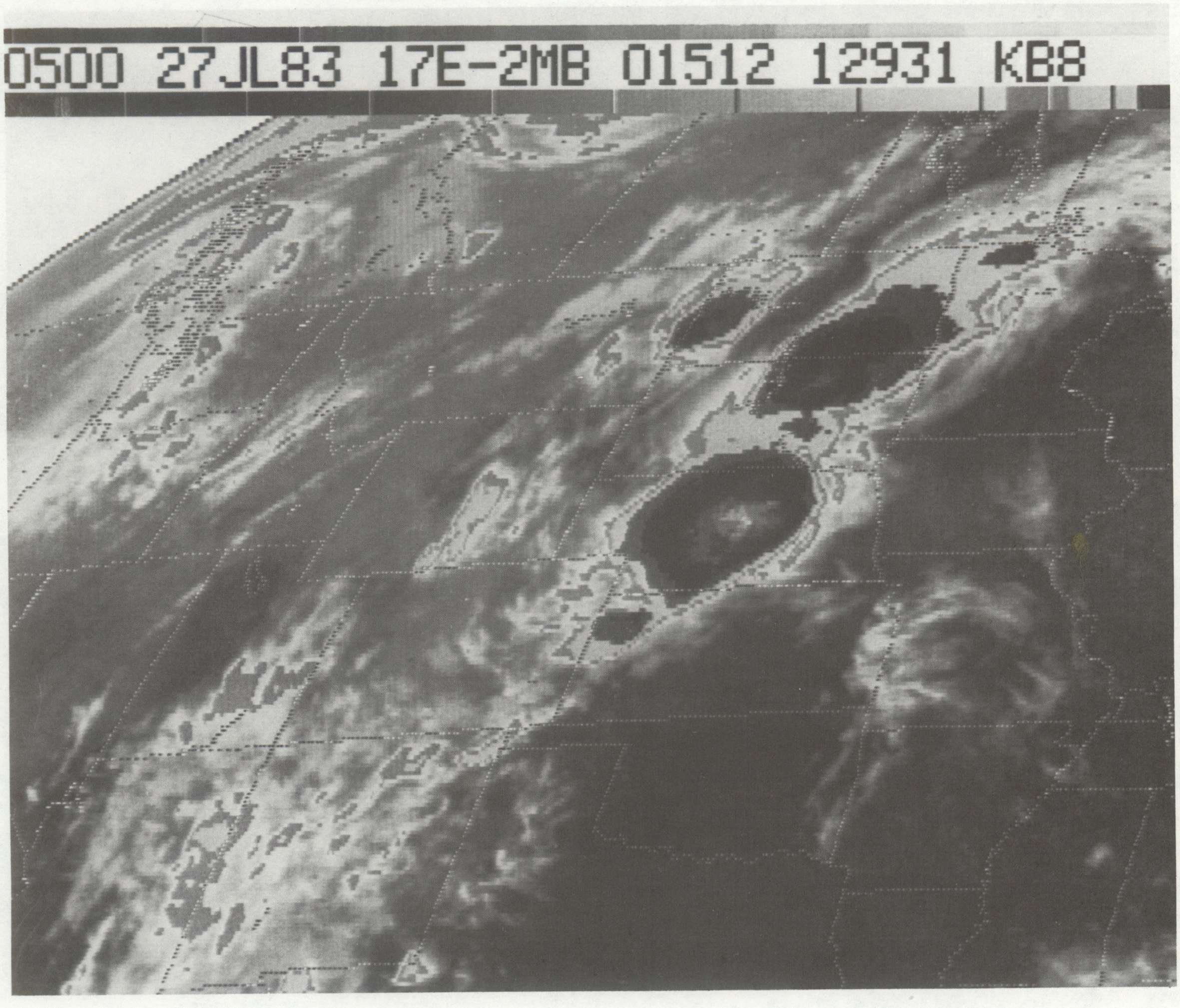


Figure 17a. GOES-East infrared image for 0500 GMT on July 27, 1983.

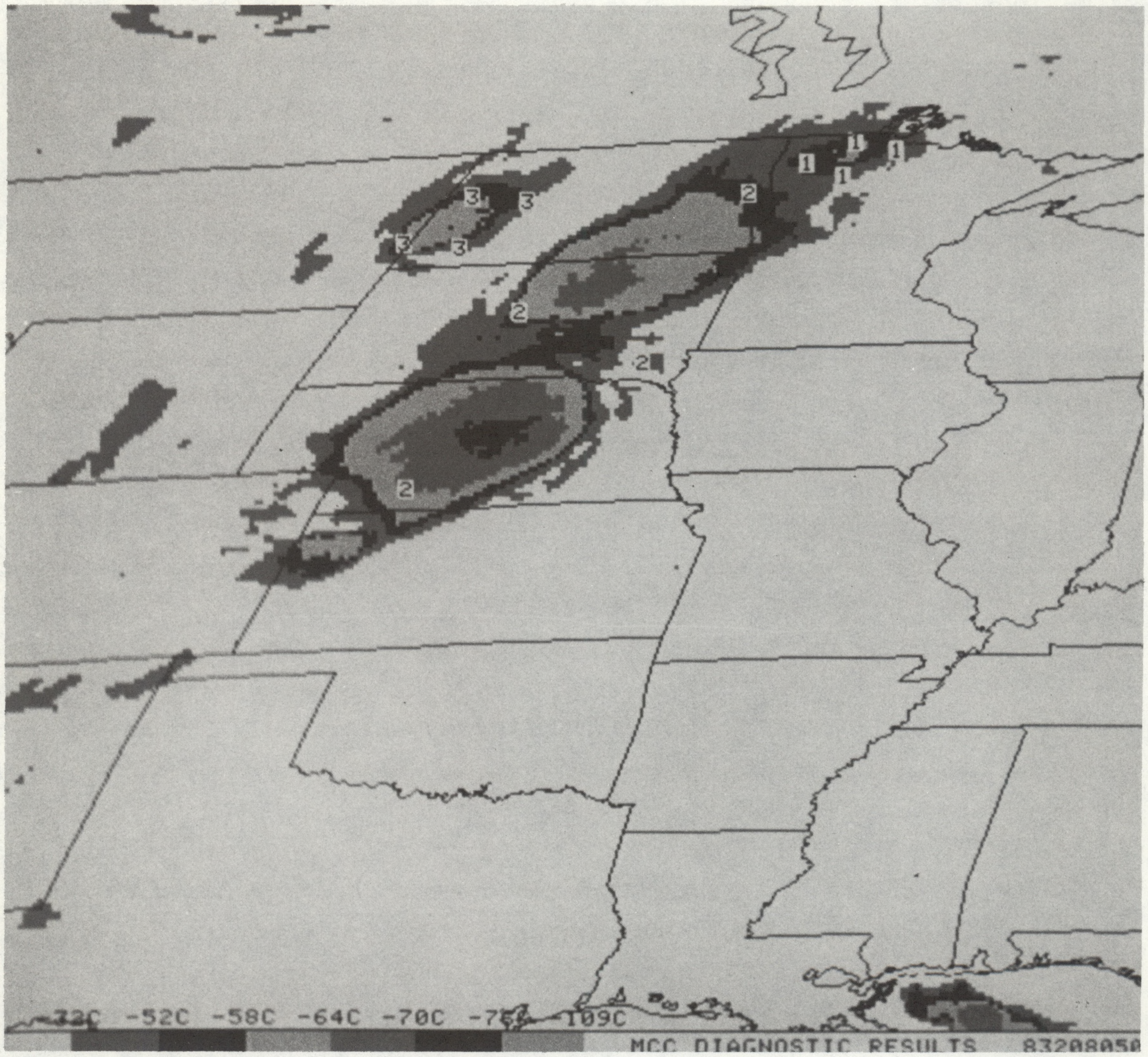


Figure 17b. Video display product of the processed 0500 GMT image in Fig. 17a.

07/27/83 STORM 1 DESCRIPTION 0506 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	135	0.105230E+05 KM**2	48.48	-95.83
-58C	17	0.132326E+04 KM**2	48.59	-95.69

THE ECCENTRICITY OF THE -52C AREA IS 0.776

07/27/83 STORM 2 DESCRIPTION 0506 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	2873	0.207301E+06 KM**2	43.16	-99.20
-58C	2133	0.153275E+06 KM**2	42.66	-99.22
-64C	820	0.576053E+05 KM**2	41.89	-99.63
-70C	98	0.675907E+04 KM**2	41.69	-99.53

THE ECCENTRICITY OF THE -52C AREA IS 0.368

07/27/83 STORM 3 DESCRIPTION 0506 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	245	0.202429E+05 KM**2	47.24	-103.40
-58C	130	0.107184E+05 KM**2	47.15	-103.64

THE ECCENTRICITY OF THE -52C AREA IS 0.918

Figure 17c. Output file 832080500.DAT for labeled storms in Fig. 17b.

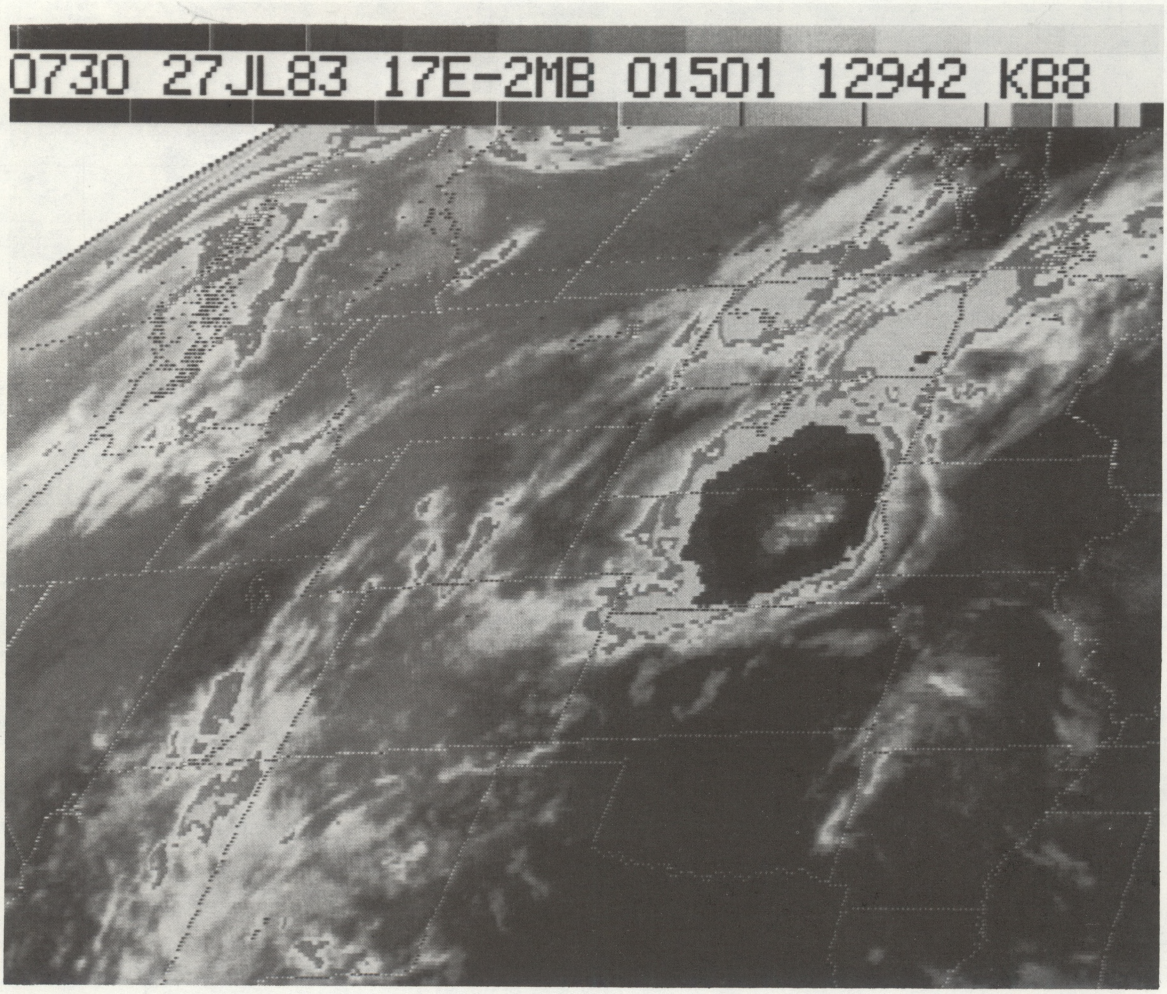


Figure 18a. GOES-East infrared image for 0730 GMT on July 27, 1983.

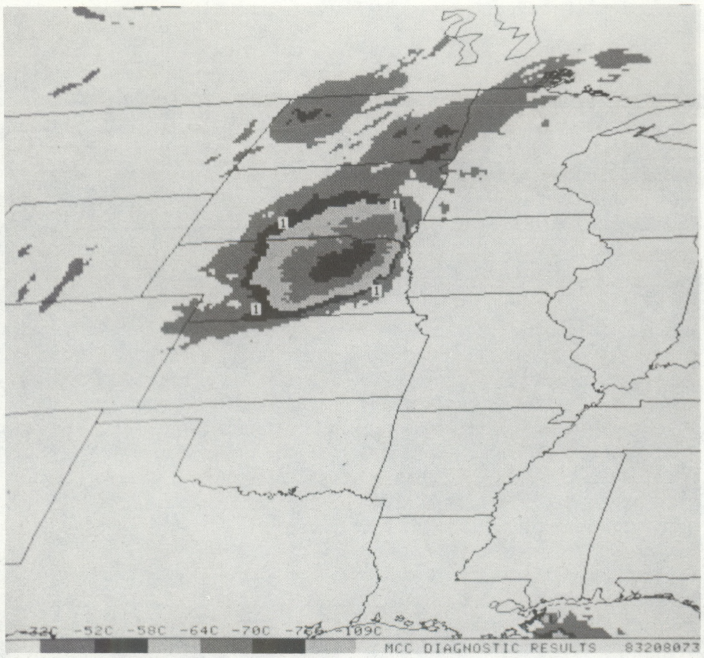


Figure 18b. Video display product of the processed 0730 GMT image in Fig. 18a.

07/27/83 STORM 1 DESCRIPTION 0736 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	2048	0.146153E+06 KM**2	42.34	-99.06
-58C	1553	0.110430E+06 KM**2	42.33	-98.79
-64C	816	0.576598E+05 KM**2	42.22	-98.56
-70C	205	0.143801E+05 KM**2	42.00	-98.29

THE ECCENTRICITY OF THE -52C AREA IS 0.973

Figure 18c. Output file 832080730.DAT for the labeled storm in Fig. 18b.

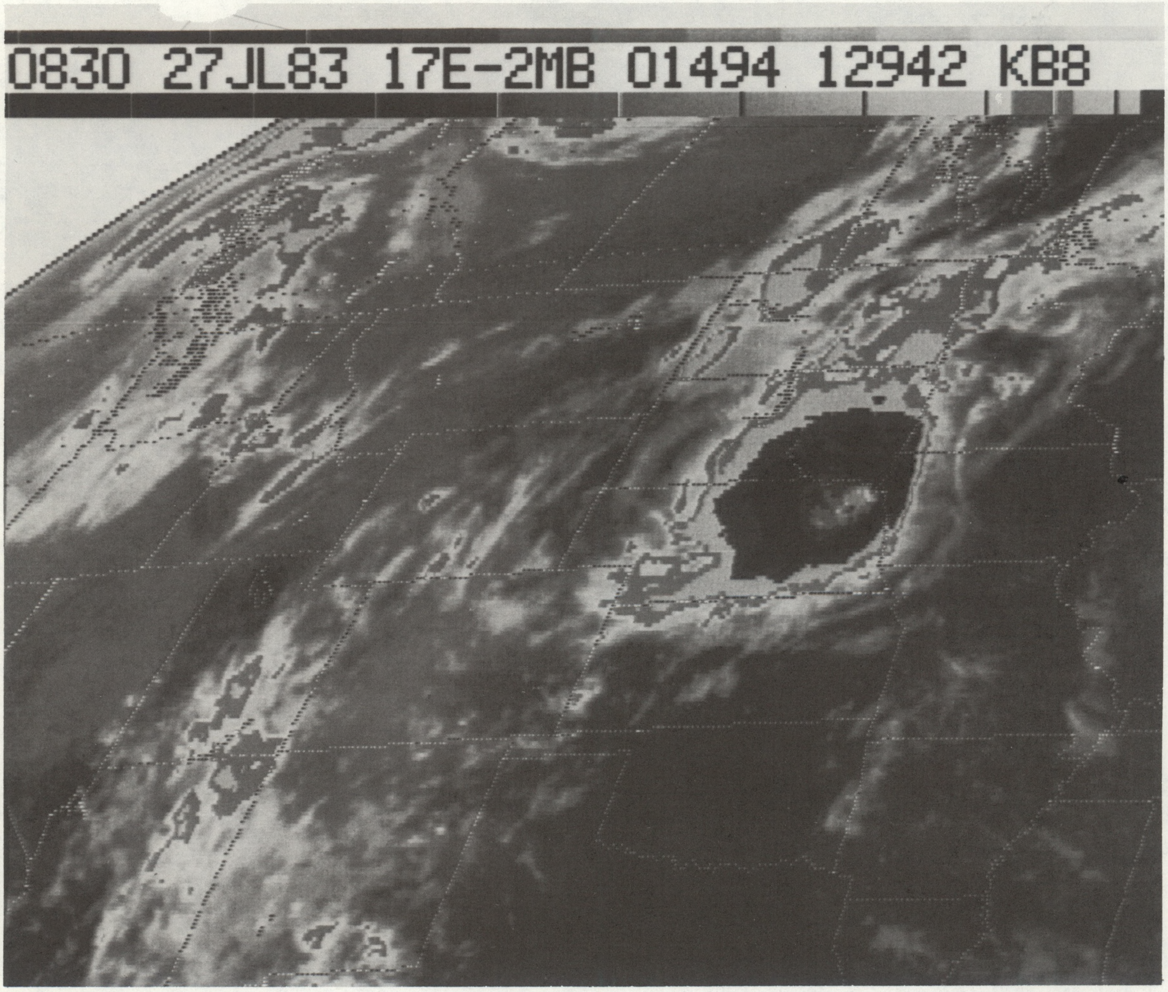


Figure 19a. GOES-East infrared image for 0830 GMT on July 27, 1983.

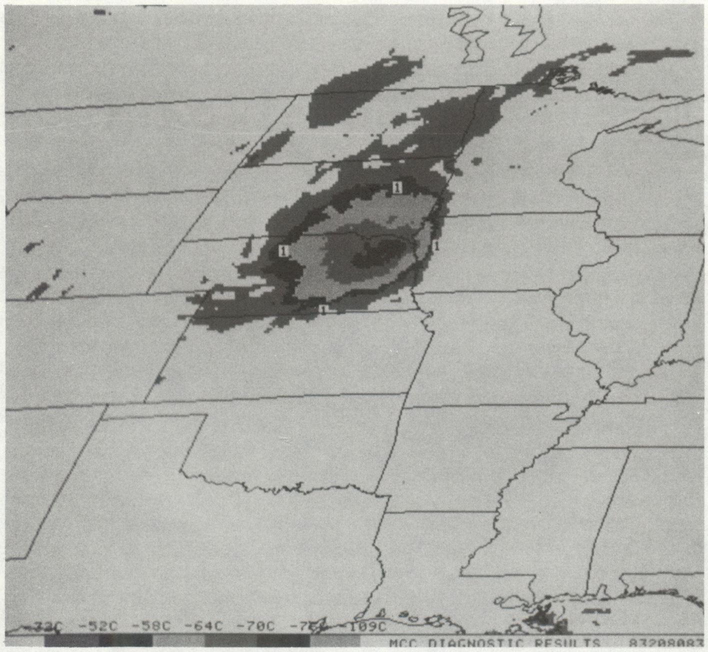


Figure 19b. Video display product of the processed 0830 GMT image in Fig. 19a.

07/27/83 STORM 1 DESCRIPTION 0836 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	2066	0.148617E+06 KM**2	42.59	-98.30
-58C	1494	0.106989E+06 KM**2	42.47	-98.07
-64C	603	0.428433E+05 KM**2	42.37	-97.94
-70C	130	0.915803E+04 KM**2	42.14	-97.40

THE ECCENTRICITY OF THE -52C AREA IS 0.773

Figure 19c. Output file 832080830.DAT for the labeled storm in Fig. 19b.

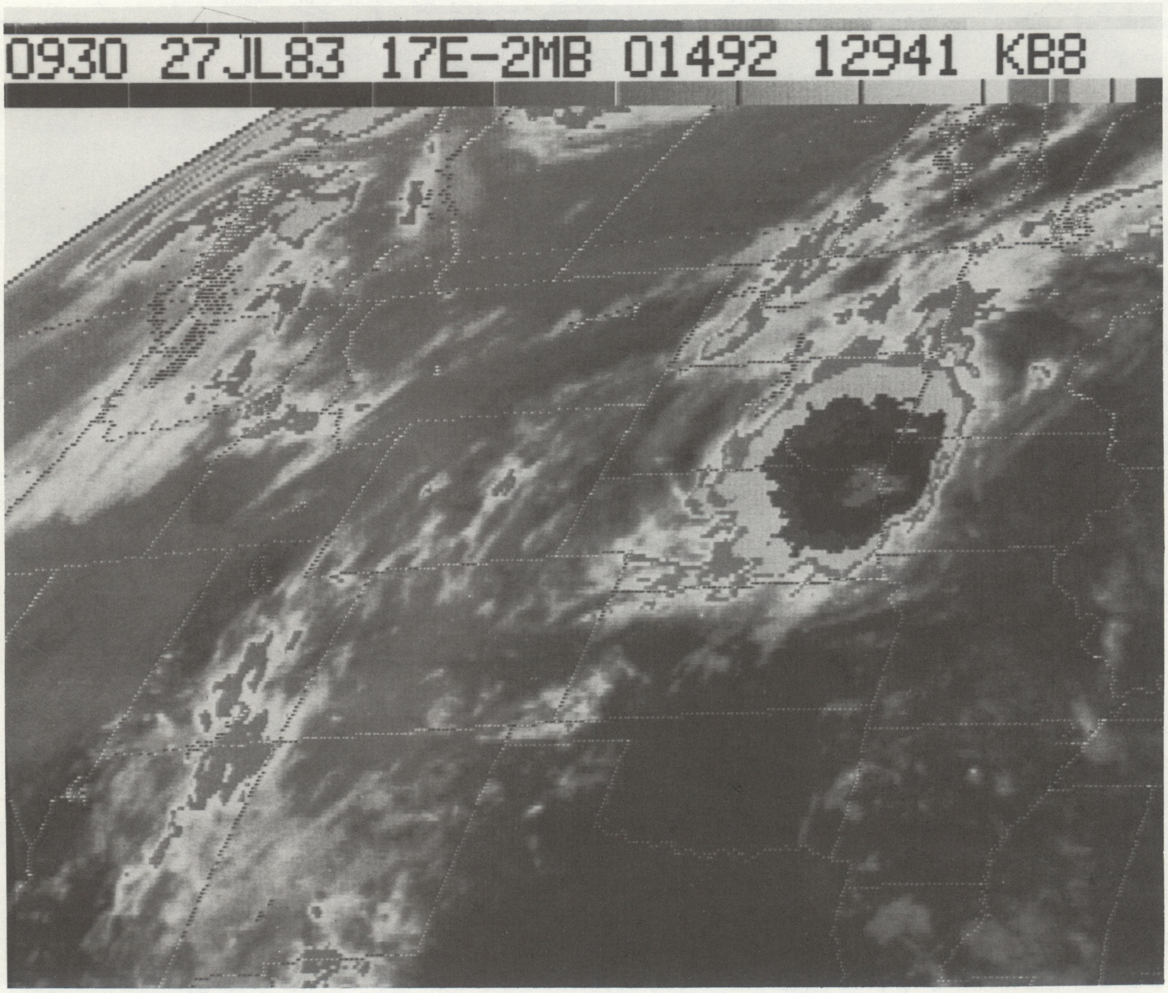


Figure 20a. GOES-East infrared image for 0930 GMT on July 27, 1983.

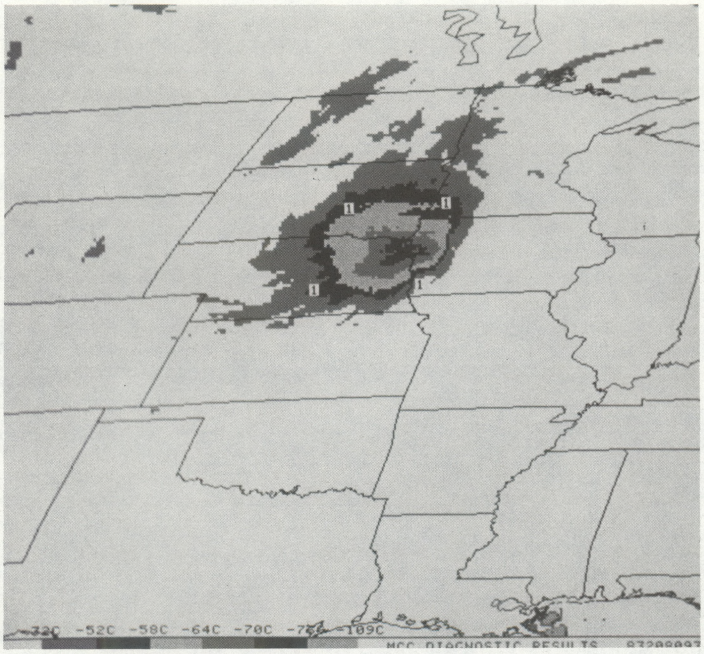


Figure 20b. Video display product of the processed 0930 GMT image in Fig. 20a.

07/27/83 STORM 1 DESCRIPTION 0936 GMT

	N. PIXELS	AREA	CENTROID LAT	CENTROID LONG
-52C	1778	0.128308E+06 KM**2	42.68	-97.45
-58C	1099	0.788179E+05 KM**2	42.58	-97.32
-64C	394	0.279150E+05 KM**2	42.26	-96.92
-70C	93	0.657600E+04 KM**2	42.25	-96.66

THE ECCENTRICITY OF THE -52C AREA IS 0.935

Figure 20c. Output file 832080930.DAT for the labeled storm in Fig. 20b.

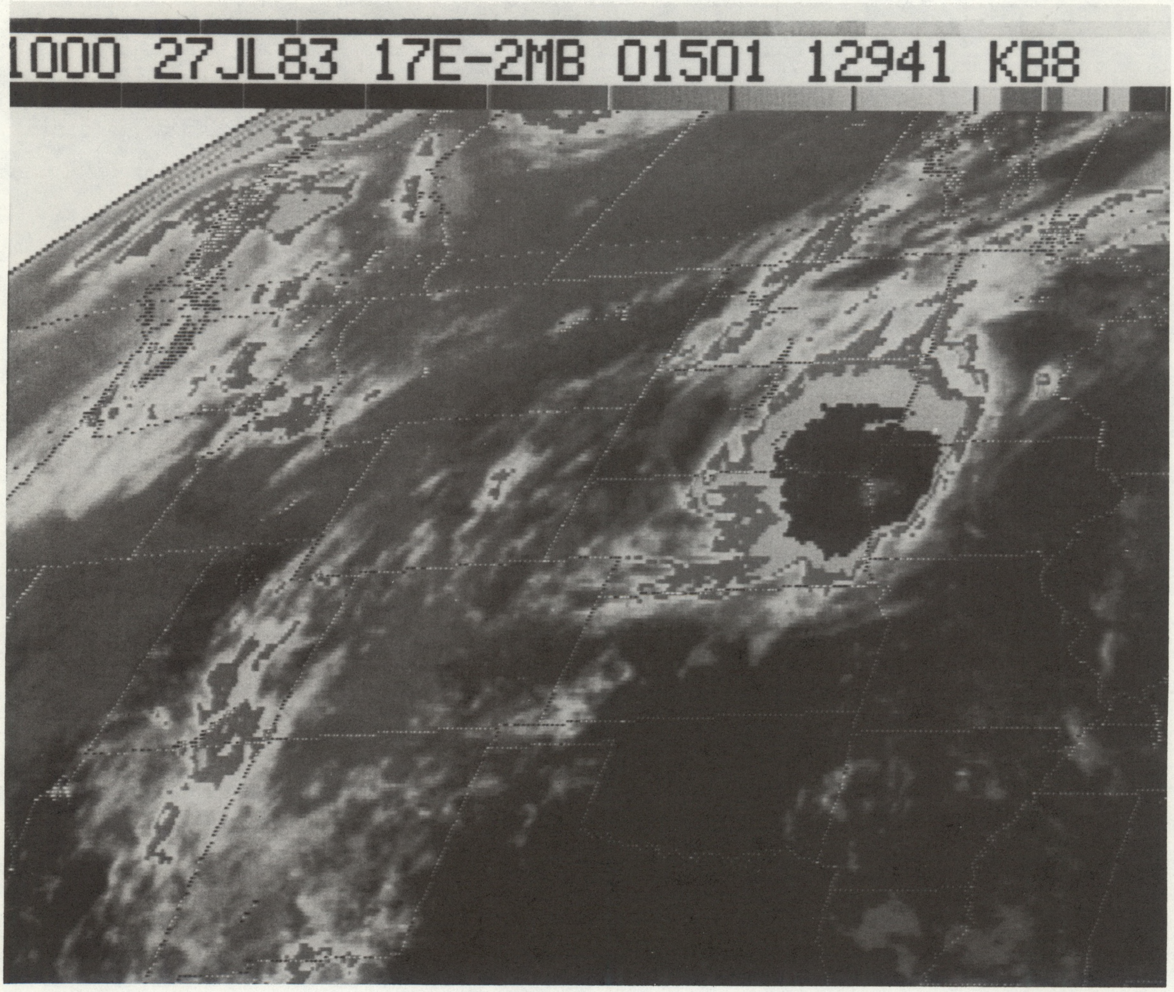


Figure 21a. GOES-East infrared image for 1000 GMT on July 27, 1983.

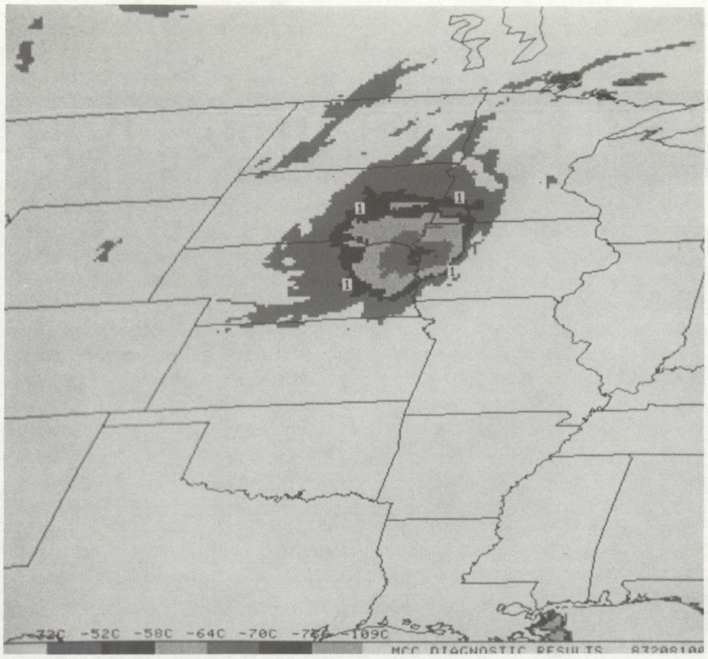


Figure 21b. Video display product of the processed 1000 GMT image in Fig. 21a.

07/27/83		STORM 1 DESCRIPTION		1006 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	1561	0.113407E+06 KM**2	42.93	-97.29
-58C	935	0.673126E+05 KM**2	42.62	-96.97
-64C	315	0.224501E+05 KM**2	42.40	-96.62
-70C	36	0.255710E+04 KM**2	42.39	-96.45

THE ECCENTRICITY OF THE -52C AREA IS 0.892

Figure 21c. Output file 832081000.DAT for the labeled storm in Fig. 21b.

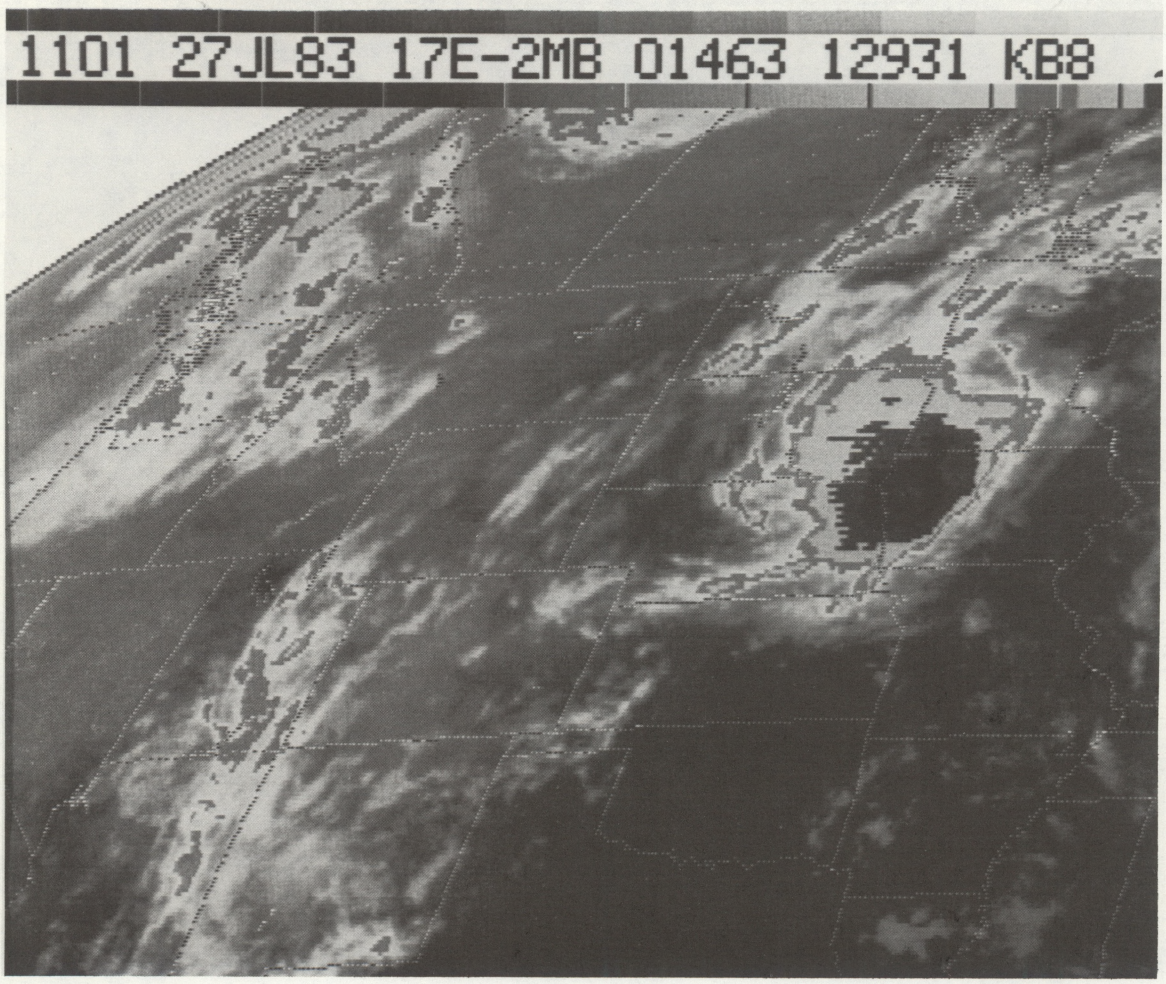


Figure 22a. GOES-East infrared image for 1100 GMT on July 27, 1983.

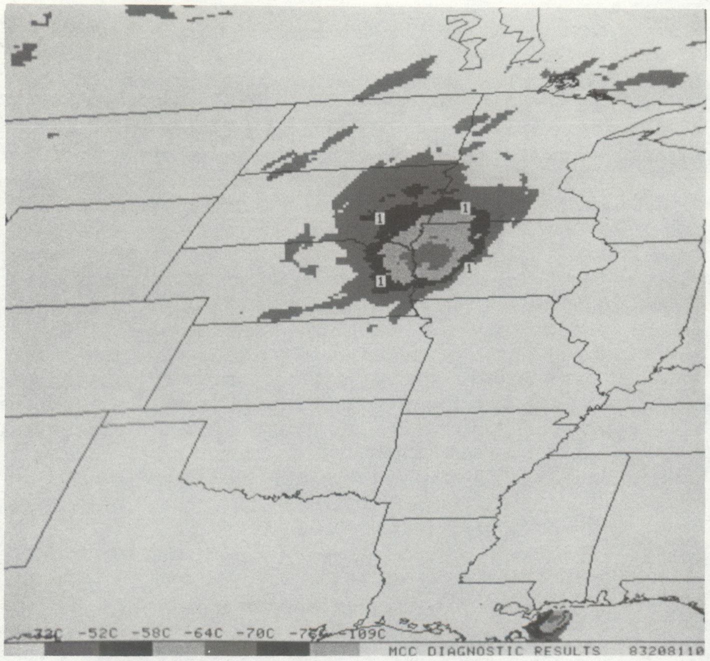


Figure 22b. Video display product of the processed 1100 GMT image in Fig. 22a.

07/27/83		STORM 1 DESCRIPTION		1106 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	1198	0.866987E+05 KM**2	42.75	-96.30
-58C	664	0.477697E+05 KM**2	42.53	-95.95
-64C	144	0.102627E+05 KM**2	42.23	-95.82

THE ECCENTRICITY OF THE -52C AREA IS 0.850

Figure 22c. Output file 832081100.DAT for the labeled storm in Fig. 22b.

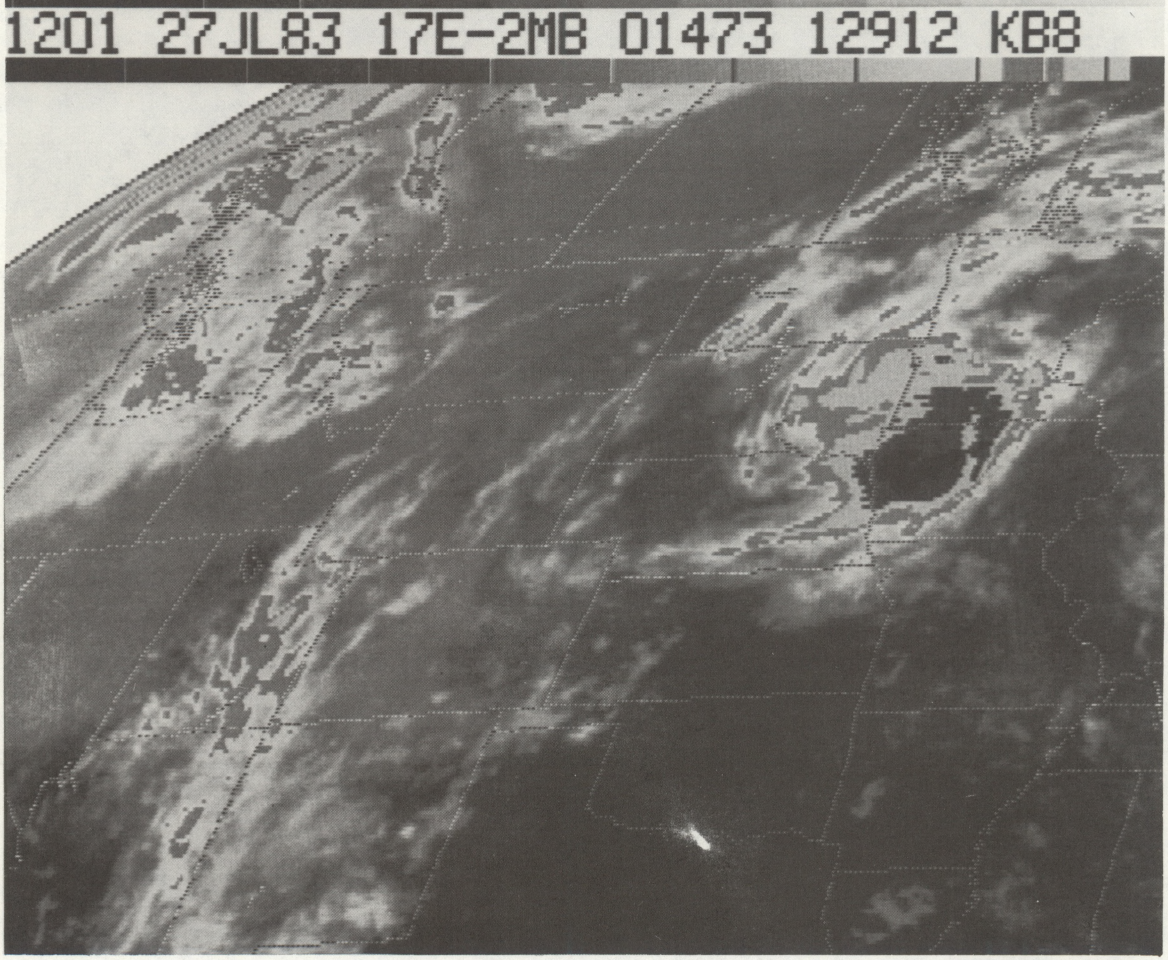


Figure 23a. GOES-East infrared image for 1200 GMT on July 27, 1983.

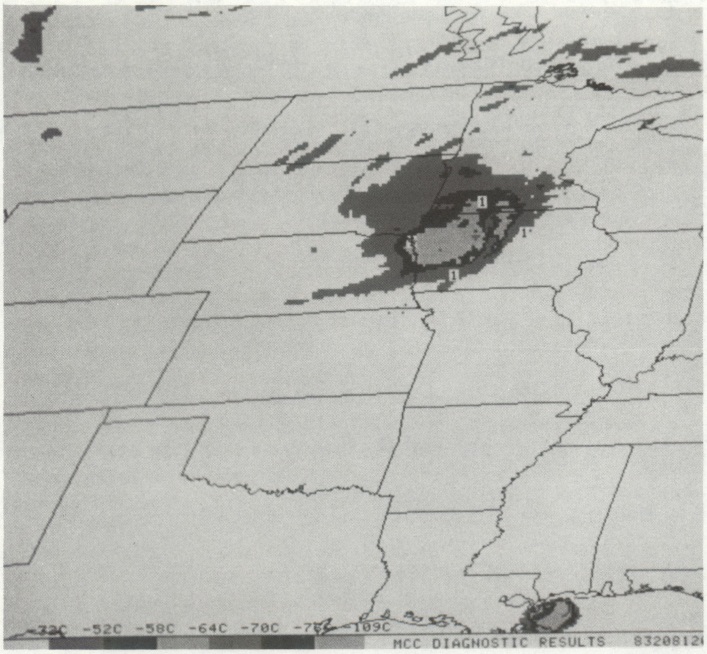


Figure 23b. Video display product of the processed 1200 GMT image in Fig. 23a.

07/27/83		STORM 1 DESCRIPTION		1210 GMT
	N. PIXELS	AREA	CENTROID	
			LAT	LONG
-52C	931	0.673136E+05 KM**2	42.78	-95.22
-58C	443	0.318351E+05 KM**2	42.47	-95.19
-64C	5	0.356469E+03 KM**2	42.45	-94.59

THE ECCENTRICITY OF THE -52C AREA IS 0.885

Figure 23c. Output file 832081200.DAT for the labeled storm in Fig. 23b.

V. SUMMARY AND CONCLUSIONS

An automated technique has been developed to objectively document the cloud-top characteristics of mesoscale convective systems. It replaces a time-consuming manual procedure which has been in use since 1981, in which hardcopy images were analyzed. The software developed operates only on the PROFS-2 VAX 11/780 computer, and accesses digital imagery received at the local ground station. At PROFS, imagery is nominally available up to 48 hours after it is received. The program is user-activated and processes one image per run. It operates on a user-defined sector which is subset from a larger standard digital image. This sector is searched for all cloud tops colder than, or equal to, the cloud definition threshold of -52°C. If the area of the -52°C envelope of a particular cloud top is greater than 10,000 km^2, then its characteristics are documented; if not, the storm is not considered. Cloud-top areas and centroids are measured at several temperatures including -52°C, -58°C, -64°C, -70°C and -76°C. A circularity index (eccentricity) for the cloud top is also measured by computing the minor to major axis ratio of the -52°C envelope. From program initiation to the final display, the algorithm requires approximately one to two minutes of clock time; however, this time span is strongly dependent on the degree of activity by other computer users.

Results are presented in two forms, printed and video. The printed information gives a summary of the documented parameters for all storms which exceed the 10,000 km^2 area threshold at -52°C. The video display shows the processed sector with cloud tops color-coded according to the various temperature thresholds considered. Endpoints of the best-fit ellipses' major and minor axes are also highlighted.

Testing has shown that as long as individual storms are segregated from others at the -52°C level, the computed cloud-top characteristics are accurate and reliable. However, owing to the objectivity of the technique, cloud top mergers at the -52°C level causes problems. Because the cloud-top isolation algorithm links all contiguous pixels as cold as, or colder than -52°C, merged clouds at that level are treated as one cloud. Consequently, the areas, centroids and eccentricity computed for merged cloud tops are practically meaningless, at least from a documentation standpoint. Unfortunately, there is no viable technique to objectively separate merged storms so that they could be analyzed individually. Therefore the user must recognize these situations and flag those storms for manual documentation. Because this problem rarely occurs with well-developed, mature MCS's, this automated technique should prove useful. It has been reliable in testing and should save many valuable man-hours in the annual task of summarizing mesoscale convective systems over the United States.

ACKNOWLEDGMENTS

The author expresses his appreciation to Cecilia G. Griffith, Charles A. Doswell III, Ronald Holle, and Joseph Wakefield for their technical guidance in the development of procedures applied in this work, and critique of this manuscript. Also, thanks go to Stellan Östlund and David Zittel who laid the mathematical foundations upon which the method for the eccentricity calculation was derived. The author is also indebted to the many at PROFS who offered their assistance on computer problems, especially Joseph Wakefield and Joan Brundage. Finally, thanks to PROFS managers who permitted access to their computing facilities and data base.

REFERENCES

Bartels, D. L., J. M. Skradski, and R. D. Menard, 1984: Mesoscale Convective Systems: a satellite-data-based climatology. NOAA Tech. Memorandum ESG 8, Dept. of Commerce, Boulder, CO, 63 pp.

GOES User's Guide, 1983: Compiled and edited by J. D. Clark, National Environmental Satellite, Data, and Information Service, Washington, D.C.

Hambrick, L.N., and D.R. Phillips, 1980: Earth locating image data of spin-stabilized geosynchronous satellites. NOAA Tech. Memorandum, NESS 111, 49 pp.

Maddox, R.A., 1980: Mesoscale convective complexes. Bull. Amer. Soc., 61 (11), 1347-1387.

Maddox, R. A., D. M. Rodgers, and K. W. Howard, 1982: Mesoscale convective complexes over the United States during 1981 - annual summary. Mon. Wea. Rev., 110, 1501-1514.

Östlund, S.S., 1974: Computer software for rainfall analyses and echo tracking of digitized radar data. NOAA Tech. Memorandum ERL WMPO-15, 82 pp.

Rodgers, D. M., K. W. Howard, and E. C. Johnston, 1983: Annual Summary - Mesoscale convective complexes over the United States during 1982. Mon. Wea. Rev., 111, 2363-2369.

Smith E. A., and T. H. Vonder Haar, 1976: Hourly Synchronous Meteorological Satellite-1 (SMS-1) data collected during the GARP Atlantic Tropical Experiment (GATE). Dept. of Atmos. Sci., Colorado State Univ., Fort Collins, CO, 174 pp.

Zittel, W.D., 1976: Evaluation of a remote weather radar display Vol. II-Computer applications for storm tracking and warning. Final Report to U.S. Dept. of Transportation--FAA, Report No. FAA-RD-75-60, II, 94 pp.

APPENDIX A

Approximation of an MCS's -52°C perimeter

The first step in the automated procedure to determine the eccentricity of an MCS's -52°C area is to describe its perimeter. This is accomplished through a specialized contour subroutine called PATHLENGTH, which was developed for this application. Supplied with the starting point, this subroutine locates quasi-regularly spaced, discrete points which define the perimeter of the -52°C area. It searches the perimeter until the initial point is retraced, and the description of the pathlength (in x,y coordinates) is stored in two one-dimensional arrays.

The contouring method used is adapted from that developed for radar data by Östlund (1974) in that it operates on a two-dimensional array in which the threshold (value to be contoured) is subtracted from all elements. This leaves the zero line as the desired contour. This differencing is carried out in subroutine READ; there, the modified image array is loaded into the Real*4 array Z(M,N) which, subsequently, is accessed strictly for contouring. To ensure that all contours are closed, the border values of this contouring array are set to high negative values. Finally, to simplify the routine, all zero (or threshold) values in the contouring array are set to +0.1 so that all contour crossings occur between data points and never at a data point node.

PATHLENGTH is designed to operate on a two-dimensional array with the origin either in the upper left or lower left corner. The first index should always increase from left to right. For this application, the origin of the data array is the upper left corner; therefore the second index increases from top to bottom -- just in the manner in which digital GOES data are stored (see Fig. A1). An important assumption is that the starting point of an entity to be contoured is the first point encountered above (or at) the threshold as the rows are searched from left to right, sequentially, beginning with the first row. Example FORTRAN code showing how a start point must be located is given below:

```
DO J = 2, N-1
   DO I = 2, M-1
      IF(Z(I,J).GE.THRESH) CALL PATHLENGTH(I,J)
   ENDDO
ENDDO
```

This method of locating the starting point ensures that the contour path through the initial box to be searched is not ambiguous.

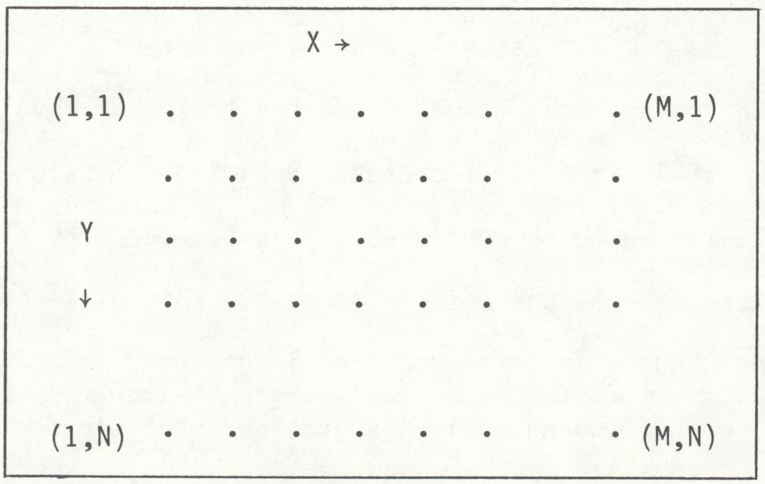


Figure A1. Array orientation convention for digital satellite sectors.

Only one contour is traced per call to PATHLENGTH. The contouring begins at the start point (ISTART, JSTART) which is supplied to PATHLENGTH through input arguments, and the contour is traced through a series of adjacent four-point boxes. An example in Fig. A2 shows a box defined by four data points with the zero contour entering through the left side and exiting through the bottom. The first four-point box to be searched is defined with the start point (ISTART, JSTART) as its lower left corner (for the array orientation

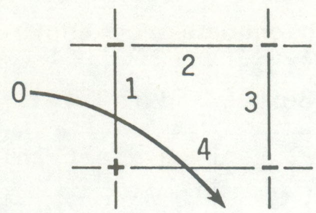


Figure A2. Example of a zero contour traversing a box of four data points within the data matrix.

shown in Fig. A1). The left vertical side is defined as side 1; the other sides are defined sequentially in a clockwise direction (see Fig. A2). This side labeling convention is used in the software. Sides are searched for contour crossings sequentially, beginning with side 1. A side containing a contour crossing is identified when a negative product of two data points defining the side is computed; the exact location of the contour crossing point is determined by linear interpolation. x locations of contour crossing points are stored in the XS array, and the locations of y crossing points are stored in the YS array. The initial box is assumed to have two contour crossing points, one where the contour enters the box and the other where it exits. The side where a contour crossing point is first encountered is considered the contour entry side, and the second is the contour exit side.

The side through which the contour exits the initial box determines the second box to be searched. This exit side then becomes the contour entry side of the new box. In Fig. A3, for example, the contour exits the initial box through side 3, thus designating the box to the right of the initial box as the second box to be searched. The second box is then searched and its contour exit side found, thus defining the third box. This process continues until the first point of the perimeter (XS(1),YS(1)) is retraced. At that point, the complete description of the perimeter will have been stored in the XS and YS arrays.

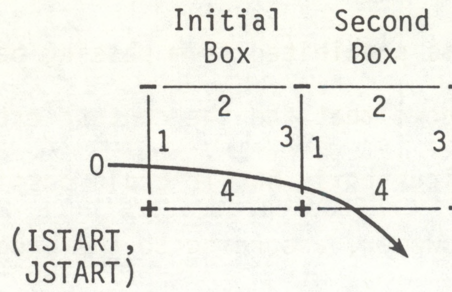


Figure A3. Example of a contour exiting the initial box through side 3 and entering the second search box through side 1. The lower left node of the initial box is the start point, (ISTART, JSTART).

The only two situations in the program logic in which the path of a contour through a box is ambiguous are shown in Fig. A4. In such cases, two zero lines pass through the same box, and all sides possess possible contour crossing locations. When either of these situations occurs, the program logic must decide which of the three possible contour intersection points represents the correct path out of the box for the entity being contoured.

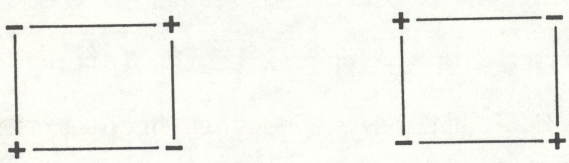


Figure A4. Examples of the two situations for which the path of a contour through a box is ambiguous in the program logic.

Unfortunately, there is more than one intrinsically correct path which may be considered. In order to remain objective, the author chose one as correct; it is the path out of the box that will not intersect the diagonal connecting the two positive data points. The logic considered is that since those two points are positive, the zero line is prohibited from passing between them. For example, in Fig. A5, it is known that the the contour enters through side 1 (this is defined by the previous box), but it could possibly exit through any of the other three sides. However, according to the program logic, the zero contour must exit through side 4; otherwise it would violate the positive diagonal connecting the upper left and lower right nodal points. Specialized logic in PATHLENGTH recognizes these situations and determines the prescribed path out of the box. The method suggested for locating the start point of a -52°C cloud top prevents this type of situation from occurring in the initial search box.

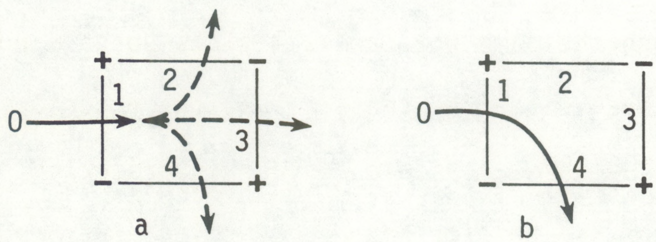


Figure A5. (a) Example of an ambiguous contour path. (b) So as not to violate the diagonal connecting the upper left and lower right positive nodal points, the contour must exit through side 4.

To compute the best-fit ellipse for the -52°C area, its perimeter is subjected to harmonic analysis. Because this technique requires evenly spaced data, it is desirable that the points defining the perimeter, recorded in the XS and YS arrays, be as uniformly spaced as possible. To achieve this, the location of each newly found perimeter point is compared with the previous one. If the distance separating them is less than a quarter of a grid square in both the x and y directions, then the new point is rejected. This places a minimum threshold distance between any two adjacent points along the perimeter. By virtue of the sequential box-searching method used by this subroutine, the maximum distance between successive points on the perimeter is limited by the diagonal of a grid square box. These limitations result in a quasi-evenly spaced perimeter description.

APPENDIX B

Mathematical formulation for the ellipse-fitting technique in a Cartesian coordinate system

This method uses harmonic analysis to determine the best-fit ellipse for the perimeter of some area. It requires as input a description of the pathlength around the outer edge of an area in discrete data points that are evenly spaced. The true path around the entity is approximated by all N/2 harmonics:

$$x(s) = \sum_{n=0}^{N/2} a_n \cos\left(\frac{2\pi ns}{L}\right) + b_n \sin\left(\frac{2\pi ns}{L}\right) , \tag{B1}$$

$$y(s) = \sum_{n=0}^{N/2} c_n \cos\left(\frac{2\pi ns}{L}\right) + d_n \sin\left(\frac{2\pi ns}{L}\right) , \tag{B2}$$

where N is the number of data points along the path, s is the arc length parameter, n is the harmonic number, a, b, c, and d are amplitude coefficients, and L is the total path length which is described by

$$L = \sum_{i=1}^{N-1} \Delta s_i , \tag{B3}$$

where

$$\Delta s_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} , \tag{B4}$$

Because the best-fit ellipse is described by the first harmonic only, (B1) and (B2) reduce to

$$x(s) = a_o + a \cos\left(\frac{2\pi s}{L}\right) + b \sin\left(\frac{2\pi s}{L}\right) , \tag{B5}$$

and

$$y(s) = c_o + c\cos(\frac{2\pi s}{L}) + d\sin(\frac{2\pi s}{L}) \ , \tag{B6}$$

where a, b, c, and d are the amplitude coefficients for the first harmonic, and a_o and c_o are the zeroth harmonic coefficients, or the arithmetic means of x(s) and y(s) respectively. Expressions for these are

$$a = \frac{2}{L}\int_o^s x(s)\cos(\frac{2\pi s}{L})\,ds \ , \tag{B7}$$

$$b = \frac{2}{L}\int_o^s x(s)\sin(\frac{2\pi s}{L})\,ds \ , \tag{B8}$$

$$c = \frac{2}{L}\int_o^s y(s)\cos(\frac{2\pi s}{L})\,ds \ , \tag{B9}$$

$$d = \frac{2}{L}\int_o^s y(s)\sin(\frac{2\pi s}{L})\,ds \ , \tag{B10}$$

$$a_o = \frac{1}{L}\int_o^L x(s)\,ds \ , \tag{B11}$$

$$c_o = \frac{1}{L}\int_o^L y(s)\,ds \ , \tag{B12}$$

In (B7) and (B8), both x and s are variables. However, they are parametric; and so x may be written in terms of s. Linearity is assumed between successive data points, and thus an expression for x is

$$x(s) = \sum_{i=1}^{N} p_i s_i + q_i \ , \tag{B13}$$

where p_i is the slope of the line between two data points and q_i is the intercept:

$$p_i = \frac{x_i - x_{i-1}}{s_i - s_{i-1}} \ , \tag{B14}$$

$$q_i = x_i - s_i p_i . \tag{B15}$$

Substituting (B13) into (B7) gives the harmonic coefficient "a" as a function of s only:

$$a = \frac{2}{L} \sum_{i=1}^{N} \int_{s_i}^{s_{i+1}} (q_i + p_i s_i) \cos(\frac{2\pi s}{L}) \, ds . \tag{B16}$$

Integrating (B16) yields

$$a = \frac{2}{L} \sum_{i=1}^{N} \left[\frac{q_i L}{2\pi} \sin(\frac{2\pi s_i}{L}) + \frac{p_i L^2}{(2\pi)^2} \cos(\frac{2\pi s_i}{L}) + \frac{s_i L}{2\pi} \sin(\frac{2\pi s_i}{L}) \right]_{s_i}^{s_{i+1}} . \tag{B17}$$

For harmonic coefficient "b", substituting (B13) into (B8) and integrating yields:

$$b = \frac{2}{L} \sum_{i=1}^{N} \left[\frac{-q_i L}{2\pi} \cos(\frac{2\pi s_i}{L}) + \frac{p_i L^2}{(2\pi)^2} \sin(\frac{2\pi s_i}{L}) - \frac{s_i L}{2\pi} \cos(\frac{2\pi s_i}{L}) \right]_{s_i}^{s_{i+1}} . \tag{B18}$$

The expression for harmonic coefficient "c" is analogous to that for "a", and the expression for "d" is analogous to that for "b". However, in these latter expressions, p_i and q_i are computed from the y(s) variable.

Now that the harmonic coefficients are known, substituting expressions for x(s) and y(s) into the equation describing an ellipse yields information from which the major and minor axes' lengths can be computed. Because ellipses fit to MCC's are not necessarily aligned along the x or y axis, the general form of the equation for an ellipse must be used:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey = 1. \tag{B19}$$

Harmonic analysis assigns the mean to the zeroth harmonic coefficients

(a_0, c_0). The ellipse described by the first harmonic of x(s) and y(s) is centered on the point (a_0, c_0). When the terms describing the zeroth harmonic are removed, the ellipse is translated to the origin, so (B19) reduces to

$$Ax^2 + Bxy + Cy^2 = 1, \tag{B20}$$

where the coefficients A, B, and C are unknown. To solve for these three coefficients, three equations are needed. Substituting (B5) and (B6) (without the a_0 and c_0 terms) into (B20) yields

$$(Aa^2 + Bac + Cc^2) \cos^2(\frac{2\pi s}{L}) + (Ab^2 + Bbd + Cd^2) \sin^2(\frac{2\pi s}{L}) +$$

$$2(Aab + B[1/2(ad + bc)] + Ccd) \sin(\frac{2\pi s}{L}) \cos(\frac{2\pi s}{L}) = 1. \tag{B21}$$

For $\frac{2\pi s}{L} = 0$, (B21) reduces to

$$Aa^2 + Bac + Cc^2 = 1. \tag{B22}$$

When $\frac{2\pi s}{L} = \frac{\pi}{2}$,

$$Ab^2 + Bbd + Cd^2 = 1. \tag{B23}$$

Because all terms in (B22) and (B23) are constants, these equations must hold true for all s, and therefore are universal. Substituting (B22) and (B23) back into (B21) leads to the third equation:

$$\cos^2(\frac{2\pi s}{L}) + \sin^2(\frac{2\pi s}{L}) +$$

$$2(Aab + B[1/2(ad + bc)] + Ccd) \sin(\frac{2\pi s}{L}) \cos(\frac{2\pi s}{L}) = 1. \tag{B24}$$

From trigonometry, $\cos^2(\frac{2\pi s}{L}) + \sin^2(\frac{2\pi s}{L}) = 1$

therefore,

$$(Aab + B[1/2(ad + bc)] + Ccd) \sin(\frac{2\pi s}{L}) \cos(\frac{2\pi s}{L}) = 0. \quad \text{(B25)}$$

For this to be true for all s,

$$Aab + B[1/2(ad + bc)] + Ccd = 0. \quad \text{(B26)}$$

Coefficients a, b, c, and d are known from harmonic analysis equations (B7)-(B10). Therefore, (B22), (B23), and (B26) contain three unknowns, A, B, and C:

$$Aa^2 + Bac + Cc^2 = 1,$$

$$Ab^2 + Bbd + Cd^2 = 1,$$

$$Aab + B[1/2(ad + bc)] + Ccd = 0,$$

which are easily solved for by application of Cramer's rule:

$$A = \frac{\begin{vmatrix} 1 & ac & c^2 \\ 1 & bd & d^2 \\ 0 & 1/2(ad + bc) & cd \end{vmatrix}}{\begin{vmatrix} a^2 & ac & c^2 \\ b^2 & bd & d^2 \\ ab & 1/2(ad + bc) & cd \end{vmatrix}},$$

$$B = \frac{\begin{vmatrix} a^2 & 1 & c^2 \\ b^2 & 1 & d^2 \\ ab & 0 & cd \end{vmatrix}}{\begin{vmatrix} a^2 & ac & c^2 \\ b^2 & bd & d^2 \\ ab & 1/2(ad + bc) & cd \end{vmatrix}}, \quad \text{and} \quad C = \frac{\begin{vmatrix} a^2 & ac & 1 \\ b^2 & bd & 1 \\ ab & 1/2(ad + bc) & 0 \end{vmatrix}}{\begin{vmatrix} a^2 & ac & c^2 \\ b^2 & bd & d^2 \\ ab & 1/2(ad + bc) & cd \end{vmatrix}}.$$

At this point, all coefficients in (B20) are known; this relationship represents the best-fit ellipse oriented at some angle Θ from either the x or y axis as in Fig. B1.

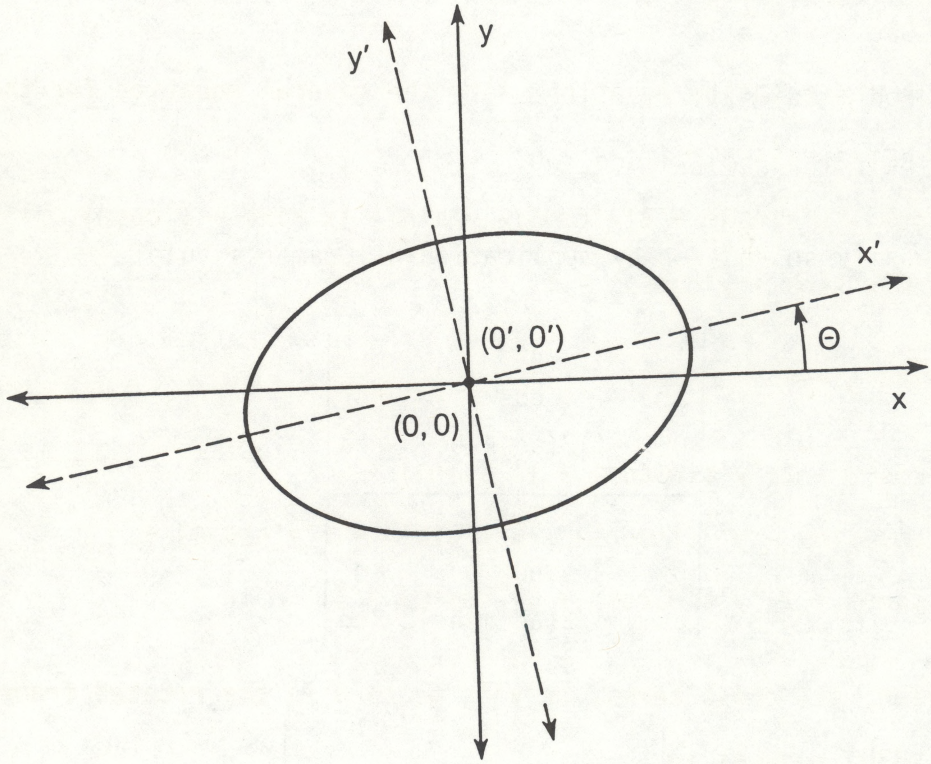


Figure B1. Best-fit ellipse rotated through Θ degrees such that the ellipse's major and minor axes are aligned along the x' and y' axes.

For the lengths of the major and minor axes to be computed, the ellipse must be rotated through Θ. This eliminates the cross term (Bxy) and permits the computation of the axes' lengths from the equation of the ellipse in rotated (x',y') frame.

Axis rotation formulae are

$$x = x'\cos\Theta - y'\sin\Theta, \tag{B27}$$

$$y = x'\sin\Theta + y'\cos\Theta. \tag{B28}$$

An expression for Θ in terms of the ellipse's coefficients is

$$\Theta = \frac{1}{2}\arctan\left(\frac{B}{A - C}\right) \quad . \tag{B29}$$

Substituting the rotation equations into the general equation for the ellipse gives

$$A(x'\cos\Theta - y'\sin\Theta)^2 + B(x'\cos\Theta - y'\sin\Theta)(x'\sin\Theta + y'\cos\Theta)$$
$$+ C(x'\sin\Theta + y'\cos\Theta)^2 = 1,$$

and expanding gives

$$A(x'^2\cos^2\Theta - 2x'y'\sin\Theta\cos\Theta + y'^2\sin^2\Theta) +$$
$$B(x'^2\cos\Theta\sin\Theta - y'^2\sin\Theta\cos\Theta - x'y'\sin^2\Theta + x'y'\cos^2\Theta) +$$
$$C(x'^2\sin^2\Theta + 2x'y'\sin\Theta\cos\Theta + y'^2\cos^2\Theta) = 1. \tag{B30}$$

By definition, all cross terms, $(x'y')$, go to 0 in the rotated frame. Therefore (B30) reduces to

$$A\,x'^2\cos^2\Theta + A\,y'^2\sin^2\Theta + B\,x'^2\cos\Theta\sin\Theta - B\,y'^2\sin\Theta\cos\Theta$$
$$+ C\,x'^2\sin^2\Theta + C\,y'^2\cos^2\Theta = 1, \tag{B31}$$

so collecting terms gives

$$x'^2[A\cos^2\Theta + B\sin\Theta\cos\Theta + C\sin^2\Theta] +$$
$$y'^2[A\sin^2\Theta - B\sin\Theta\cos\Theta + C\cos^2\Theta] = 1. \tag{B32}$$

Therefore the equation of the ellipse on the rotated frame is

$$A'x'^2 + C'y'^2 = 1, \tag{B33}$$

where

$$A' = [A \cos^2\Theta + B \sin\Theta\cos\Theta + C \sin^2\Theta], \tag{B34}$$

$$C' = [A \sin^2\Theta - B \sin\Theta\cos\Theta + C \cos^2\Theta]. \tag{B35}$$

Rearrangement of (B33) to a form recognized as the equation describing an ellipse having its center at the origin and whose axes are aligned along the x' and y' axes, namely,

$$\frac{x'^2}{1/A'} + \frac{y'^2}{1/C'} = 1 , \tag{B36}$$

reveals expressions for the semi-axes' lengths. They are $\sqrt{1/A'}$, which is the length of the ellipse's semi-axis aligned along the x' axis, and $\sqrt{1/C'}$, which is the length of the ellipse's semi-axis aligned along the y' axis.

APPENDIX C

Program listing

```
C MCC DOCUMENTATION PROGRAM
        COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
        COMMON/FNAMES/SATELLITE,FNAM,OUTPUT_FILE
C
        CHARACTER*9 SATELLITE
        CHARACTER*50 FNAM,OUTPUT_FILE
C
C OPEN THE FILE FOR DETAILED LINE PRINTER OUTPUT
        CALL LIB$GET_LUN(LUN_INF)
        OPEN(UNIT=LUN_INF,NAME='INFO.DAT',STATUS='OLD')
C
C
        CALL INITIAL
        CALL READ
        CALL AREA_COR_TABLE
        CALL ISOLATE
        CALL DISPLAY
C
        TYPE 10, OUTPUT_FILE
 10     FORMAT(1X,'RUN COMPLETED, RESULTS ON ',A,/,
     1         1X,'READY FOR PRINTING')
C
C CLOSE FILES
        CLOSE(UNIT=LUN_OUT)
        CLOSE(UNIT=LUN_INF)
C
        STOP
        END
C-----------------------------------------------------------------------------
        SUBROUTINE INITIAL
C
        COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
        COMMON/RAIN/PXAREA,NVALS,NPIXEL(5)
        COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
        COMMON/TABLE/COR(256,250),INC
        COMMON/FNAMES/SATELLITE,FNAM,OUTPUT_FILE
C
        CHARACTER*50 FNAM,OUTPUT_FILE
        CHARACTER*9 SATELLITE
        CHARACTER*10 ANS
C
C
C INDICATE WHICH SATELLITE IS TO BE USED
C (GOES_EAST OR GOES_WEST)
  5     TYPE *, 'ENTER E FOR GOES-EAST OR W FOR GOES-WEST'
        ACCEPT 6, ANS
  6     FORMAT(A)
```

```
C
            IF((ANS.EQ.'E').OR.(ANS.EQ.'e')) THEN
               SATELLITE='GOES EAST'
            ELSE IF((ANS.EQ.'W').OR.(ANS.EQ.'w')) THEN
               SATELLITE='GOES_WEST'
            ELSE
               TYPE *,'ANSWER UNRECOGNIZED, RE-ENTER SATELLITE IDENT.'
               GO TO 5
            ENDIF
C
C SET THE CLOUD ARRAY SIZE
        NRANK=32000
C
C SET THE WARM THRESHOLD TEMPERATURE FOR MCC DEFINITION IN DIGITAL
C   COUNT UNITS (SHOULD BE 197 (-52C))
C
        ITHRES=197
C
C
C SET THE SUBSECTOR SIZE (256 BY 250)
        NLINE=250
        NELEM=256
C
C
C SET THE NAVIGATION INTERPOLATION INCREMENT FOR THE SATELLITE
C DATA ARRAY, AND INITIALIZE ALL ELEMENTS OF THE AREA CORRECTION
C TABLE TO 1.0
        INC=10
        LIMIT=NLINE*NELEM
        DO 10 I=1,LIMIT
 10     COR(I,1)=1.0
C
        RETURN
        END
C---------------------------------------------------------------------------
        SUBROUTINE READ
C
C CONNECT THE SATELLITE DATA BASE, RETRIEVE THE DESIRED
C IMAGE AND CHECK ITS ACCEPTABILITY, IF ALL IS CORRECT:
C
C     1) CONVERT THE DIGITAL COUNTS FROM A 'BYTE' ORIENTED
C        STORAGE (-127 TO +127) TO INTEGER*2 STORAGE (0-255)
C
C     2) ZERO THE BORDER FOR CLOUD ISOLATION PURPOSES
C
        External Blk_IO_CREAT_GG

        COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
        COMMON/UNIT/LUN_OUT,LUN_RAM,LUN_INF
        COMMON/RAIN/PXAREA,NVALS,NPIXEL(5)
        COMMON/IMAGE/ISEC(256,250),Z(256,250)
        COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
        COMMON/FNAMES/SATELLITE,FNAM,OUTPUT_FILE
        COMMON/RMCONV/IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
        COMMON/TIME/NMONTH,NDAY,NYEAR,NTIME
        COMMON/PICS/BSEC(256,250)
        COMMON/SEMI/SEMIMAJOR_AXIS
C
        INTEGER*4 GET_NUM_ELEM_SA,GET_NUM_LINE_SA,LINE_RES
        INTEGER*4 GET_ELEM_RES_SA,GET_LINE_RES_SA,ELEM_RES
        INTEGER*4 GET_I4T_HDR_SA
        INTEGER*4 GET_STRT_LEM_SA,GET_STRT_LIN_SA
```

```
      INTEGER*4 START_ELEM,END_ELEM,START_LINE,END_LINE
      INTEGER*4 HEAD_DECODE_SA
C
      INTEGER*2 ISEC
      INTEGER*2 MASKB
      INTEGER*2 IE,IDE,IL,IDL,IRX,IRY
      INTEGER*2 HEADDOC(6),NUMELE,RECLEN
      INTEGER*2 ICE
C
      REAL*4 CX(11),CY(11),CZ(11)
      REAL*4 Z
C
      CHARACTER*50 FNAM,DISP_FILE,DOC_FILE,MAP_FILE,OUTPUT_FILE
      CHARACTER*9 TIME_STRING,SATELLITE
      CHARACTER FILENAME*13,LABELSTRING*21,STYPE*1,ANS*1
C
      BYTE BSEC
C
      DATA MASKB/'377'O/
C
C
C HAVE THE USER ENTER THE TIME STRING DEFINING THE SATELLITE DATA FILE
      TYPE *,'ENTER THE DATE/TIME OF THE DESIRED IMAGE (YYDDDHHMM)'
      ACCEPT 10,TIME_STRING
 10   FORMAT(A)
C
C
      CALL I4TIME_FNAME_GG(TIME_STRING,I4TIME,ISTATUS)
      WRITE(LUN_INF,1) TIME_STRING,I4TIME
 1    FORMAT(1X,'REQUESTED TIME= ',A9,'  CORRESPONDING I4TIME = ',I10)
C
      WRITE(LUN_INF,*) 'SATELLITE=',SATELLITE
C
C
C
C CONNECT THE SATELLITE DATA BASE FOR THE TIME AND SATELLITE SPECIFIED
C
C    FIRST DECIDE WHETHER OR NOT THE DATA IS TO BE READ FROM THE USER'S
C    DIRECTORY OR THE DEFAULT DIRECTORY [SATDAT]
C     TYPE *,'DO YOU WANT TO READ THE DATA FILE FROM THE USERS'
C     TYPE *,'DIRECTORY? [Y OR N] ...(USE SINGLE QUOTES)'
C     TYPE *,'(IF NO, THE READ WILL DEFAULT TO [SATDAT])'
C     ACCEPT *,ANS
C
C     IF(ANS.EQ.'Y')THEN
C
C   DATA IS TO BE READ FROM THE USER'S DIRECTORY
C
C        FILENAME=TIME_STRING//'.ILE'
C        ISTATUS=HEAD_DECODE_SA(FILENAME,' ',,.FALSE.,HEADDOC,
C    1         LABELSTRING,NUMELE,RECLEN,STYPE)
C        IF(.NOT.ISTATUS)THEN
C           TYPE *,'PROBLEM WITH CONNECT'
C           CALL OUTPUT_DIAG_GG(' ',ISTATUS)
C           STOP
C        ENDIF
C
C
C OPEN AND NAME THE FILE TO CONTAIN THE RAMTEK DISPLAY ARRAY
C        DISP_FILE=TIME_STRING//'.RAM'
C        CALL LIB$GET_LUN(LUN_RAM)
```

```
C           OPEN(UNIT=LUN_RAM,NAME=DISP_FILE,STATUS='NEW',ACCESS='DIRECT',
C     1     RECL=128,USEROPEN=BLK_IO_CREAT_GG)
C
C OPEN THE OUTPUT FILE WHERE MCC DIAGNOSTICS ARE TO BE PRINTED
C           OUTPUT_FILE=TIME_STRING//'.DAT'
C           CALL LIB$GET_LUN(LUN_OUT)
C           OPEN(UNIT=LUN_OUT,NAME=OUTPUT_FILE,STATUS='NEW')
C
C
C FILL IN A DUMMY VALUE FOR "FNAM"
C           FNAM='USER_DEV:[XXXXXX]'//FILENAME
C
C
C       ELSE
C
C   DATA IS TO BE READ FROM [SATDAT]
C
            ISTATUS=HEAD_DECODE_SA(SATELLITE,'IR_LARGE',I4TIME,.FALSE.,
     1                    HEADDOC,LABELSTRING,NUMELE,RECLEN,STYPE)
            IF(.NOT.ISTATUS)THEN
              TYPE *,'PROBLEMS WITH CONNECT'
              CALL OUTPUT_DIAG_GG(' ',ISTATUS)
              STOP
            ENDIF
C
C GET THE FILE NAME OF THE CONNECTED FILE
            CALL GET_CNCT_FNM_SA(FNAM,ISTATUS)
            WRITE(LUN_INF,*) 'FNAM=',FNAM
            LOC=INDEX(FNAM,'.')
C
C CHECK IF THE CONNECTED FILE IS THE SAME AS THE REQUESTED FILE
C
            NLOC=LOC-1
            TIME_STRING = FNAM(18:NLOC)
            CALL I4TIME_FNAME_GG(TIME_STRING,I4NEW,ISTATUS)
            IF(I4TIME.NE.I4NEW) THEN
              TYPE *,' '
              TYPE *,'--WARNING-- REQUESTED IMAGE NOT AVAILABLE,'
              TYPE *,'            IMAGE REPRESENTING ',TIME_STRING
              TYPE *,'            WAS SUBSTITUTED'
              TYPE *,' '
              TYPE *,'  DO YOU WANT TO CONTINUE?'
              TYPE *,'  Y OR N'
              ACCEPT 10,ANS
              IF((ANS.EQ.'N').OR.(ANS.EQ.'n')) THEN
                STOP
              ENDIF
            ENDIF
C
C
C OPEN AND NAME THE FILE TO CONTAIN THE RAMTEK DISPLAY ARRAY
            DISP_FILE=FNAM(18:LOC)//'RAM'
            CALL LIB$GET_LUN(LUN_RAM)
            OPEN(UNIT=LUN_RAM,NAME=DISP_FILE,STATUS='NEW',ACCESS='DIRECT',
     1      RECL=128,USEROPEN=BLK_IO_CREAT_GG)
C
C OPEN THE OUTPUT FILE WHERE MCC DIAGNOSTICS ARE TO BE WRITTEN
            OUTPUT_FILE = FNAM(18:LOC)//'DAT'
            CALL LIB$GET_LUN(LUN_OUT)
            OPEN(UNIT=LUN_OUT,NAME=OUTPUT_FILE,STATUS='NEW')
C
C       ENDIF
```

```
C
C
C
C
C SATELLITE DATA SUCCESSFULLY RETRIEVED, CONTINUE PROCESSING
C
C   GET THE ABSOLUTE COORDINATES OF THE UPPER LEFT CORNER OF THE
C   SATELLITE DATA FILE FOR NAVIGATION PURPOSES (IR RES.)
C
            ISTELEM=GET_STRT_LEM_SA(ISTATUS)/GET_ELEM_RES_SA(ISTATUS)
            ISTLINE=GET_STRT_LIN_SA(ISTATUS)/GET_LINE_RES_SA(ISTATUS)
            WRITE(LUN_INF,*)'STARTING IR LINE AND ELEMENT OF THE LARGE'
            WRITE(LUN_INF,*)'FILE ARE :',ISTLINE,ISTELEM
C
C   DEFINE THE RESOLUTION OF THE IMAGERY FOR COORDINATE TRANSFORMATION
            LINE_RES=GET_LINE_RES_SA(ISTATUS)
            ELEM_RES=GET_ELEM_RES_SA(ISTATUS)
C
C   CHECK THE DIMENSIONS OF THE ACCESSED FILE (IR RES)
            IDIM=GET_NUM_ELEM_SA(ISTATUS)/GET_ELEM_RES_SA(ISTATUS)
            JDIM=GET_NUM_LINE_SA(ISTATUS)/GET_LINE_RES_SA(ISTATUS)
C
C
C FILL THE /RMCONV/ COMMON BLOCK TO BE REFERENCED BY SUBSEQUENT
C   RAMTEK GRAPHICS ROUTINES (FULL RES VIS)
        IE=GET_STRT_LEM_SA(ISTATUS)
        IDE=GET_NUM_ELEM_SA(ISTATUS)
        RIE=FLOAT(GET_ELEM_RES_SA(ISTATUS))
        IL=GET_STRT_LIN_SA(ISTATUS)
        IDL=GET_NUM_LINE_SA(ISTATUS)
        RIL=FLOAT(GET_LINE_RES_SA(ISTATUS))
        IRX=0
        IRY=0
        WRITE(LUN_INF,*) ,  'LARGE SECTOR COORDINATES'
        WRITE(LUN_INF,*) 'IE,IDE,RIE,IL,IDL,RIL,IRX,IRY'
        WRITE(LUN_INF,*) IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
C
C ACCESS TIME STRING OF THE CURRENT IMAGE
            I4TIME=GET_I4T_HDR_SA(ISTATUS)
            IF(.NOT.ISTATUS) THEN
               WRITE(LUN_OUT,15)
               TYPE 15
 15            FORMAT(1X,'INVALID TIME/DATA STRING, STOP EXECUTION')
               STOP
            ENDIF
C
C
C PRINT THE TIME OF THE CURRENT IMAGE
            CALL CV_I4TIM_INT_GG(I4TIME,NYEAR,NMONTH,NDAY,NHOUR,NMIN,
     1      NSEC,ISTATUS)
            NTIME=NHOUR*100 + NMIN
            WRITE(LUN_INF,50) NMONTH,NDAY,NYEAR,NTIME
            WRITE(LUN_OUT,51) NMONTH,NDAY,NYEAR,NTIME
            TYPE 50, NMONTH,NDAY,NYEAR,NTIME
 50         FORMAT(1X,'THE CURRENT IMAGE IS FOR',I3,'/',I2,'/',I2,
     1      '  AT',I4.4,' GMT')
 51         FORMAT(1X,'THE IMAGE DATE/TIME IS',I3,'/',I2,'/',I2,
     1      '  AT',I4.4,' GMT')
C
C
C
C INITIALIZE THE NAVIGATION PROCEDURE
```

```
C
C   FILL THE SATELLITE LINE DOC FILE, AND IF ONE DOES NOT EXIST,
C    CREATE ONE
          CALL GETSATDOC_SA(ISTATUS)
          IF(.NOT.ISTATUS)THEN
             TYPE *,'PROBLEMS WITH GETSATDOC_SA, STOP EXECUTION'
             STOP
          ENDIF
C
C
C   COMPUTE THE SATELLITE SUBPOINT FOR USE IN AREA_COR_TABLE
C
             CALL EARLOC_SC(7284.0,7644.0,SUBLAT,SUBLON)
             WRITE(LUN_INF,*)'SUBPOINT LATITUDE =',SUBLAT,
     1               '      SUBPOINT LONGITUDE =',SUBLON
C
C
C
C   COMPUTE THE SUBPOINT PIXEL SIZE FOR USE IN AREA COMPUTATIONS

C      DEFINE CONSTANTS (KM AND RADIANS)

             EARTH_RADIUS = 6378.39
             ANGLE_EW = 0.00008391005
             ANGLE_NS = 0.00019174763

C         READ THE VALUE OF THE SEMIMAJOR AXIS FROM THE FILE LOGICALLY
C         ASSOCIATED WITH THE NAME "SATDOC"

          Open(In,Name='SatDoc',Form='UnFormatted',Status='Old',ReadOnly)
          Read(In)GRA1,GRA2,Time1,Date1,SpRA1,SpRA2,SpDc1,SpDc2,CX,CY,CZ,
     1              Zeta,Rho,Eta,SpPer,FrTime,Date,ICE,
     2              IEDate,IETime,SemiMA,OEccen,OrbInc,OAnoml,Perige,AsNode

          WRITE(LUN_INF,90) SEMIMA
   90     FORMAT(1X,'(FROM SATDOC) SEMIMAJOR AXIS =',F12.2)

             IF((SEMIMA.LT.41000.0).OR.(SEMIMA.GT.43000.0)) THEN
                SEMIMAJOR_AXIS=42166.0
             ELSE
                SEMIMAJOR_AXIS = SEMIMA
             ENDIF

          WRITE(LUN_INF,70) SEMIMAJOR_AXIS
   70     FORMAT(1X,'(VALUE USED) SEMIMAJOR_AXIS =',F12.2)


C      SATELLITE ALTITUDE
             ALT = SEMIMAJOR_AXIS - EARTH_RADIUS

C      PIXEL EAST-WEST DIMENSION (MULT. BY 2 BECAUSE PROFS AVERAGES
C                                                  ADJACENT PIXELS)
             DIM_EW=(ANGLE_EW * ALT) * 2.0

C      PIXEL NORTH-SOUTH DIMENSION
             DIM_NS = ANGLE_NS * ALT

C    SUBPOINT PIXEL AREA:
          PXAREA = DIM_EW * DIM_NS

          WRITE(LUN_INF,80) DIM_EW,DIM_NS,PXAREA
   80     FORMAT(1X,'PIXEL AREA INFORMATION:',/,
```

```
     1                '    PIXEL E-W DIMENSION =',F5.2,/,
     2              '    PIXEL N-S DIMENSION =',F5.2,/,
     3              ' PIXEL AREA AT SUBPOINT=',F6.2,//)
C
C
C
C   SUBSET THE SUBSECTOR (256 BY 250) FROM THE LARGE SEARCH AREA

        CALL SUBSET(START_ELEM,END_ELEM,START_LINE,END_LINE)
C
C
C
C READ IN THE IMAGE AND CONVERT FROM BYTE TO INTEGER*2 STORAGE

        WRITE(LUN_INF,*)'SECTOR DEF COORDS AS THEY GO INTO GET_IMAGE:'
        WRITE(LUN_INF,*)'START ELEM=',START_ELEM
        WRITE(LUN_INF,*)'END ELEM=   ',END_ELEM
        WRITE(LUN_INF,*)'START LINE=',START_LINE
        WRITE(LUN_INF,*)'END_LINE=   ',END_LINE
C
        XLINE=FLOAT(IL)
        XELE=FLOAT(IE)
        CALL EARLOC_SC(XLINE,XELE,SLAT,SLON)
           IF((SLAT.EQ.888.8).OR.(SLON.EQ.888.8)) THEN
              TYPE *, 'NORTHWEST CORNER OFF EARTH LIMB, RERUN'
              TYPE *, 'PROGRAM WITH CENTER POINT FURTHER TO THE'
              TYPE *, 'SOUTHEAST, CURRENT PROGRAM TERMINATING'
              STOP
           ENDIF
        WRITE(LUN_INF,*)'NORTHWEST CORNER:',SLAT,SLON
C
        XELE=FLOAT(IE+IDE-INT(RIE))
        CALL EARLOC_SC(XLINE,XELE,SLAT,SLON)
           IF((SLAT.EQ.888.8).OR.(SLON.EQ.888.8)) THEN
              TYPE *, 'NORTHEAST CORNER OFF EARTH LIMB, RERUN'
              TYPE *, 'PROGRAM WITH CENTER POINT FURTHER TO THE'
              TYPE *, 'SOUTHWEST, CURRENT PROGRAM TERMINATING'
              STOP
           ENDIF
        WRITE(LUN_INF,*)'NORTHEAST CORNER:',SLAT,SLON
C
        XLINE=FLOAT(IL+IDL-INT(RIL))
        CALL EARLOC_SC(XLINE,XELE,SLAT,SLON)
        WRITE(LUN_INF,*)'SOUTHEAST CORNER:',SLAT,SLON
C
        XELE=FLOAT(IE)
        CALL EARLOC_SC(XLINE,XELE,SLAT,SLON)
        WRITE(LUN_INF,*)'SOUTHWEST CORNER:',SLAT,SLON
C
        WRITE(LUN_INF,100)
  100   FORMAT(1X,///,5X,'START MCC DIAGNOSIS:',//)
C
        CALL GET_IMAGE_SA(START_ELEM,END_ELEM,1,START_LINE,END_LINE,1,
     1                    NELEM,NLINE,BSEC,ISTATUS)
        IF(.NOT.ISTATUS) THEN
           CALL OUTPUT_DIAG_GG (' ',ISTAT)
           TYPE *,'GET_IMAGE HAD PROBLEMS, DO NOT TRUST THE OUTPUT'
           TYPE *,'UNTIL YOU CHECK THE RAMTEK TO SEE HOW MUCH OF THE'
           TYPE *,'DESIRED SECTOR YOU GOT'
        ELSE
           TYPE *,'IMAGE SUCCESSFULLY RETRIEVED'
```

```
            ENDIF
C
C
            DO 52 J=1,NLINE
            DO 52 I=1,NELEM
 52         ISEC(I,J)=BSEC(I,J)
C
            DO 55 J=1,NLINE
            DO 55 I=1,NELEM
                ISEC(I,J)=ISEC(I,J).AND.MASKB
C       FILL THE CONTOURING ARRAY
                Z(I,J)=FLOAT(ISEC(I,J))
 55         CONTINUE
C
C DISCONNECT THE SATELLITE DATA BASE IF DATA WAS ACCESSED FROM [SATDAT]
C           IF(ANS.NE.'Y') THEN
C
                CALL DSCT_SAT_DB_SA(ISTATUS)
                IF(.NOT.ISTATUS) CALL OUTPUT_DIAG_GG('DRIVECNCT',ISTATUS)
C
C           ENDIF
C
C
C
C SUBTRACT THE THRESHOLD FROM EACH POINT IN THE CONTOURING ARRAY (Z)
        DO 56 I=1,LIMIT
            Z(I,1)=Z(I,1)-FLOAT(ITHRES)
            IF(Z(I,1).EQ.0.0) Z(I,1)=0.1
 56     CONTINUE
C
C ZERO THE BORDER OF THE IMAGE ARRAY FOR CLOUD ISOLATION PURPOSES,
C ALSO SET THE BORDER OF THE CONTOURING ARRAY TO A HIGH NEGATIVE
C VALUE
        DO 60 I=1,NELEM
            ISEC(I,1)=0
            ISEC(I,NLINE)=0
            Z(I,1)=-FLOAT(ITHRES)
 60     Z(I,NLINE)=-FLOAT(ITHRES)
        DO 65 J=1,NLINE
            ISEC(1,J)=0
            ISEC(NELEM,J)=0
            Z(1,J)=-FLOAT(ITHRES)
 65     Z(NELEM,J)=-FLOAT(ITHRES)
C
        RETURN
        END
C-----------------------------------------------------------------------------
C
C
        SUBROUTINE SUBSET(START_ELEM,END_ELEM,START_LINE,END_LINE)
C
C FIND THE COORDINATES OF THE 256 BY 250 SUBSECTOR OF THE LARGE SEARCH
C AREA WITHIN WHICH THE STORM IS CENTRALLY LOCATED
C
        COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
        COMMON/RMCONV/IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
        COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
        COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
C
        INTEGER*2 IE,IDE,IL,IDL,IRX,IRY
        INTEGER*4 START_ELEM,END_ELEM,START_LINE,END_LINE
C
C SET THE SUBSECTOR CENTER POINT MOVE FLAG
```

```
        MOVE=0
C
C DETERMINE IF THE CENTER OF THE DESIRED SUBSECTOR IS WITHIN THE LARGE AREA
C
C    ENTER THE DESIRED SUBSECTOR'S CENTER AND TEST IF IT IS WITHIN THE
C    LIMITS OF THE LARGE SEARCH SECTOR
          TYPE *, 'ENTER THE LATITUDE AND LONGITUDE OF THE DESIRED'
          TYPE *, 'SUBSECTORS  CENTER'
          TYPE *, '   (E.G., 40.5, 90.0)'
          ACCEPT *, CLAT,CLON
C
          CLAT = ABS(CLAT)
          CLON = ABS(CLON)
          CLON = -CLON
C
          WRITE(LUN_INF,*)'SUBSECTOR CENTER LAT/LON=',CLAT,CLON
          CALL IMGLOC_SC(CLAT,CLON,CLINE,CELEM)
          TYPE *,' '
          TYPE *,'THE CORRESPONDING LINE AND ELEMENT ARE: ',CLINE,CELEM
C
C   CHECK IF THE CENTER OF THE SUBSECTOR IS WITHIN THE LARGE SECTOR
          ENDELEM=FLOAT(IE)+(IDIM*RIE)
          ENDLINE=FLOAT(IL)+(JDIM*RIL)
          IF((CELEM.LT.IE).OR.(CELEM.GT.ENDELEM)) THEN
            TYPE *,'CENTER POINT IS NOT WITHIN THE ELEMENT BOUNDS'
            TYPE *,'OF THE LARGE SECTOR, STOP EXECUTION'
            STOP
          ELSE IF((CLINE.LT.IL).OR.(CLINE.GT.ENDLINE)) THEN
            TYPE *,'CENTER POINT IS NOT WITHIN THE LINE BOUNDS'
            TYPE *,'OF THE LARGE SECTOR, STOP EXECUTION'
            STOP
          ENDIF
          TYPE *,' '
          TYPE *,'CENTER OF THE SUBSECTOR IS WITHIN THE LARGE'
          TYPE *,' SEARCH SECTOR'
C
C
C FIND THE COORDINATES OF THE SUBSECTOR
C
C   DEFINE THE INTEGER CENTER OF THE STORM (FULL RES VIS)
          J_CENTER=NINT(CLINE)
          I_CENTER=NINT(CELEM)
          WRITE(LUN_INF,*)'I_CENTER=',I_CENTER,'  J_CENTER=',J_CENTER
C
C   COMPUTE THE OFFSETS FROM THE CENTER POINT (J_BACK LINES NORTH AND I_BACK
C   ELEMENTS EAST) -FULL RES VIS
          J_BACK=125*INT(RIL)
          I_BACK=128*INT(RIE)
          WRITE(LUN_INF,*)'I_BACK=',I_BACK,'   J_BACK=',J_BACK
C
C   COMPUTE THE BORDER VALUES OF THE SUBSECTOR BASED ITS CENTER POINT
C   BUT ACCOUNTING FOR THE EDGES OF THE LARGE AREA (FULL RES VIS)
C
          I_LOW_LIMIT=INT(128.*RIE)+IE
          J_LOW_LIMIT=INT(125.*RIL)+IL
          UI=FLOAT(IDIM-127)
          UJ=FLOAT(JDIM-124)
          I_UPPER_LIMIT=INT(UI*RIE)+IE
          J_UPPER_LIMIT=INT(UJ*RIL)+IL
          WRITE(LUN_INF,*)' '
          WRITE(LUN_INF,*)'INITIAL I_LOW,J_LOW,I_UP,J_UP'
          WRITE(LUN_INF,*)I_LOW_LIMIT,J_LOW_LIMIT,I_UPPER_LIMIT,J_UPPER_LIMIT
```

```
C
            IF(J_CENTER.LT.J_LOW_LIMIT) THEN
               J_CENTER=J_LOW_LIMIT
               MOVE=1
            ELSE IF(J_CENTER.GT.J_UPPER_LIMIT) THEN
               J_CENTER=J_UPPER_LIMIT
               MOVE=1
            ENDIF
C
            IF(I_CENTER.LT.I_LOW_LIMIT) THEN
               I_CENTER=I_LOW_LIMIT
               MOVE=1
            ELSE IF(I_CENTER.GT.I_UPPER_LIMIT) THEN
               I_CENTER=I_UPPER_LIMIT
               MOVE=1
            ENDIF
            WRITE(LUN_INF,*)' '
            WRITE(LUN_INF,*)'ADJUSTED CENTER POINT'
            WRITE(LUN_INF,*)'I_CENTER=',I_CENTER,'   J_CENTER=',J_CENTER
C
C TYPE MESSAGE IF THE CENTER POINT WAS MOVED BY THE PROGRAM
            IF(MOVE.EQ.1) THEN
               XLINE=FLOAT(J_CENTER)
               XELEM=FLOAT(I_CENTER)
               CALL EARLOC_SC(XLINE,XELEM,ADJLAT,ADJLON)
               TYPE *,'THE SUBSECTOR CENTER POINT WAS MOVED TO '
               TYPE *,ADJLAT,',',ADJLON,' BECAUSE THE REQUESTED'
               TYPE *,'SUBSECTOR VIOLATES THE LARGE SECTOR BOUNDARY'
            ENDIF
C
C
C COMPUTE THE RELATIVE "FILE" COORDINATES OF THE UPPER LEFT CORNER
C FOR INPUT ARGUMENTS OF GET IMAGE
            IREL_VIS_FILE=((I_CENTER-I_BACK)-IE) + INT(RIE)
               IREL_FILE=NINT(IREL_VIS_FILE/RIE)
            JREL_VIS_FILE=((J_CENTER-J_BACK)-IL) + INT(RIL)
               JREL_FILE=NINT(JREL_VIS_FILE/RIL)
C
C   COMPUTE THE INPUT ARGUMENTS FOR GET_IMAGE
            START_ELEM=IREL_FILE
            END_ELEM=IREL_FILE+255
            START_LINE=JREL_FILE
            END_LINE=JREL_FILE+249
C
            WRITE(LUN_INF,*)' '
            WRITE(LUN_INF,*)'COORDINATES FOR GET IMAGE IN RELATIVE COORDINATS:'
            WRITE(LUN_INF,*)'START_ELEM=',START_ELEM
            WRITE(LUN_INF,*)'END_ELEM=  ',END_ELEM
            WRITE(LUN_INF,*)'START_LINE=',START_LINE
            WRITE(LUN_INF,*)'END_LINE=  ',END_LINE
C
C DO HOUSE CLEANING - REDEFINE SECTOR DEFINITION VALUES FOR NAVIGATION
C PURPOSES
C   DEFINE THE UPPER LEFT CORNER OF THE SUBSECTOR (IR RES)
            ISTELEM=NINT((I_CENTER-I_BACK)/RIE)
            ISTLINE=NINT((J_CENTER-J_BACK)/RIL)
            WRITE(LUN_INF,*)' '
            WRITE(LUN_INF,*)'NEW START ELEMENT=',ISTELEM,'   NEW START LINE=',
     1                      ISTLINE
C
C   ADJUST THE SECTOR DEFINITION IN THE RMCONV COMMON BLOCK (FULL RES VIS)
            IE=I_CENTER-I_BACK
```

```
        IL=J_CENTER-J_BACK
        IDE=INT(256.*RIE)
        IDL=INT(250.*RIL)
        WRITE(LUN_INF,*)' '
        WRITE(LUN_INF,*)'NEW IE,IL,IDE,IDL'
        WRITE(LUN_INF,*)IE,IL,IDE,IDL
C
      RETURN
      END
C-------------------------------------------------------------------------

      SUBROUTINE AREA_COR_TABLE
C
C PRODUCE AN AREA CORRECTION TABLE CORRESPONDING TO THE
C SATELLITE DATA ARRAY.
C
C   DO THIS BY NAVIGATING EVERY "INC"TH PIXEL, COMPUTING
C   AN AREA CORRECTION FACTOR AT EACH NAVIGATED POINT AND
C   USING LINEAR INTERPOLATION TO GET THE CORRECTION
C   FACTORS FOR THE PIXELS IN BETWEEN.
C
      COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
      COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
      COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
      COMMON/TABLE/ COR(256,250),INC
      COMMON/SEMI/SEMIMAJOR_AXIS
C
      INTEGER*4 LINE_RES,ELEM_RES
C
      DO 20 J=1,NLINE,INC
      JNEW=J
      CALL COR_LINE(J)
      IF(J.EQ.1) THEN
         JOLD=J
         GO TO 20
      ELSE
         DO 15 I=1,NELEM
         CORINC=(COR(I,JNEW)-COR(I,JOLD))/FLOAT(INC)
         LOW=JOLD+1
         IHIGH=JNEW-1
         DO 10 K=LOW,IHIGH
 10      COR(I,K)=COR(I,K-1)+CORINC
 15      CONTINUE
         JOLD=J
      ENDIF
 20   CONTINUE
C
C     DO 21 J=1,15
C  21 WRITE(LUN_INF,22) (COR(I,J),I=1,15)
C  22 FORMAT(1X,15F5.2,/)
C
C FINISH THE PART OF THE TABLE THAT THE 20 LOOP DID NOT REACH
      IF(MOD(NLINE,INC).EQ.0) THEN
         RETURN
      ELSE
         JNEW=NLINE
         IDIF=NLINE-JOLD
         CALL COR_LINE(JNEW)
         DO 30 I=1,NELEM
         CORINC=(COR(I,JNEW)-COR(I,JOLD))/FLOAT(IDIF)
         LOW=JOLD+1
         IHIGH=JNEW-1
```

```
          DO 25 K=LOW,IHIGH
 25          COR(I,K)=COR(I,K-1)+CORINC
 30          CONTINUE
        ENDIF
        RETURN
        END
C-------------------------------------------------------------------------
        SUBROUTINE COR_LINE (J)
C
C PRODUCE A PIXEL AREA CORRECTION FEILD FOR ONE LINE OF
C THE SATELLITE DATA ARRAY.
C
C    DO THIS BY NAVIGATING EVERY "INC"TH PIXEL, CALCULATING
C    AREA CORRECTION FACTORS AND USING LINEAR INTERPOLATION
C    FOR THE VALUES IN BETWEEN.
C
        COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
        COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
        COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
        COMMON/TABLE/ COR(256,250),INC
        COMMON/SEMI/SEMIMAJOR_AXIS
C
        INTEGER*4 LINE_RES,ELEM_RES
C
        J_ABS=(J+ISTLINE-1)*LINE_RES
C
        DO 10 I=1,NELEM,INC
        INEW=I
        I_ABS=(I+ISTELEM-1)*ELEM_RES
        CALL EARLOC_SC(FLOAT(J_ABS),FLOAT(I_ABS),ELAT,ELON)
        CALL ACORR(ELAT,ELON,AREACOR,Z)
        COR(I,J)=AREACOR
        IF(I.EQ.1) THEN
           IOLD=I
           GO TO 10
        ELSE
C INTERPOLATE BETWEEN NAVIGATED POINTS
           CORINC=(COR(INEW,J)-COR(IOLD,J))/FLOAT(INC)
           LOW=IOLD+1
           IHIGH=INEW-1
           DO 5 K=LOW,IHIGH
 5         COR(K,J)=COR(K-1,J)+CORINC
           IOLD=I
        ENDIF
 10     CONTINUE
C
C FINISH THE PART OF THE LINE THAT THE 10 LOOP DID NOT REACH
        IF(MOD(NELEM,INC).EQ.0) THEN
           RETURN
        ELSE
           I_ABS=(NELEM+ISTELEM-1) * ELEM_RES
           CALL EARLOC_SC(FLOAT(J_ABS),FLOAT(I_ABS),ELAT,ELON)
           CALL ACORR(ELAT,ELON,AREACOR,Z)
           COR(NELEM,J)=AREACOR
           IDIF=NELEM-IOLD
           CORINC=(COR(NELEM,J)-COR(IOLD,J))/FLOAT(IDIF)
           LOW=IOLD+1
           IHIGH=NELEM-1
           DO 15 K=LOW,IHIGH
 15        COR(K,J)=COR(K-1,J) + CORINC
        ENDIF
        RETURN
```

```
      END
C-----------------------------------------------------------------------------
      SUBROUTINE ACORR(ELAT,ELON,COR,Z)
      COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
      COMMON/SEMI/SEMIMAJOR_AXIS
C
C COMPUTE THE AREA CORRECTION FACTOR FOR PIXEL AS A
C FUNCTION OF DISTANCE FROM THE SATELLITE SUBPOINT
C
C    INPUT:
C        ELAT = LATITUDE OF THE OBSERVATION POINT
C        ELON = LONGITUDE OF THE OBSERVATION POINT
C        SUBLON = LONGITUDE OF THE SATELLITE SUBPOINT
C        SUBLAT = LATITUDE OF THE SATELLITE SUBPOINT
C    OUTPUT:
C        COR = THE AREA CORRECTION FACTOR
C        Z = ZENITH ANGLE OF THE OBSERVATION POINT
C
      DATA DR/.0174533/,R/6371./,PI/3.141593/
C
      DS=SEMIMAJOR_AXIS
      DIFLON=ABS(ELON-SUBLON)
      DIFLAT=ABS(ELAT-SUBLAT)
      COS_ALPHA=COS(DR*DIFLAT)*COS(DR*DIFLON)
      D=SQRT(R*R + DS*DS - 2.*R*DS*COS_ALPHA)
      ALPHA=ACOS(COS_ALPHA)
      SIN_Z=DS*SIN(ALPHA)/D
      Z=ASIN(SIN_Z)
      COR=1./COS(Z)
      RETURN
      END
C-----------------------------------------------------------------------------
      SUBROUTINE ISOLATE
C
C SEARCH THROUGH A DIGITAL IR SATELLITE DATA SECTOR DIMENSIONED BY
C   'NLINE' LINES AND 'NELEM' ELEMENTS AND:
C
C          1. ISOLATE CLOUDS AT THE -52C THRESHOLD
C          2. COMPUTE THE AREAS AND CENTROIDS AT -52C, -58C,
C               -64C, -70C AND -76C
C          3. COMPUTE THE ECCENTRICITY OF THE SYSTEM (AT -52C)
C
      COMMON/GLOBAL/ITHRES,NRANK,NLINE,NELEM,JDIM,IDIM,LIMIT
      COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
      COMMON/RAIN/PXAREA,NVALS,NPIXEL(5)
      COMMON/IMAGE/ ISEC(256,250),Z(256,250)
      COMMON/DT/ VALUE(32000),IANDJ(32000)
      COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
      COMMON/TABLE/COR(256,250),INC
      COMMON/MAJMIN/ AXIS_LAT(4,20),AXIS_LON(4,20),SYS_CNT
C
      DIMENSION ITOGO(20000),JTOGO(20000),ISQ(3),ICT(255)
C
      INTEGER*2 ISEC
      INTEGER*4 LINE_RES,ELEM_RES
      INTEGER*4 SYS_CNT
C
      REAL*4 Z
C
      DATA ISQ/-1,0,1/, IQUEUE/20000/
C
C INITIALIZE THE SYSTEM COUNTER
```

```
          SYS_CNT=0
C
C
          JSTART=1
          ISTART=1
C
C SEARCH THE SECTOR UNTIL A POINT ABOVE THRESHOLD IS LOCATED
 15       DO 20 J=JSTART,NLINE
          DO 20 I=ISTART,NELEM
          IF(ISEC(I,J).GE.ITHRES) GO TO 25
 20       CONTINUE
C
          IF(SYS_CNT.EQ.0) THEN
            WRITE(LUN_OUT,*) 'NO QUALIFIED STORMS FOUND'
            WRITE(LUN_INF,*) 'NO QUALIFIED STORMS FOUND'
          ENDIF

          RETURN
C
C
  POINT ABOVE THRESHOLD FOUND, START DIAGNOSIS
C
 25       ILOC=I
          JLOC=J
C         WRITE(LUN_INF,30) ILOC,JLOC
C 30      FORMAT(1X,'THE STARTING POINT OF THE CLOUD IS',2I5)
C
C RESET THE SEARCH LINE INDEX TO THE CURRENT LINE
C
          JSTART=J
C
C INITIALIZE THE DIGITAL COUNT HISTOGRAM ARRAY
C
          DO 35 K=1,255
 35       ICT(K)=0
C
C BEGIN ISOLATION OF THE SUBJECT CLOUD
C
          NVALS=0
          NTOGO=1
          NDONE=0
          ITOGO(1)=ILOC
          JTOGO(1)=JLOC
C
C HAS THE ENTIRE CLOUD BEEN FOUND?
C
 40       IF(NDONE.EQ.NTOGO) GO TO 90
C
C IF NOT, REMOVE A LOCATION FROM THE BUFFER
          NDONE=NDONE+1
          IF(NDONE.GT.IQUEUE) NDONE=1
          I=ITOGO(NDONE)
          J=JTOGO(NDONE)
C
C DO AN 8-POINT SEARCH OF THE CURRENT POINT'S NEIGHBORS
C
          DO 85 K=1,3
          DO 85 L=1,3
          ID=I+ISQ(K)
          JD=J+ISQ(L)
          ICV=ISEC(ID,JD)
          IF(ICV.LT.ITHRES) GO TO 80
```

```
C
C CHECK IF DIGITAL COUNT IS IN THE VALID RANGE
        IF((ISEC(ID,JD).GE.0).AND.(ISEC(ID,JD).LE.254)) GO TO 42
C POINT IS NOT IN THE VALID RANGE FOR DC VALUES (0-254)
        WRITE(LUN_INF,41) ID,JD,ISEC(ID,JD)
 41     FORMAT(1X,'ISEC(',2I4,') = ',I5,' NOT IN VALID DC RANGE')
        GO TO 80
C
C ANOTHER POINT OF THE CLOUD HAS BEEN FOUND
 42     NTOGO=NTOGO+1
        IF(NTOGO.GT.IQUEUE) NTOGO=1
        ITOGO(NTOGO)=ID
        JTOGO(NTOGO)=JD
C
C UPDATE PARAMETERS NEEDED FOR THE HISTOGRAM
        INDEX=ICV+1
        ICT(INDEX)=ICT(INDEX)+1
C
C STORE THE VALUE AND LOCATION OF THE CLOUD PIXEL FOR LATER USE
        NVALS=NVALS+1
        IF(NVALS.GE.NRANK) GO TO 45
        VALUE(NVALS)=FLOAT(ICV)
        IANDJ(NVALS)=ID*1024 + JD
        GO TO 80
C
C IF THE CLOUD ARRAY IS FILLED TO CAPACITY (NRANK POINTS), BEGIN THE
C   RANKING PROCESS TO ELIMINATE THE OVERFLOW
C
 45     IF(NVALS.GT.NRANK) GO TO 75
        VALUE(NVALS)=FLOAT(ICV)
        IANDJ(NVALS)=ID*1024 + JD
        N=NVALS/2
 50     NN=2*N
        TEMP=VALUE(N)
        ITEMP=IANDJ(N)
 55     IF(NN.GT.NVALS) GO TO 70
        IF(NN.GE.NVALS) GO TO 60
        IF(VALUE(NN).GT.VALUE(NN+1)) NN=NN+1
 60     IF(TEMP.LT.VALUE(NN)) GO TO 70
        VALUE(NN/2)=VALUE(NN)
        IANDJ(NN/2)=IANDJ(NN)
        NN=NN*2
        GO TO 55
 70     VALUE(NN/2)=TEMP
        IANDJ(NN/2)=ITEMP
        N=N-1
        IF(N.GT.0) GO TO 50
        GO TO 80
C
 75     NVALS=NVALS-1
        IF(FLOAT(ICV).LT.VALUE(1)) GO TO 80
        VALUE(1)=FLOAT(ICV)
        IANDJ(1)=ID*1024+JD
        N=1
        GO TO 50
C
C CLEAR THE PIXEL JUST INDEXED SO THAT IT IS NOT FOUND AGAIN
C
 80     ISEC(ID,JD)=0
 85     CONTINUE
C
C END 8-POINT SEARCH LOOP FOR THE CURRENT POINT, GO BACK AND
```

```
C   SEE IF THE CLOUD HAS BEEN COMPLETELY ISOLATED
C
          GO TO 40
C
C
C
C
C ISOLATION OF THE CLOUD IS COMPLETE, DOCUMENT THE APPROPRIATE
C   MCC CHARACTERISTICS
C
C
C       FIRST COMPUTE THE NUMBER OF PIXELS COLDER THAN -52C FROM
C       THE HISTOGRAM
C
 90       NPIXEL(1)=0
          DO 100 K=198,255
 100      NPIXEL(1)=ICT(K)+NPIXEL(1)
          IF(NVALS.LE.NPIXEL(1)) GO TO 110
          WRITE(LUN_OUT,105) NVALS,NPIXEL(1),ILOC,JLOC
 105      FORMAT(1X,'WARNING, NVALS (',I4,') IS .GT. NPIXEL (',I4,
     1    ') FOR THE CLOUD BEGINNING AT',2I5)
 110      CONTINUE
C         WRITE(LUN_INF,115) NPIXEL(1)
C 115     FORMAT(1X,'NPIX52=',I8)
C
C
C COMPUTE THE NUMBER OF PIXELS COLDER THAN -58C
          NPIXEL(2)=0
          DO 120 K=204,255
 120      NPIXEL(2)=NPIXEL(2)+ICT(K)
C         WRITE(LUN_INF,125) NPIXEL(2)
C 125     FORMAT(1X,'NPIX58=',I8)
C
C
C COMPUTE THE NUMBER OF PIXELS COLDER THAN -64C
          NPIXEL(3)=0
          DO 130 K=210,255
 130      NPIXEL(3)=NPIXEL(3)+ICT(K)
C         WRITE(LUN_INF,135) NPIXEL(3)
C 135     FORMAT(1X,'NPIX64=',I8)
C
C
C COMPUTE THE NUMBER OF PIXELS COLDER THAN -70C
          NPIXEL(4)=0
          DO 140 K=216,255
 140      NPIXEL(4)=NPIXEL(4)+ICT(K)
C         WRITE(LUN_INF,145) NPIXEL(4)
C 145     FORMAT(1X,'NPIX70=',I8)
C
C
C COMPUTE THE NUMBER OF PIXELS COLDER THAN -76C
          NPIXEL(5)=0
          DO 150 K=222,255
 150      NPIXEL(5)=NPIXEL(5)+ICT(K)
C         WRITE(LUN_INF,155) NPIXEL(5)
C 155     FORMAT(1X,'NPIX76=',I8)
C
C
C CHECK FOR THE UNUSUAL CONDITION THAT THERE ARE NO PIXELS IN THE CLOUD
C --IN SUCH CASES DO NOT PROCESS THE CLOUD TOP
C
          NPSUM=0
```

```
      NPSUM=NPIXEL(1)+NPIXEL(2)+NPIXEL(3)+NPIXEL(4)+NPIXEL(5)
      IF(NPSUM.EQ.0) GO TO 15
C
C RANK THE DIGITAL COUNTS THAT MAKE UP THE CLOUD FROM COLDEST TO WARMEST
C
      CALL RANK(NVALS)
C
C COMPUTE THE AREAS AND CENTROIDS AT THE THREE THRESHOLDS
C
      CALL DOCUMENT(ILOC,JLOC)
C
C
C GO TO THE BEGINNING OF THE ALGORITHM TO SEARCH FOR ANOTHER CLOUD
      GO TO 15
      END
C-----------------------------------------------------------------------

      SUBROUTINE RANK(ITOT)
      COMMON/DT/ D(32000),INDEC(32000)
      M=ITOT
  1   M=M/2
      IF(M.EQ.0) RETURN
      K=ITOT-M
      J=1
 2    I=J
 3    L=I+M
      IF(D(L)-D(I)) 5,5,4
 4    T=D(I)
      TI=INDEC(I)
      D(I)=D(L)
      INDEC(I)=INDEC(L)
      D(L)=T
      INDEC(L)=TI
      I=I-M
      IF(I-1) 5,3,3
 5    J=J+1
      IF(J-K) 2,2,1
      END
C-----------------------------------------------------------------------
      SUBROUTINE SHSORT(ID,N)
      DIMENSION ID(32000)
      M=N
   1  M=M/2
      IF(M.EQ.0)RETURN
      K=N-M
      J=1
   2  I=J
   3  L=I+M
      IF(ID(L)-ID(I))5,5,4
   4  IT=ID(I)
      ID(I)=ID(L)
      ID(L)=IT
      I=I-M
      IF(I-1)5,3,3
   5  J=J+1
      IF(J-K)2,2,1
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE DOCUMENT(ILOC,JLOC)
C
C COMPUTE THE PARAMETERS NECESSARY FOR MCC DOCUMENTATION
```

```
C
      COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
      COMMON/RAIN/PXAREA,NVALS,NPIXEL(5)
      COMMON/DT/ VALUE(32000),IANDJ(32000)
      COMMON/NAVIG8/ISTELEM,ISTLINE,SUBLON,SUBLAT,LINE_RES,ELEM_RES
      COMMON/TIME/NMONTH,NDAY,NYEAR,NTIME
      COMMON/TABLE/COR(256,250),INC
      COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
      COMMON/RMCONV/IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
      COMMON/MAJMIN/ AXIS_LAT(4,20),AXIS_LON(4,20),SYS_CNT
C
      INTEGER*4 SYS_CNT
      INTEGER*4 LINE_RES,ELEM_RES
      INTEGER*4 INDEX_SORT(32000)
C
      REAL*4 AREA(5),CENT_LAT(5),CENT_LON(5)
      REAL*4 XS,YS
C
      CHARACTER*4 LABEL(5)
C
C
      DATA LABEL/'-52C','-58C','-64C','-70C','-76C'/
C
C
C
C
C DOCUMENT THE MCC:
C
C    COMPUTE THE AREAS AND CENTROIDS AT -52C, -58C, -64C, -70C AND -76C
C    AND THE ECCENTRICITY OF THE -52C AREA
C
C
C AREA AREA AREA AREA AREA AREA AREA AREA AREA AREA AREA AREA AREA AREA
C
C     AREA CALCULATIONS :
C
      DO 40 L=1,5
         AREA(L)=0.0
         CENT_LAT(L)=0.0
         CENT_LON(L)=0.0
         NPIX=NPIXEL(L)
         IF(NPIX.EQ.0) GO TO 40
C        WRITE(LUN_INF,10) L,NPIX
C  10    FORMAT(1X,'L =',I2,'  NPIX =',I6)
C
         DO 30 K=1,NPIX
            VAL=VALUE(K)
            ID=IANDJ(K)/1024
            JD=MOD(IANDJ(K),1024)
            INDEX_SORT(K) = ID
   30    AREA(L)=AREA(L)+(PXAREA*COR(ID,JD))
C
C REJECT CLOUDS WHOSE AREA AT -52C IS LESS THAN 10,000 KM**2
         IF (AREA(1).LT.10000.0) RETURN
C
C
C
C CENTROID CENTROID CENTROID CENTRIOD CENTROID CENTROID CENTROID CENTROID
C
C     CENTROID CALCULATIONS:
C
C        WRITE(LUN_INF,35) (INDEX_SORT(K),K=1,NPIX)
```

```
C  35       FORMAT(1X,25I4)
            CALL SHSORT(INDEX_SORT,NPIX)
C
C           WRITE(LUN_INF,36)
C  36       FORMAT(1X,//,'SORTED ARRAY')
C           WRITE(LUN_INF,35) (INDEX_SORT(K),K=1,NPIX)
C           WRITE(LUN_INF,37)
C  37       FORMAT(1X,///)
C
C    FIND THE MEDIAN OF THE I VALUES
C
            MIDPOINT=NPIX/2
            IF(NPIX.EQ.1)MIDPOINT=1
C           WRITE(LUN_INF,20) MIDPOINT
C  20       FORMAT(1X,'MIDPOINT =',I6)
            I_MEDIAN=INDEX_SORT(MIDPOINT)
C           WRITE(LUN_INF,38) I_MEDIAN
C  38       FORMAT(1X,'I_MEDIAN=',I4)
C
C     FIND THE MEDIAN OF THE J VALUES
C
C
            DO 39 K=1,NPIX
  39        INDEX_SORT(K)=MOD(IANDJ(K),1024)
C           WRITE(LUN_INF,35) (INDEX_SORT(K),K=1,NPIX)
C           WRITE(LUN_INF,36)
C
            CALL SHSORT(INDEX_SORT,NPIX)

C           WRITE(LUN_INF,35) (INDEX_SORT(K),K=1,NPIX)
C           WRITE(LUN_INF,37)

            J_MEDIAN = INDEX_SORT(MIDPOINT)

C           WRITE(LUN_INF,41) J_MEDIAN
C  41       FORMAT(1X,'J_MEDIAN=',I4)
C
C
C    COMPUTE THE GEOGRAPHICAL COORDINATES OF THE CENTROID
C
C      CONVERT TO FULL RESOLUTION VIS COORDINATES
            J_ABS = (J_MEDIAN + ISTLINE-1) * LINE_RES
            I_ABS = (I_MEDIAN + ISTELEM-1) * ELEM_RES
C      NAVIGATE
            CALL EARLOC_SC(FLOAT(J_ABS),FLOAT(I_ABS),CENT_LAT(L),
     1                    CENT_LON(L))
C
  40    CONTINUE
C
C
C INCREMENT THE SYSTEM COUNTER, AS ANOTHER ABOVE THE 10000 KM**2
C THRESHOLD HAS BEEN FOUND
            SYS_CNT = SYS_CNT + 1
C
C    CHECK IF THE MAX NUMBER OF SYSTEMS HAS BEEN EXCEDED
            IF(SYS_CNT.GT.20) THEN
                WRITE(LUN_OUT,34) SYS_CNT
  34            FORMAT(1X,'SYSTEM LIMIT EXCEDED, SYSTEM',I3,' IGNORED')
                TYPE *,'SYSTEM COUNT (20) EXCEDED'
                RETURN
            ENDIF
C
```

```
C
C
C
CECCENTRICITY ECCENTRICITY ECCENTRICITY ECCENTRICITY ECCENTRICITY ECCENTRICITY
C
C     ECCENTRICITY CALCULATION (-52C AREA ONLY):
C
C        FIRST DESCRIBE THE PATH AROUND THE -52C AREA
C
          CALL PATHLENGTH(ILOC,JLOC)
C
          IF(ICNT.LT.5) THEN
             ECCEN=0.00
             GO TO 49
          ENDIF
          IF(ICNT.GT.9998) THEN
             WRITE(LUN_INF,41) SYS_CNT
             TYPE 41, SYS_CNT
  41         FORMAT(1X,'WARNING - CONTOUR BUFFER FILLED, SUBROUTINE'/,
     1             1X,'PATHLENGTH FAILED AND ECCENTRICITY COULD',/,
     2             1X,'NOT BE COMPUTED FOR STORM NUMBER',I3,/)
             ECCEN=0.00
             GO TO 49
          ENDIF
C
C
C CHECK THE DISTANCE BETWEEN THE LAST TWO POINTS
             XDIF = ABS(XS(ICNT)-XS(ICNT-1))
             YDIF = ABS(YS(ICNT)-YS(ICNT-1))
C
             IF((XDIF.LT..25).AND.(YDIF.LT..25)) THEN
                XS(ICNT-1)=XS(ICNT)
                YS(ICNT-1)=YS(ICNT)
                ICNT=ICNT-1
             ENDIF
C
C NAVIGATE THE POINTS DESCRIBING THE PATH LENGTH
          DO 45 K=1,ICNT
             J_ABS=(YS(K) + ISTLINE-1) * LINE_RES
             I_ABS=(XS(K) + ISTELEM-1) * ELEM_RES
             CALL EARLOC_SC(FLOAT(J_ABS),FLOAT(I_ABS),YLAT,XLONG)
             YS(K)=YLAT
             XS(K)=XLONG
  45      CONTINUE
C
C         WRITE(LUN_INF,46)ILOC,JLOC
C  46     FORMAT (1X,'LONGITUDE VALUES FOR THE PATH STARTING AT',2I4)
C         WRITE(LUN_INF,47) (XS(I),I=1,ICNT)
C  47     FORMAT(1X,10F8.2)
C         WRITE(LUN_INF,48)ILOC,JLOC
C  48     FORMAT(1X,'LATITUDE VALUES FOR THE PATH STARTING AT',2I4)
C         WRITE(LUN_INF,47) (YS(I),I=1,ICNT)
C
C FIT THE BEST-FIT ELLIPSE TO THE -52C AREA AND COMPUTE ITS
C ECCENTRICITY
          CALL ELLIPSE(ECCEN)
C
C
C WRITE TO OUTPUT THE MMCC DOCUMENTATION RESULTS
  49      WRITE(LUN_OUT,50) NMONTH,NDAY,NYEAR,SYS_CNT,NTIME
          WRITE(LUN_INF,50) NMONTH,NDAY,NYEAR,SYS_CNT,NTIME
  50      FORMAT(1X,//,1X,I2.2,'/',I2.2,'/',I2.2,
```

```
     1  10X,'STORM',I3,' DESCRIPTION',12X,I4.4,' GMT',//,
     1  7X,'N. PIXELS',9X,
     1 'AREA',16X,'CENTROID',/,43X,'LAT',6X,'LONG')
C
        DO 60 I=1,5
           IF(NPIXEL(I).NE.0) THEN
             WRITE(LUN_OUT,70) LABEL(I),NPIXEL(I),AREA(I),CENT_LAT(I),
     1       CENT_LON(I)
             WRITE(LUN_INF,70) LABEL(I),NPIXEL(I),AREA(I),CENT_LAT(I),
     1       CENT_LON(I)
           ENDIF
  60    CONTINUE
  70    FORMAT(1X,A4,2X,I6,5X,E12.6,1X,'KM**2',6X,F5.2,3X,F7.2,/)
C
        IF(ECCEN.NE.0.00) THEN
           WRITE(LUN_OUT,80) ECCEN
           WRITE(LUN_INF,80) ECCEN
        ELSE
           WRITE(LUN_OUT,85)
           WRITE(LUN_INF,85)
        ENDIF
C
  80    FORMAT(1X,'THE ECCENTRICITY OF THE -52C AREA IS ',F5.3)
  85    FORMAT(1X,'ECCENTRICITY FOR THE -52C AREA COULD'/,
     1        1X,'NOT BE COMPUTED')
C
        WRITE(LUN_OUT,90)
        WRITE(LUN_INF,90)
  90    FORMAT(1X,////)
C
C
        RETURN
        END
C----------------------------------------------------------------------
        SUBROUTINE PATHLENGTH(ISTART,JSTART)
C
C THIS SUBROUTINE TRACES THE PATH AROUND AN ENTITY DEFINED
C BY THE THRESHOLD, AND GIVEN THE STARTING POINT (ISTART,JSTART)
C  AND ASSUMES:
C
C         1. THE THRESHOLD HAS BEEN SUBTRACTED FROM ALL POINTS
C            IN THE Z ARRAY, AND THAT THE BORDER OF THE ARRAY HAS
C            BEEN SET TO HIGH NEGATIVE VALUES (TO ENSURE THAT
C            ALL CONTOURS ARE CLOSED)
C
C         2. ONLY ONE CONTOUR IS TRACED PER CALL
C
C         3. THE ORIGIN OF THE ARRAY CONTAINING THE DATA FIELD
C            IS IN THE UPPER (OR LOWER) LEFT CORNER, WITH THE I INDEX
C            INCREASING FROM LEFT TO RIGHT AND THE J INDEX
C            INCREASING FROM TOP TO BOTTOM (OR BOTTOM TO TOP).
C
C         4. THE STARTING POINT GIVEN AS INPUT ARGUMENTS WERE
C            LOCATED AS THE FIRST POINT FOUND ABOVE THRESHOLD BY
C            SEARCHING ROWS FROM LEFT TO RIGHT, BEGINNING WITH THE
C            FIRST ROW
C
C
C
        COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
        COMMON/IMAGE/ISEC(256,250),Z(256,250)
        COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
```

```
C
         INTEGER*4 ICROSS(4)
         INTEGER*2 ISEC
C
         REAL*4 Z
         REAL*4 XS,YS
C
C
C INITIALIZE THE CONTOUR CROSSING COUNTER
         ICNT=0
C
C START THE SEARCH AT LOCATION (ISTART,JSTART)
         I=ISTART
         J=JSTART
C
C
C DEFINE THE INITIAL BOX FOR WHICH CONTOUR CROSSINGS ARE TO BE
C COMPUTED AND SEARCH ALL FOUR SIDES FOR CROSSINGS
C
         K=0
C
C   SIDE 1 :
         IF((Z(I,J)*Z(I,J-1)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=1
            ICNT=ICNT+1
            CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
         ENDIF
C
C   SIDE 2 :
         IF((Z(I,J-1)*Z(I+1,J-1)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=2
            ICNT=ICNT+1
            CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
         ENDIF
C
C   SIDE 3 :
         IF((Z(I+1,J-1)*Z(I+1,J)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=3
            ICNT=ICNT+1
            CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
         ENDIF
C
C   SIDE 4 :
         IF((Z(I+1,J)*Z(I,J)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=4
            ICNT=ICNT+1
            CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
         ENDIF
C
C CHECK IF MORE THAN TWO CONTOUR CROSSINGS WERE FOUND IN THIS INITIAL
C BOX
         IF(K.GT.2) THEN
            TYPE *,'INITIAL BOX HAS ',K,' CONTOUR CROSSINGS, THEREFORE'
            TYPE *,'PATHLENGTH AROUND STORM COULD NOT BE FOUND'
            RETURN
         ENDIF
C
         CALL CHECK_DIST
```

```
C
C
C
C BEGIN CONTOURING BEYOND THE INITIAL BOX
C
C
C   DEFINE THE LOWER LEFT CORNER OF THE NEXT BOX BASED ON THE SIDE
C   OF THE SECOND CONTOUR CROSSING OF THE PREVIOUS BOX (ICROSS(2))
C
  45      GO TO (50,60,70,80),ICROSS(2)
C
C
CSIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1 SIDE1
C
C  SIDE 1 IS THE SITE OF THE SECOND CONTOUR CROSSING, DEFINE THE
C  NEXT BOX ACCORDINGLY:
C
  50      I=I-1
          ICROSS(1)=3
          K=1
C
C TEST THE OTHER THREE SIDES OF THE NEW BOX FOR CROSSINGS
C    SIDE 1 :
          IF((Z(I,J)*Z(I,J-1)).LT.0.0) THEN
             K=K+1
             ICROSS(K)=1
             ICNT=ICNT+1
             CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
C         CHECK IF THE CONTOUR IS COMPLETE
             IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
          ENDIF
C
C     SIDE 2 :
          IF((Z(I,J-1)*Z(I+1,J-1)).LT.0.0) THEN
             K=K+1
             ICROSS(K)=2
             ICNT=ICNT+1
             CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
C         CHECK IF THE CONTOUR IS COMPLETE
             IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
          ENDIF
C
C     SIDE 4 :
          IF((Z(I+1,J)*Z(I,J)).LT.0.0) THEN
             K=K+1
             ICROSS(K)=4
             ICNT=ICNT+1
             CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
C         CHECK IF THE CONTOUR IS COMPLETE
             IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
          ENDIF
C
C PRESENT BOX COMPLETELY SEARCHED,
C CHECK IF IT HAS MORE THAN TWO CONTOUR CROSSINGS, IF SO NULLIFY
C WHAT HAS BEEN DONE ON THE PRESENT BOX AND CHOOSE THE SIDE
C OF THE SECOND CROSSING SUCH THAT THE POSITIVE DIAGONAL IS NOT
C VIOLATED.
             IF(K.GT.2) THEN
                ICNT=ICNT-3
                IF(Z(I,J).LT.0.) THEN
C               SIDE 2 IS THE CORRECT SECOND CROSSING SIDE
                   ICROSS(2)=2
```

```
                 ICNT=ICNT+1
                 CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
                 IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                 GO TO 90
              ENDIF
              IF(Z(I,J-1).LT.0.) THEN
C             SIDE 4 IS THE CORRECT SECOND CROSSING SIDE
                 ICROSS(2)=4
                 ICNT=ICNT+1
                 CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
                 IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                 GO TO 90
              ENDIF
           ELSE
              GO TO 90
           ENDIF
C
C
C SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2 SIDE2
C
C SIDE 2 IS THE SITE OF THE SECOND CROSSING
C
 60      J=J-1
         ICROSS(1)=4
 61      K=1
C
C TEST THE OTHER THREE SIDES OF THE NEW BOX
C     SIDE 1 :
         IF((Z(I,J)*Z(I,J-1)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=1
            ICNT=ICNT+1
            CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
C        CHECK IF THE CONTOUR IS COMPLETE
            IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
         ENDIF
C
C      SIDE 2 :
         IF((Z(I,J-1)*Z(I+1,J-1)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=2
            ICNT=ICNT+1
            CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
C        CHECK IF THE CONTOUR IS COMPLETE
            IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
         ENDIF
C
C      SIDE 3 :
         IF((Z(I+1,J-1)*Z(I+1,J)).LT.0.0) THEN
            K=K+1
            ICROSS(K)=3
            ICNT=ICNT+1
            CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
C        CHECK IF THE CONTOUR IS COMPLETE
            IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
         ENDIF
C
C PRESENT BOX COMPLETELY SEARCHED,
C CHECK IF IT HAS MORE THAN TWO CONTOUR CROSSINGS, IF SO NULLIFY
C WHAT HAS BEEN DONE ON THE PRESENT BOX AND CHOOSE THE SIDE
C OF THE SECOND CROSSING SUCH THAT THE POSITIVE DIAGONAL IS NOT
C VIOLATED.
```

```
            IF(K.GT.2) THEN
               ICNT=ICNT-3
               IF(Z(I+1,J-1).LT.O.) THEN
C              SIDE 1 IS THE CORRECT SECOND CROSSING SIDE
                  ICROSS(2)=1
                  ICNT=ICNT+1
                  CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
                  IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                  GO TO 90
               ENDIF
               IF(Z(I,J-1).LT.O.) THEN
C              SIDE 3 IS THE CORRECT SECOND CROSSING SIDE
                  ICROSS(2)=3
                  ICNT=ICNT+1
                  CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
                  IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                  GO TO 90
               ENDIF
            ELSE
               GO TO 90
            ENDIF
C
C
C SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3 SIDE3
C
C SIDE 3 IS THE SITE OF THE SECOND CONTOUR CROSSING
C
  70       I=I+1
           ICROSS(1)=1
  71       K=1
C
C TEST THE OTHER THREE SIDES OF THE NEW BOX (SIDES 2,3 AND 4) FOR THE
C SECOND CROSSING IN THAT BOX
C
C       SIDE 2 :
           IF((Z(I,J-1)*Z(I+1,J-1)).LT.0.0) THEN
              K=K+1
              ICROSS(K)=2
              ICNT=ICNT+1
              CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
C          CHECK IF THE CONTOUR IS COMPLETE
              IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
           ENDIF
C
C       SIDE 3 :
           IF((Z(I+1,J-1)*Z(I+1,J)).LT.0.0) THEN
              K=K+1
              ICROSS(K)=3
              ICNT=ICNT+1
              CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
C          CHECK IF THE CONTOUR IS COMPLETE
              IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
           ENDIF
C
C       SIDE 4 :
           IF((Z(I+1,J)*Z(I,J)).LT.0.0) THEN
              K=K+1
              ICROSS(K)=4
              ICNT=ICNT+1
              CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
C          CHECK IF THE CONTOUR IS COMPLETE
              IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
```

```
          ENDIF
C
C PRESENT BOX COMPLETELY SEARCHED,
C CHECK IF IT HAS MORE THAN TWO CONTOUR CROSSINGS, IF SO NULLIFY
C WHAT HAS BEEN DONE ON THE PRESENT BOX AND CHOOSE THE SIDE
C OF THE SECOND CROSSING SUCH THAT THE POSITIVE DIAGONAL IS NOT
C VIOLATED.
C
              IF(K.GT.2) THEN
                 ICNT=ICNT-3
                 IF(Z(I+1,J-1).LT.0.) THEN
C                SIDE 4 IS THE CORRECT SECOND CROSSING SIDE
                    ICROSS(2)=4
                    ICNT=ICNT+1
                    CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
                    IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                    GO TO 90
                 ENDIF
                 IF(Z(I+1,J).LT.0.) THEN
C                SIDE 2 IS THE CORRECT SECOND CROSSING SIDE
                    ICROSS(2)=2
                    ICNT=ICNT+1
                    CALL INTERP_X(Z(I,J-1),Z(I+1,J-1),I,I+1,J-1)
                    IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                    GO TO 90
                 ENDIF
              ELSE
                 GO TO 90
              ENDIF
C
C
C
C SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4 SIDE4
C
C SIDE 4 IS THE SITE OF THE SECOND CONTOUR CROSSING
C
  80       J=J+1
           ICROSS(1)=2
  81       K=1
C
C TEST THE OTHER THREE SIDES OF THE NEW BOX FOR CONTOUR CROSSINGS
C (SIDES(1,3 AND 4))
C
C     SIDE 1 :
           IF((Z(I,J)*Z(I,J-1)).LT.0.0) THEN
              K=K+1
              ICROSS(K)=1
              ICNT=ICNT+1
              CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
C          CHECK IF THE CONTOUR IS COMPLETE
              IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
           ENDIF
C
C
C      SIDE 3 :
           IF((Z(I+1,J-1)*Z(I+1,J)).LT.0.0) THEN
              K=K+1
              ICROSS(K)=3
              ICNT=ICNT+1
              CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
C          CHECK IF THE CONTOUR IS COMPLETE
              IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
```

```
        ENDIF
C
C     SIDE 4 :
          IF((Z(I+1,J)*Z(I,J)).LT.0.0) THEN
             K=K+1
             ICROSS(K)=4
             ICNT=ICNT+1
             CALL INTERP_X(Z(I+1,J),Z(I,J),I+1,I,J)
C         CHECK IF THE CONTOUR IS COMPLETE
             IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
          ENDIF
C
C
C PRESENT BOX COMPLETELY SEARCHED,
C CHECK IF IT HAS MORE THAN TWO CONTOUR CROSSINGS, IF SO NULLIFY
C WHAT HAS BEEN DONE ON THE PRESENT BOX AND CHOOSE THE SIDE
C OF THE SECOND CROSSING SUCH THAT THE POSITIVE DIAGONAL IS NOT
C VIOLATED.
              IF(K.GT.2) THEN
                 ICNT=ICNT-3
                 IF(Z(I,J).LT.0.) THEN
C                SIDE 3 IS THE CORRECT SECOND CROSSING SIDE
                    ICROSS(2)=3
                    ICNT=ICNT+1
                    CALL INTERP_Y(Z(I+1,J-1),Z(I+1,J),J-1,J,I+1)
                    IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                    GO TO 90
                 ENDIF
                 IF(Z(I+1,J).LT.0.) THEN
C                SIDE 1 IS THE CORRECT SECOND CROSSING SIDE
                    ICROSS(2)=1
                    ICNT=ICNT+1
                    CALL INTERP_Y(Z(I,J),Z(I,J-1),J,J-1,I)
                    IF((XS(ICNT).EQ.XS(1)).AND.(YS(ICNT).EQ.YS(1))) RETURN
                    GO TO 90
                 ENDIF
              ELSE
                 GO TO 90
              ENDIF
C
C
C
C CHECK THE DISTANCE BETWEEN THE CROSSING POINT JUST FOUND AND
C THE PREVIOUS ONE, IF THEY ARE LESS THAN A THRESHOLD DISTANCE
C APART, REJECT THE CURRENT POINT
C
  90      CALL CHECK_DIST
C
C CHECK IF THERE ARE TOO MANY POINTS IN THE PATH LENGTH
          IF(ICNT.GT.9998) THEN
             WRITE(LUN_OUT,100) ICNT
 100         FORMAT(1X,'TOO MANY POINTS IN PATH LENGTH',I6,/,
     1              1X,'ECCENTRICITY MAY BE INCORRECT')
             RETURN
          ENDIF
C
C         TYPE *,'XS=',XS(ICNT),' YS=',YS(ICNT)
C
C
C CONTOUR IS NOT COMPLETE,
C GO BACK AND DEFINE A NEW BOX TO BE SEARCHED
C
```

```
      GO TO 45
C
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE INTERP_Y(Z1,Z2,J1,J2,I)
      COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
C
C COMPUTE THE LOCATION OF THE CROSSING POINT ALONG A
C NORTH-SOUTH SIDE OF THE BOX BY LINEAR INTERPOLATION
C
      YS(ICNT) = -Z2*(FLOAT(J1-J2)/(Z1-Z2)) + FLOAT(J2)
      XS(ICNT) = FLOAT(I)
C
      RETURN
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE INTERP_X(Z1,Z2,I1,I2,J)
      COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
C
C COMPUTE THE LOCATION OF THE CROSSING POINT ALONG AN
C EAST-WEST SIDE OF THE BOX BY LINEAR INTERPOLATION
C
      XS(ICNT) = -Z2*(FLOAT(I1-I2)/(Z1-Z2)) + FLOAT(I2)
      YS(ICNT) = FLOAT(J)
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CHECK_DIST
C
C CHECK THE DISTANCE BETWEEN THE CURRENT CONTOUR CROSSING POINT AND
C THE ONE PREVIOUS, IF BOTH XS AND YS VALUES ARE WITHIN .25 OF A GRID
C POINT FROM EACH OTHER, THEN REJECT THE CURRENT POINT
C
      COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
C
      XDIF = ABS(XS(ICNT)-XS(ICNT-1))
      YDIF = ABS(YS(ICNT)-YS(ICNT-1))
C
      IF((XDIF.LT..25).AND.(YDIF.LT..25)) ICNT=ICNT-1
C
      RETURN
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE ELLIPSE(ECCEN)
      COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
      COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
      COMMON/MAJMIN/ AXIS_LAT(4,20),AXIS_LON(4,20),SYS_CNT
      COMMON/RMCONV/IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
C
      REAL*4 COEF(4),ELPCOF(4)
C
      INTEGER*2 IE,IDE,IL,IDL,IRX,IRY
      INTEGER*2 J_ABS,I_ABS,LIMIT,I_VIS_REL,J_VIS_REL
      INTEGER*2 I_IR_REL,J_IR_REL
C
      INTEGER*4 SYS_CNT
```

```
C
C
          N=ICNT
          NHAR=N/2
C
C
C COMPUTE DISTANCES BETWEEN SUCCESSIVE POINTS ON THE
C CONTOUR AND LOAD THEM CUMULATIVELY INTO THE S ARRAY
      S(1)=0.0
      DO 15 I=2,N
   15 S(I)=S(I-1) + DIST(YS(I),XS(I),YS(I-1),XS(I-1))
      SLENG=S(N)
C
C HARMONIC ANALYSIS ON PERIMETER
C
      CALL HARMONC(SLENG,COEF,AVEX,AVEY)
      WRITE(LUN_INF,20) AVEX,AVEY,SLENG
   20 FORMAT(1X,'AVE. X=',G8.2,' AVE. Y=',G8.2,/,
     1       1X,'PERIMETER LENGTH =',G9.2)
C        WRITE(LUN_INF,10) (COEF(I),I=1,4)
C  10    FORMAT(1X,
C    1        '1ST HARMONIC COEFFICIENTS :  A     B     C     D',/,
C    2        28X,4G9.3,/)
C
C
C TEST THE FIRST HARMONIC
C
C   FIRST HARMONIC COMPUTATION
C     CN=(2.*3.1415926)/SLENG
C     WRITE(LUN_INF,14)
C  14 FORMAT(1X,'   X    ','   Y    ','   S    ','   HX   ','   HY   ',/)
C     DO 16 I=1,N
C     COS1=COS(CN*S(I))
C     SIN1=SIN(CN*S(I))
C     HX= AVEX + COEF(1)*COS1 + COEF(2)*SIN1
C     HY= AVEY + COEF(3)*COS1 + COEF(4)*SIN1
C  16 WRITE(LUN_INF,17) XS(I),YS(I),S(I),HX,HY
C  17 FORMAT(1X,5G7.2)
C
C
C COMPUTE VALUES FOR THE A, B, AND C IN THE GENERAL EQUATION
C FOR AN ELLIPSE : A(X**2) + B(X*Y) + C(Y**2) = 1.0
C  AND LOAD THESE COEFFICIENTS INTO ELPCOF(3)
      CALL METRIC(COEF,ELPCOF)
C
      WRITE(LUN_INF,40) (ELPCOF(I),I=1,3)
   40 FORMAT(1X,'COEFICIENTS FOR THE EQUATION AX**2 + BXY + CY**2 = 1'
     1' ARE :',
     1/,4X,'A =',G10.3,4X,'B =',G10.3,4X,'C =',G10.3,4X)
C
C COMPUTE THE TILT ANGLE (W) OF THE ELLIPSE
      W=ATAN(ELPCOF(2)/(ELPCOF(1)-ELPCOF(3)))/2.0
      WRITE(LUN_INF,45) W * 57.2958
   45 FORMAT(1X,'TILT ANGLE OF THE FIT ELLIPSE = ',F9.3,' DEG.',/)
C
C ROTATE THE ELLIPSE THROUGH ITS TILT ANGLE
      SINW=SIN(W)
      COSW=COS(W)
      APRIME=ELPCOF(1)*(COSW**2) + ELPCOF(2)*SINW*COSW +
     1       ELPCOF(3)*(SINW**2)
      CPRIME=ELPCOF(1)*(SINW**2) - ELPCOF(2)*SINW*COSW +
     1       ELPCOF(3)*(COSW**2)
```

```
C
C
      WRITE(LUN_INF,46) APRIME,CPRIME
   46 FORMAT(1X,'A PRIME =',G10.3,'    C PRIME =',G10.3)
C
C
C BECAUSE DEGREES LAT. AND LONG. ARE NOT CONSISTENT, THE
C MAJOR AND MINOR AXES LENGTHS MUST BE COMPUTED BY LOCATING
C THE POINTS WHERE THESE AXES INTERSECT THE ELLIPSE BORDER,
C THEN APPLY FUNCTION DIST TO COMPUTE THEIR LENGTHS.
C
C   NORTHERN MOST INTERSECTION POINT:
         XLONG=-(SQRT(1.0/CPRIME)) * SINW
         XLAT=(SQRT(1.0/CPRIME)) * COSW
         AXIS_LON(1,SYS_CNT)=XLONG + AVEX
         AXIS_LAT(1,SYS_CNT)=XLAT + AVEY
C
C   SOUTHERN MOST INTERSECTION POINT:
         AXIS_LON(2,SYS_CNT)=-XLONG + AVEX
         AXIS_LAT(2,SYS_CNT)=-XLAT + AVEY
C
C   THE NORTH-SOUTH AXIS LENGTH:
         AXIS_LEN1=DIST(AXIS_LAT(1,SYS_CNT),AXIS_LON(1,SYS_CNT),
     1                  AXIS_LAT(2,SYS_CNT),AXIS_LON(2,SYS_CNT))
C
C
         WRITE(LUN_INF,61)AXIS_LAT(1,SYS_CNT),AXIS_LON(1,SYS_CNT),
     1                     AXIS_LAT(2,SYS_CNT),AXIS_LON(2,SYS_CNT)
   61    FORMAT(1X,'NORTH-SOUTH AXIS INTERSECTION POINTS:',/,
     1          10X,'LAT_NORTH,LON_NORTH   LAT_SOUTH,LON_SOUTH',/,
     2          10X,2F8.2,3X,2F8.2,//)
C
C
C   THE EASTERN MOST INTERSECTION POINT:
         XLONG=(SQRT(1.0/APRIME)) * COSW
         XLAT=(SQRT(1.0/APRIME)) * SINW
         AXIS_LON(3,SYS_CNT)=XLONG + AVEX
         AXIS_LAT(3,SYS_CNT)=XLAT + AVEY
C
C   THE WESTERN MOST INTERSECTION POINT:
         AXIS_LON(4,SYS_CNT)=-XLONG + AVEX
         AXIS_LAT(4,SYS_CNT)=-XLAT + AVEY
C
C   THE EAST-WEST AXIS LENGTH:
         AXIS_LEN2=DIST(AXIS_LAT(3,SYS_CNT),AXIS_LON(3,SYS_CNT),
     1                  AXIS_LAT(4,SYS_CNT),AXIS_LON(4,SYS_CNT))
C
         WRITE(LUN_INF,62)AXIS_LAT(3,SYS_CNT),AXIS_LON(3,SYS_CNT),
     1                     AXIS_LAT(4,SYS_CNT),AXIS_LON(4,SYS_CNT)
   62    FORMAT(1X,'EAST-WEST AXIS INTERSECTION POINTS:',/,
     1          10X,'LAT_EAST,LON_EAST    LAT_WEST,LON_WEST',/,
     2          10X,2F8.2,3X,2F8.2,//)
C
C
      WRITE(LUN_INF,47) AXIS_LEN2,AXIS_LEN1
   47 FORMAT(1X,'ELLIPSE AXIS LENGTH ALONG THE EAST-WEST AXIS =',G10.3,
     1' KM',/
     1 1X,'ELLIPSE AXIS LENGTH ALONG THE NORTH-SOUTH AXIS =',G10.3,
     2' KM')
C
C COMPUTE THE ECCENTRICITY
      IF(AXIS_LEN1.GT.AXIS_LEN2) ECCEN=AXIS_LEN2/AXIS_LEN1
```

```
      IF(AXIS_LEN2.GE.AXIS_LEN1) ECCEN=AXIS_LEN1/AXIS_LEN2
C       WRITE(LUN_INF,48) ECCEN
C   48  FORMAT(/,1X,'ECCENTRICITY = ',G6.3,/)
C
C
C COMPUTE THE RAMTEK COORDINATES OF THE POINTS DEFINING THE MAJOR
C AND MINOR AXES OF THE BEST-FIT ELLIPSE
         DO 30 I=1,4
            CALL IMGLOC_SC(AXIS_LAT(I,SYS_CNT),AXIS_LON(I,SYS_CNT),
     1                     REAL_J_ABS,REAL_I_ABS)
            J_ABS=NINT(REAL_J_ABS)
            I_ABS=NINT(REAL_I_ABS)
C           WRITE(LUN_INF,*)'J_ABS=',J_ABS,'  I_ABS=',I_ABS
C
C        CHECK IF THESE COORDINATES ARE OUT OF THE SECTOR IN THE J DIR.
               LIMIT=IL + IDL
               IF((J_ABS.LT.IL).OR.(J_ABS.GT.LIMIT))THEN
                  DO 35 II=1,4
                     AXIS_LAT(II,SYS_CNT)=0.0
   35                AXIS_LON(II,SYS_CNT)=0.0
                  WRITE(LUN_OUT,37)
                  WRITE(LUN_INF,37)
   37             FORMAT(1X,'MAJOR OR MINOR AXIS COORDS. COULD NOT'/,
     1                   1X,'BE COMPUTED,  OUT OF RANGE')
                  GO TO 31
               ENDIF
C
C        NOW DO THE SAME IN THE I DIRECTION
               LIMIT=IE + IDE
               IF((I_ABS.LT.IE).OR.(I_ABS.GT.LIMIT))THEN
                  DO 36 II=1,4
                     AXIS_LAT(II,SYS_CNT)=0.0
   36                AXIS_LON(II,SYS_CNT)=0.0
                  WRITE(LUN_OUT,37)
                  WRITE(LUN_INF,37)
                  GO TO 31
               ENDIF
C
C POINTS WITHIN SECTOR, CONVERT TO RELATIVE VISIBLE COORDINATES:
            I_VIS_REL=(I_ABS - IE) + INT(RIE)
            J_VIS_REL=(J_ABS - IL) + INT(RIL)
C
C        TO RELATIVE IR COORDINATES:
            I_IR_REL=((I_VIS_REL - 1)/INT(RIE)) + 1
            J_IR_REL=((J_VIS_REL - 1)/INT(RIL)) + 1
C
C        TO RAMTEK COORDINATES:
            AXIS_LON(I,SYS_CNT) = FLOAT((I_IR_REL - 1)*2) + 1.
            AXIS_LAT(I,SYS_CNT) = FLOAT((J_IR_REL - 1)*2) + 1. -4.
C      (4.0 SUBTRACTED FROM THE J POSITION SO THE THE MIDDLE OF
C       THE DISPLAY LETTER (8 CHARS. HIGH) REPRESENTS THE LOCATION
C       OF THE AXES INTERSECTION POINTS)
   30    CONTINUE
   31    CONTINUE
C
C
C
C TEST THE ELLIPSE
C     WRITE(LUN_INF,60)
C  60 FORMAT(4X,'TESTY',4X,'TESTX',4X,'TESTNX',/)
C     DO 50 I=1,25
C        TESTY=FLOAT(I-1) * 0.25
```

```
C          A=ELPCOF(1)
C          B=ELPCOF(2)*TESTY
C          C=(ELPCOF(3)*(TESTY**2))-1.0
C          IF((B**2-(4.*A*C)).LE.0.0) GO TO 50
C          TESTX=SQRT(B**2-(4.*A*C))-B
C          TESTNX=-SQRT(B**2-(4.*A*C))-B
C          TESTX=TESTX/(2.*A)
C          TESTNX=TESTNX/(2.*A)
C       WRITE(LUN_INF,80) TESTY,TESTX,TESTNX
C   80  FORMAT(1X,G8.2,1X,G8.2,1X,G8.2)
C   50  CONTINUE
C       DO 100 I=1,25
C          TESTY=-FLOAT(I-1) * 0.25
C          A=ELPCOF(1)
C          B=ELPCOF(2)*TESTY
C          C=(ELPCOF(3)*(TESTY**2))-1.0
C          IF((B**2-(4.*A*C)).LE.0.0) GO TO 100
C          TESTX=SQRT(B**2-(4.*A*C))-B
C          TESTNX=-SQRT(B**2-(4.*A*C))-B
C          TESTX=TESTX/(2.*A)
C          TESTNX=TESTNX/(2.*A)
C          WRITE(LUN_INF,80) TESTY,TESTX,TESTNX
C  100  CONTINUE
C
         RETURN
         END
C-----------------------------------------------------------------------
      FUNCTION DIST(XLAT1,XLON1,XLAT2,XLON2)
C
C FUNCTION RETURNS THE DISTANCE (KM) BETWEEN TWO POINTS
C (XLAT1,XLON1) AND (XLAT2,XLON2) ON A TANGENT PLANE ON
C THE EARTH. (COURTESY OF DAVE JORGENSEN)
C
C UNITS OF XLAT AND YLAT IS DEGREES
C
C   COMPUTE THE COSINE OF THE MEAN LATITUDE
C
      XMLAT=(XLAT1+XLAT2)/2.0
      COSM=COS(XMLAT*0.017453292)
C
C   CALCULATE THE Y DISTANCE
      YDIST=ABS(XLAT1-XLAT2)*111.19
C
C   CALCULATE THE X DISTANCE
      XDIST=ABS(XLON1-XLON2)*111.19*COSM
C
C   CALCULATE THE TOTAL DISTANCE
C
      DIST=SQRT(XDIST*XDIST + YDIST*YDIST)
C
      RETURN
      END
C-----------------------------------------------------------------------
         SUBROUTINE HARMONC(SLENG,COEF,AVEX,AVEY)
C
         COMMON/PERIM/XS(10000),YS(10000),S(10000),ICNT
C
         REAL*4 COEF(4)
C
         N=ICNT
      DF=1.E-4
      PI=3.1415926
```

```
C
      DO 1 K=1,4
 1    COEF(K)=0.0
C
C
C
      DO 100 I=2,N
      IF(ABS(S(I-1)-S(I)).LE.DF) GO TO 100
      PX=(XS(I)-XS(I-1))/(S(I)-S(I-1))
      PY=(YS(I)-YS(I-1))/(S(I)-S(I-1))
      QX=XS(I)-PX*S(I)
      QY=YS(I)-PY*S(I)
C
      CONST=2.*PI/SLENG
C
      U1=CONST*S(I-1)
      U2=CONST*S(I)
      SIN1=SIN(U1)
      SIN2=SIN(U2)
      COS1=COS(U1)
      COS2=COS(U2)
C
C COMPUTE THE COEFFICIENT FOR THE COS TERM IN X
      T2=QX/CONST*SIN2 + PX*(CONST**(-2)*COS2 + S(I)/CONST*SIN2)
      T1=QX/CONST*SIN1 + PX*(CONST**(-2)*COS1 + S(I-1)/CONST*SIN1)
      COEF(1)=COEF(1) + (2.0/SLENG*(T2-T1))
C
C COMPUTE THE COEFFICIENT FOR THE SIN TERM IN X
      T2=-QX/CONST*COS2+PX*(CONST**(-2)*SIN2-S(I)/CONST*COS2)
      T1=-QX/CONST*COS1+PX*(CONST**(-2)*SIN1-S(I-1)/CONST*COS1)
      COEF(2)=COEF(2) + (2.0/SLENG*(T2-T1))
C
C COMPUTE THE COEFFICIENT FOR THE COS TERM IN Y
      T2=QY/CONST*SIN2 + PY*(CONST**(-2)*COS2 + S(I)/CONST*SIN2)
      T1=QY/CONST*SIN1 + PY*(CONST**(-2)*COS1 + S(I-1)/CONST*SIN1)
      COEF(3)=COEF(3) + (2.0/SLENG*(T2-T1))
C
C COMPUTE THE COEFFICIENT FOR THE SIN TERM IN Y
      T2=-QY/CONST*COS2 + PY*(CONST**(-2)*SIN2 - S(I)/CONST*COS2)
      T1=-QY/CONST*COS1 + PY*(CONST**(-2)*SIN1 - S(I-1)/CONST*COS1)
      COEF(4)=COEF(4) + (2.0/SLENG*(T2-T1))
C
 100  CONTINUE
C
C CALCULATE THE ZEROTH HARMONIC COEFFICIENTS
      AVEX=0.0
      AVEY=0.0
      DO 200 I=2,N
      IF(ABS(S(I-1)-S(I)).LE.DF) GO TO 200
      PX=(XS(I)-XS(I-1))/(S(I)-S(I-1))
      PY=(YS(I)-YS(I-1))/(S(I)-S(I-1))
      QX=XS(I)-PX*S(I)
      QY=YS(I)-PY*S(I)
      T2=.5*PX*S(I)**2 + QX*S(I)
      T1=.5*PX*S(I-1)**2 + QX*S(I-1)
      AVEX=AVEX + (T2-T1)/SLENG
      T2=.5*PY*S(I)**2 + QY*S(I)
      T1=.5*PY*S(I-1)**2 + QY*S(I-1)
      AVEY=AVEY + (T2-T1)/SLENG
 200  CONTINUE
      RETURN
      END
```

```
C-----------------------------------------------------------------------
         SUBROUTINE METRIC(COEF,ELPCOF)
         REAL*4 BAS(3,3),CAS(3,3)
         REAL*4 COEF(4),ELPCOF(4)
C
      A=COEF(1)
      B=COEF(2)
      C=COEF(3)
      D=COEF(4)
C
      BAS(1,1)=A*A
      BAS(1,2)=A*C
      BAS(1,3)=C*C
      BAS(2,1)=B*B
      BAS(2,2)=B*D
      BAS(2,3)=D*D
      BAS(3,1)=A*B
      BAS(3,2)=(B*C+A*D)/2.0
      BAS(3,3)=C*D
C
      DENOM=BAS(1,1)*(BAS(2,2)*BAS(3,3)-BAS(3,2)*BAS(2,3))
     1      -BAS(1,2)*(BAS(2,1)*BAS(3,3)-BAS(3,1)*BAS(2,3))
     2      +BAS(1,3)*(BAS(2,1)*BAS(3,2)-BAS(2,2)*BAS(3,1))
C
C
      DO 10 K=1,3
         DO 20 I=1,3
         DO 20 J=1,3
   20    CAS(I,J)=BAS(I,J)
      CAS(1,K)=1.0
      CAS(2,K)=1.0
      CAS(3,K)=0.0
C
      FNUM = CAS(1,1)*(CAS(2,2)*CAS(3,3)-CAS(3,2)*CAS(2,3))
     1       -CAS(1,2)*(CAS(2,1)*CAS(3,3)-CAS(3,1)*CAS(2,3))
     2       +CAS(1,3)*(CAS(2,1)*CAS(3,2)-CAS(2,2)*CAS(3,1))
C
   10 ELPCOF(K)=FNUM/DENOM
C
      RETURN
      END
C-----------------------------------------------------------------------
         SUBROUTINE DISPLAY
C
C PROPERLY SCALE A 1/2 HOUR RAINFALL IMAGE AND CREATE A FILE TO
C    DISPLAY ON THE RAMTEK
C
         COMMON/UNIT/ LUN_OUT,LUN_RAM,LUN_INF
         COMMON/PICS/ BSEC(256,250)
         COMMON/RMCONV/IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
         COMMON/MAPCOL/MPCOL
         COMMON/FNAMES/SATELLITE,FNAM,OUTPUT_FILE
         COMMON/MAJMIN/ AXIS_LAT(4,20),AXIS_LON(4,20),SYS_CNT
C
         INTEGER*2 IE,IDE,IL,IDL,IRX,IRY
         INTEGER*2 HEADER(256)
         INTEGER*2 NUMHEAD
C
         INTEGER*4 SYS_CNT
         INTEGER*4 FORTERROR_GG
C
         CHARACTER*50 FNAM,OUTPUT_FILE
```

```
      CHARACTER*9 SATELLITE
      CHARACTER*9 DATE_TIME
      CHARACTER SYS_1_9(9)*1,SYS_10_20(11)*2,SYS_1*1,SYS_2*2
C
      BYTE BSEC
      BYTE RAM(0:511,0:511)
      BYTE BLANK(512)
      BYTE LABEL(36)
      BYTE ELABLE1(4),ELABLE2(8)
C
      EQUIVALENCE (DATE_TIME,LABEL)
      EQUIVALENCE (SYS_1,ELABLE1)
      EQUIVALENCE (SYS_2,ELABLE2)
C
      DATA RAM/262144*0/
      DATA BLANK/512*0/
      DATA (SYS_1_9(I),I=1,9)/'1','2','3','4','5','6','7','8','9'/
      DATA (SYS_10_20(I),I=1,11)/'10','11','12','13','14','15',
     1                           '16','17','18','19','20'/

C
C     WRITE(LUN_INF,*)'IE,IDE,RIE,IL,IDL,RIL,IRX,IRY'
C     WRITE(LUN_INF,*)IE,IDE,RIE,IL,IDL,RIL,IRX,IRY
C
C REMOVE THE DATE AND TIME FROM "FNAM" AND LOAD IT INTO "DATE_TIME"
      LOC=INDEX(FNAM,'.')
      LOC=LOC-1
      DATE_TIME=FNAM(18:LOC)
C
C SET UP THE HEADER RECORD
C   FILL ELEMENT 2 WITH THE NUMBER OF HEADER RECORDS TO BE WRITTEN
      HEADER(2)=1
      WRITE(LUN_RAM'1) HEADER
C WRITE A LABEL AT THE BOTTOM RIGHT OF THE IMAGE
      CALL LABTEXT(LUN_RAM,275,502,22,0,0,0,
     1 'MCC DIAGNOSTIC RESULTS',ISTATUS)

      if(.not.istatus)then
        print*,'cant perform labtext...rainfall'
      end if

      CALL LABTEXT(LUN_RAM,449,502,9,0,0,0,LABEL,ISTATUS)

      if(.not.istatus)then
        print*,'cant perform labtext...rain'
      end if
C
C WRITE THE COLOR BAR LABELS TO THE RAMTEK FILE
      CALL LABTEXT(LUN_RAM,22,489,4,0,0,0,'-32C',ISTATUS)
      CALL LABTEXT(LUN_RAM,59,489,4,0,0,0,'-52C',ISTATUS)
      CALL LABTEXT(LUN_RAM,95,489,4,0,0,0,'-58C',ISTATUS)
      CALL LABTEXT(LUN_RAM,131,489,4,0,0,0,'-64C',ISTATUS)
      CALL LABTEXT(LUN_RAM,167,489,4,0,0,0,'-70C',ISTATUS)
      CALL LABTEXT(LUN_RAM,204,489,4,0,0,0,'-76C',ISTATUS)
      CALL LABTEXT(LUN_RAM,239,489,5,0,0,0,'-109C',ISTATUS)
C
C WRITE THE COORDS. OF THE MAJOR AND MINOR AXES OF THE BEST-FIT
C ELLIPSE TO THE RAMTEK FILE HEADER
      DO 20 J=1,SYS_CNT
          IF(J.LT.10) THEN
              DO 10 I=1,4
              SYS_1=SYS_1_9(J)
```

```
C                WRITE(LUN_INF,6) SYS_1
C   6            FORMAT(1X,'IN LOOP, SYS_1= ',A1)
                 CALL LABTEXT(LUN_RAM,INT(AXIS_LON(I,J)),
     1                          INT(AXIS_LAT(I,J)),
     2                          1,0,0,0,ELABLE1,ISTATUS)
  10             CONTINUE
              ELSE
                 DO 15 I=1,4
                 SYS_2=SYS_10_20(J-9)
                 CALL LABTEXT(LUN_RAM,INT(AXIS_LON(I,J)),
     1                          INT(AXIS_LAT(I,J)),
     2                          2,0,0,0,ELABLE2,ISTATUS)
  15             CONTINUE
              ENDIF
C
  20      CONTINUE
C
C
C DUMP LABTEXT MESSAGES INTO THE HEADER
          CALL LABEND(NUMHEAD,ISTATUS)
          WRITE(LUN_INF,*)'NUMBER OF HEADER RECORDS REQUIRED BY'
          WRITE(LUN_INF,*)'LABTEXT IS',NUMHEAD
C
C HEADER FINISHED
C
C
C BLOW UP THE IMAGE TO 512 BY 512
C
          CALL BLOWUP(BSEC,RAM)
C
C   ADJUST APPROPRIATE VARIABLES ACCORDING TO THE BLOW UP
              BFACTOR=2.0
C     ADJUST X RES:
              RIE=RIE/BFACTOR
C     ADJUST Y RES:
              RIL=RIL/BFACTOR
C
C SET THE INTENSITY OF THE MAP BACKGROUND
          MPCOL=255
C
C WRITE THE RAMTEK IMAGE TO THE DISPLAY FILE
          DO 30 NL=1,500
  30      WRITE(LUN_RAM'NL+NUMHEAD,ERR=400) (RAM(I,NL),I=0,511)
C
C
C DRAW THE COLOR TABLE IN THE DISPLAY ARRAY (RAM)
C
          CALL COLOR_FILL(1,36,1,BLANK)
          CALL COLOR_FILL(37,73,2,BLANK)
          CALL COLOR_FILL(74,109,3,BLANK)
          CALL COLOR_FILL(110,145,4,BLANK)
          CALL COLOR_FILL(146,181,5,BLANK)
          CALL COLOR_FILL(182,218,6,BLANK)
          CALL COLOR_FILL(219,254,7,BLANK)

          DO 40 NL=1,12
  40      WRITE(LUN_RAM'500+NL+NUMHEAD,ERR=300) BLANK
C
C         RETURN
C
C
C     DISPLAY THE FILE ON THE RAMTEK AND OVERLAY THE MAP BACKGROUND
```

```
      CALL RAMRECALL
C
C
 300  CALL LOG_ERROR_GG('DISPLAY',FORTERROR_GG(),'WRITE ERROR',0,0)
      ISTATUS=0
      RETURN
C
 400  CALL LOG_ERROR_GG('DISPLAY',FORTERROR_GG(),'IMAGE WRITE ERROR',0,0)
      ISTATUS=0
      RETURN
C
      END
C-----------------------------------------------------------------
      SUBROUTINE COLOR_FILL(IS,IE,INDEX,BLANK)
C
      BYTE BYT
      BYTE BLANK(512)
C
      INTEGER*2 IVAL
      INTEGER*2 INTENSITY(7)
C
      EQUIVALENCE (IVAL,BYT)
C
      DATA INTENSITY/100,185,200,205,211,217,230/
C
      DO 10 I=IS,IE
      IVAL=INTENSITY(INDEX)
  10  BLANK(I+1)=BYT
C
      RETURN
      END
C-----------------------------------------------------------------
$
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1985—576-000/20,016 REGION NO. 8