GODDARD
GRANT
IN-47-CR
71418
P-44

A Final Report to

The National Aeronautics and Space Administration

for

ANTECEDENT PRECIPITATION INDEX DETERMINED FROM CST ESTIMATES OF RAINFALL

Grant # NAG5-1478

University of Wisconsin Account # 144-BY14

for the period of

15 November 1990 to 15 February 1992

submitted by

David W. Martin

Space Science and Engineering Center
at the University of Wisconsin-Madison
1225 West Dayton Street
Madison, Wisconsin 53706
(608) 262-0544

February 1992

ACKNOWLEDGEMENTS

David Molitor transcribed the infrared images. Brian Goodman and Barry Rowe supported the rain and related programs. David Scheer generated most of the daily rain images. Professors Isa Silva and Edson Rocha granted use of their PC McIDAS. To each and every one, I extend profound thanks.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

INTRODUCTION 1

NATURE OF THE EXPERIMENT 1

CONTRIBUTION OF SSEC 2

Resources
Procedure

RESULTS 6

CONCLUSIONS 7

REFERENCES 8

LEGENDS 9

FIGURES 10

APPENDICES 18

A. Protocol

B. Code for CST4

C. Correction for Summed Rain-Image Directories

ANTECEDENT PRECIPITATION INDEX DETERMINED FROM CST ESTIMATES OF RAINFALL

INTRODUCTION

The Space Science and Engineering Center has been collaborating with the Goddard Space Flight Center in an experimental calculation of a satellite-based antecedent precipitation index (API). The index is to be derived from daily rain images produced from infrared images using an improved version of GSFC's Convective/Stratiform Technique (CST; Adler and Negri, 1988). The present report summarizes the nature of the experiment and then describes the contribution of the University of Wisconsin.

NATURE OF THE EXPERIMENT

Antecedent Precipitation Index, in its various forms, is meant to be a measure of soil moisture. It is based on the notion that the amount of moisture in a soil at a given time is related to precipitation at earlier times. The present experiment deals with warm-season conditions; therefore, for "precipitation" we may read "rainfall".

Because the influence of a fall of rain on API depends on amount as well as passage of time, the quality of the rain observation cannot be neglected. Whether any of the extant infrared techniques for estimating rainfall is adequate is an open question. But inasmuch as it demonstrably does distinguish between convective and stratiform rainfall, CST may hold an edge over other techniques.

Using code supplied by GSFC before the present project began, we placed old and new versions of CST on the University of Wisconsin's Man-computer Interactive Data Access System (McIDAS). Subsequent to the start of the project, GSFC scientists gave us code for a third version of CST. At about that time we developed our own version.

Both of the GSFC revisions were aimed at improving the estimate of the height of the tropopause (hence the possible extent of stratiform rain). The first was intended to take account of the difference between the local height of the tropopause and the height of the tropopause in the CST calibration region. We refer to this version as CST2. The second GSFC revision was intended to reference the estimate of tropopause height to a clump of cumulonimbus clouds rather than the whole image. We call this version CST4. Both addressed a bias in CST toward high rainfall. The SSEC revision provided a new discriminant between thin cirrus and anvil cirrus. This is CST3. Finally, in our nomenclature the original (1988) version is CST1.

CONTRIBUTION OF SSEC

The first part of SSEC's contribution to the experimental calculation of API was to test the three variations of CST. The second (and larger) part was use of the superior version to calculate daily rainfall for one month over an area centered on eastern Kansas.

Resources

In addition to the four variations of CST, resources available to us were an archive of Geostationary Operational Environment Satellite (GOES) infrared images and the McIDAS, including personal computer work stations. Through GSFC, we also had code for a second rain estimation technique, the GOES Precipitation Index (Arkin, 1979; Janowiak and Arkin, 1991), which hereafter is referred to as ARK.

The PC class of McIDAS work station is described by Ide (1988) and Dengel et al. (1989). By coincidence, a work station purchased by the Federal University of Para (UFPa), Belem, was being tested prior to delivery. This station includes a magneto/optical drive, which proved to be valuable in saving and restoring infrared as well as rain images. Archive GOES data were transcribed to computer-compatible tape and loaded when needed from tape through the mainframe McIDAS. Otherwise, images were processed entirely on the UFPa work station.

Procedure

The project at SSEC was conducted in three parts: selection of a case, test of CST and production. The case--a calendar month of hourly GOES infrared images--was to be chosen from June and July data from the years 1988, 1989 and 1990. These are years of good GOES-East coverage of the Mississippi Valley. We sought substantial falls of rain and a large gradient in soil moisture. A check of issues of the Weekly Weather and Crop Bulletin led us to select the month of July in the year 1988.

The largest area CST can process on McIDAS is 256 lines by 512 elements. We chose to center an array of this size at 39°30'N, 94°30'W. At a nominal resolution of 4 km, the array covered the area shown in Figure 1.

The test of various versions of CST at first involved data covering the Amazon Basin. By means of these data we determined that the calibration adjustment was not effective and the new discriminant offered little improvement over the old.

Tests of CST then shifted to the Mississippi Valley and CST4. A comparison of CST1 and CST4 for one day (17 July) showed fewer convective cores and less rain with CST4. Independently, scientists at GSFC also received encouraging results with this version. Hence CST4 was chosen for subsequent calculations of rain.

Rain was calculated in increments of days, beginning with 1 July. The calculation was preceded by a check of the quality of the infrared images. It was followed by a check of the rain images. For each day we calculated rain by both the ARK and CST4 techniques. The Appendix contains a protocol for this calculation and related steps.

In general the quality of the GOES images was high. This comment applies to the navigation (earth locations of pixels in images) as well as to the noise level of the data. Only five images (<1%) required a navigation adjustment.

Missing lines (dropouts) and "shot" noise were more serious problems. Shot noise manifests itself as excessively cold or warm pixels. Usually these appear in lines or bands. Shot noise (as well as garbled or "ropy" lines) is a matter of concern in any non-linear rain estimation technique. We found that a filter option on a McIDAS command (MA) which modifies areas effectively removed shot noise. However, it did not consistently remove extreme values along a ropy line.

Images containing ropy lines (or missing more than ten lines or six consecutive lines) were replaced by their nearest neighbor. Ten images (1%) were replaced. These ten resulted from many missing lines, individual ropy lines or both. A total of 24 images (3%) contained missing lines. Nine images (1%) contained shot noise. The table on page 4 breaks down these statistics by day.

TABLE: Summary of Images

DATE	INFRARED				DAILY RAINFALL			
July	Quality		Dups	Navn	ARK	C S T		
	dpts	nse		crtns		1	4	4W
1	0	0	0	0	X	X	X	X
2	0	0	0	1	X	X	X	X
3	0	1	0	0	X		X	X
4	2	1	0	0	X		X	X
5	2	0	0	0	X		X	X
6	0	0	0	0	X	X	X	X
7	0	1	0	4	X		X	X
8	2	0	2	0	X		X	X
9	0	1	1	0	X		X	X
10	2	0	0	0	X		X	X
11	1	0	0	0	X		X	X
12	0	0	0	0	X		X	X
13	1	0	1	0	X		X	X
14	0	1	0	0	X		X	X
15	1	0	0	0	X		X	X
16	4	0	2	0	X		X	X
17	0	0	0	0	X	X	X	X
18	2	0	2	0	X		X	X
19	1	1	0	0	X		X	X
20	2	0	0	0	X		X	
21	0	1	0	0	X		X	
22	0	0	0	0	X		X	
23	1	0	0	0	X		X	
24	0	0	0	0	X		X	
25	0	0	0	0	X		X	
26	0	0	0	0	X		X	
27	0	0	0	0	X		X	
28	0	1	2	0	X		X	
29	2	0	0	0	X		X	
30	1	0	0	0	X		X	
31	0	1	0	0	X		X	

Notes: (1) dups, duplications; Navn crtns, Navigation corrections; dpts, dropouts; nse, noise. (2) X under Daily Rainfall marks those days for which digital daily-rain images were created and stored on magneto/optical disks.

The three most serious problems encountered in the processing were revisions to CST, an error in the image sum code and demands on the work station. Each change to the code for CST involved sometimes lengthy tests. At times in the process of making changes the code would break. The most serious instance of "broken code" involved an element parameter. Set to 256 rather than 512, this parameter precipitated unpredictable overflows in a cloud array as well as a system-write error. More seriously, the vagrant parameter meant that CST4 produced a rain map only for the western half of the domain (i.e., elements 1 through 256). This error was discovered only after the first 21 days had been processed. It has been corrected.

The code (IMGPRO) which sums individual rain images borrows navigation and directory information from the source images (two, on the PC McIDAS). It does not, however, get the navigation information from the same image (number two) which provides the directory information. Thus, insofar as navigation is concerned, the directories of all sum images are incorrect.

The correction for incorrect directories of the sum images is straightforward. It is explained in Appendix C.

The third problem arose from the nature of the processing. Although the code for CST4 runs on a PS/2 Model 70 McIDAS work station, the complexity of the calculations heavily taxes the computer. Processing a single 256 by 512 image takes about three minutes of clock time. Because the program which sums rain images is nearly as intensive, even in the absence of other tasks the PS/2 work station takes almost two hours to produce one CST daily rain map.

Results

Samples of rain images are given in Figures 2 through 6. All are for 17 July. Figures 2 and 3 are, respectively, ARK and CST4 images of rain made from the 0901 UTC infrared image (shown in Figure 1). Because of limitations in the workstation display, stratiform rain cannot be seen in Figure 3. Nevertheless, the figures make plain the differences inherent to these two techniques.

The sum of CST4 rain for the six hours ending at 1101 UTC is shown in Figure 4. Cores are beginning to overlap. Figures 5 and 6 show, for ARK and CST4, respectively, rainfall summed over the 24 h ending at 2301 UTC. Enhancements are identical. In gross form, the patterns are similar. In detail they are sharply different.

For the box in Figure 1 (also see Figures 5 and 6) in Figure 7 we show temperature (part a) and rainfall (parts b-f). Part b indicates that CST4 assigned a core rain of 21 mm/h to the cold cloud which encroaches on the northwest corner of the array in part a. The core is narrowly bordered by stratiform rain (2 mm/h). Over most of the array CST4 assigned a rainrate of zero.

By design, ARK can assign only two rates. Over the northwest half of the array, the rate is 3 mm/h (part c). Over the southeast corner, it is zero.

Through the 6 h period embracing part b, CST4 assigned additional falls of rain over the northwest corner of the array (part d). Still more cores were recorded in the course of the day (part e). Thus, the locus of heaviest rain shifted from the northwest corner to the western border and centers of heavy rain appear elsewhere in the array. Still, a few pixels in the southeast corner remain dry. The gradient of rain across the array is nearly 70 mm. ARK has also assigned the heaviest rain to the western border (part f). No pixel is dry; the gradient is about 22 mm.

The right half of the table on page 4 summarizes processing for the month. We ran CST4 on 31 days. We saved rain images from the truncated version of CST4 (which produced a rain map only for the west half of the domain) for 19 days (1 through 19 July). We saved CST1 rain images for four days (1 and 2 July; 6 July; and 17 July). ARK rain images exist for every day.

CST1 images are included as a benchmark. CST4-West images are included in the digital record for their possible value in examining the question of how the size of the domain affects the CST estimate of rainfall.

The digital record includes the following: (1) All infrared images subjected to rain processing; (2) all daily rain images listed in the table; (3) six-hour accumulations for all days of CST4 and some or all of ARK, CST1 and CST4W; and (4) hourly images for all days of CST4 and some or all of ARK, CST1 and CST4W. These data are recorded on a magneto/optical disk, APIRAIN, which is to be forwarded to GSFC with this report. A copy of this data set will be retained at SSEC. Daily rain images for ARK and the three versions of CST also exist on 3 1/2 inch diskettes, at SSEC.

The disk, APIRAIN, has two sides, A and B. All infrared images are on side A, in directory INFRARED. Rain images through 11 July are recorded on side A; those from 12 through 31 July are recorded on side B. All rain images are in directory RAIN.

Both directories are organized by day. Within each day-subdirectory in INFRARED are 24 files, each containing one infrared image. These files are named AREA02XX or AREA02YY, where XX is a number from 01 through 24 and YY is a number from 25 through 48. Images appear chronologically and XX and YY are interchangable. Thus the files AREA0201 and AREA0225 always contain infrared images for 0001 UTC.

Within each day-subdirectory in RAIN are sub-subdirectories for techniques. The number of technique subdirectories varies from two to four, according to the pattern indicated in the table on page 4. As in the day-subdirectories of RAIN, the technique subdirectories contain files which are organized chronologically. These files are named AREA0MNN. M is 2 for CST1, 3 for ARK and 4 for both versions of CST4. NN assumes values between 51 and 99. If they are present, the individual hourly rain images correspond to the range 51 through 74. Images of rain accumulated over 6 h correspond to 93, 95, 97 and 99. The daily rain image is 91. Thus the path to the 0201 CST4 rain image for 3 July is \RAIN\03JULY\CST4\AREA0453; the path to the ARK daily rain image for 30 July is \RAIN\30JULY\ARK\AREA0391.

CONCLUSIONS

This project has demonstrated the feasibility of running the Convective/Stratiform technique on a synoptic scale. The case chosen appears to be well suited for testing the feasibility of monitoring soil moisture by means of CST. Preliminary comparisons of CST and ARK indicate significant differences in estimates of rain amount and distribution.

REFERENCES

Adler, R. A., and A. J. Negri, 1988: A satellite infrared technique to estimate tropical convective and stratiform rainfall. J. Appl. Meteor., 30-51.

Arkin, P. A., 1979: The relationship between fractional coverage of high cloud and rainfall accumulations during GATE over the B-scale array. Mon. Wea. Rev., 107, 1382-1387.

Dengel, R., J. Benson, R. Lord, J. Plutchak, D. Santek, S. Shirshac, T. Whittaker and J. T. Young, 1989: An OS/2 implementation of a PC-based McIDAS workstation. 5th International Conf. Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, Anaheim, California, 30 Jan-3 Feb 1989, American Meteorological Society.

Ide, J., 1988: The UNIDATA PC-McIDAS Workstation--a technical discussion. Preprint Volume, 4th International Conf. Meteorology, Oceanography and Hydrology, Anaheim, California, American Meteorological Society.

Janowiak, J. E., and P. A. Arkin, 1991: Rainfall variations in the tropics during 1986-1989, as estimated from observations of cloud-top temperature. J. Geophys. Res., 96, 3359-3373.

Weekly Weather and Crop Bulletin. NOAA/USDA Joint Agricultural Weather Facility, Room 5844 USDA South Building, Washington, D.C. 20250. ISSN 0043-1974.

LEGENDS

1. GOES infrared image for 090100 UTC on 17 July 1988. The graphic across the image indicates the geographical area covered by this project. The box in Oklahoma, along the border with Kansas, indicates the location and dimensions of the arrays presented in Figure 7.

2. ARK rain image corresponding with Figure 1. Rain is indicated by the lighter tone; everywhere, it is 3 mm/h.

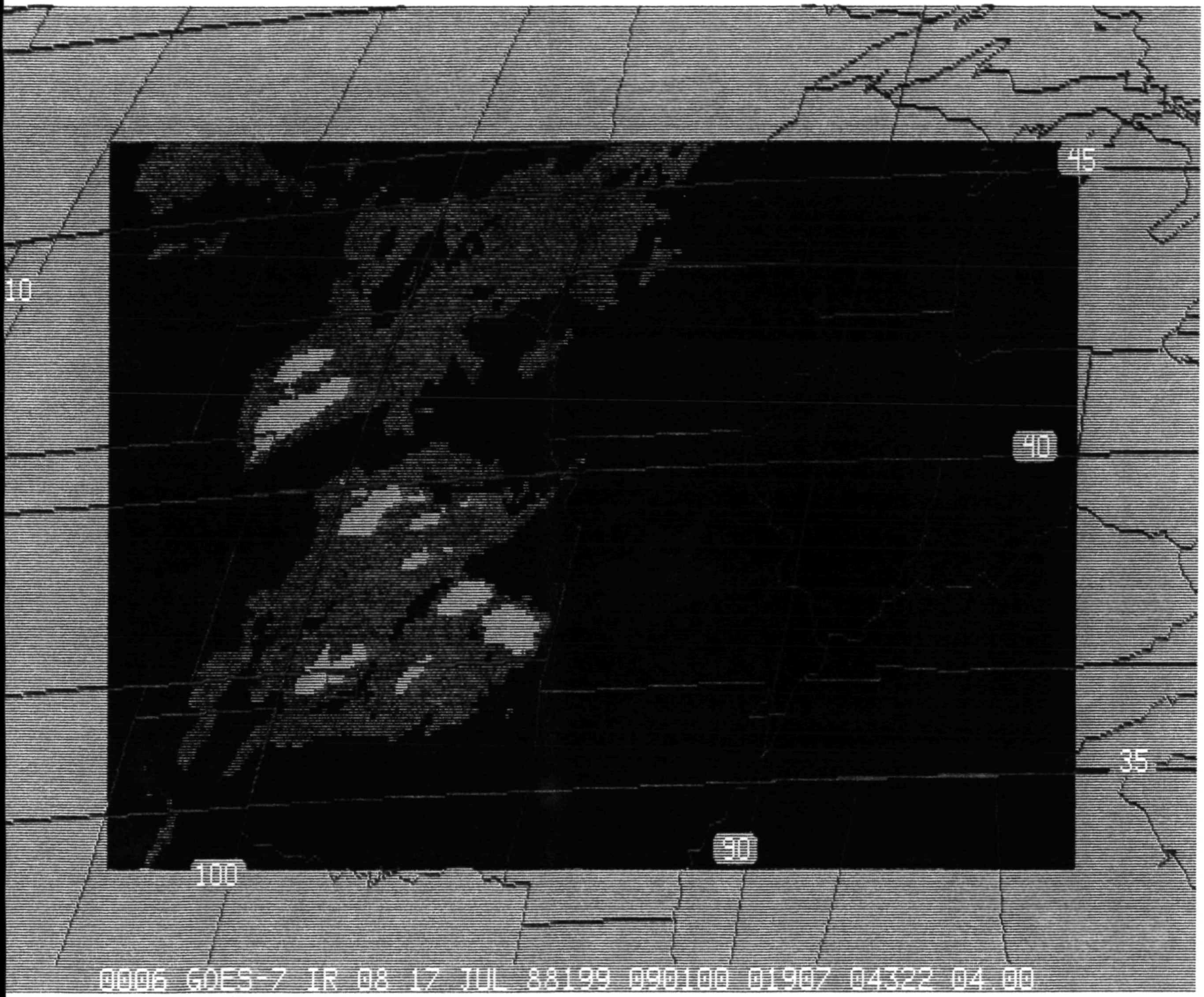
3. CST4 rain image corresponding with Figure 1. In this enhancement only rainrates greater than 3.9 mm/h can be distinguished from the background (zero) rate.

4. CST4 rain image covering the 6 h period ending at 110100 UTC.

5. As for Figure 4, except 24 h ending at 230100 UTC. Only rainrates greater than 29.9 mm/24 h can be distinguished from the background rate.

6. As for Figure 5, except ARK. Figure 5 and Figure 6 enhancements are identical.

7. Arrays of pixel values, in physical units, for the box shown in Figure 1 (also see Figures 5 and 6). Values are given in tenths, i.e., 2147 should be read as two-hundred-fourteen-point-seven. Part a: Temperature, in K; Part b: ARK rainrate for the 090100 UTC image, in mm/h; Part c: CST4 rainrate for the 090100 UTC image, in mm/h; Part d: CST4 rainrate for the 6 h ending at 110100 UTC, in mm/6h; Part e: CST4 rainrate for the 24 h ending at 230100 UTC, in mm/24h; Part f: ARK rainrate for the 24 h ending at 230100 UTC, in mm/24h.



45
10
40
35
90
100
0006 GOES-7 IR 08 17 JUL 88199 090100 01907 04322 04.00

Figure 1

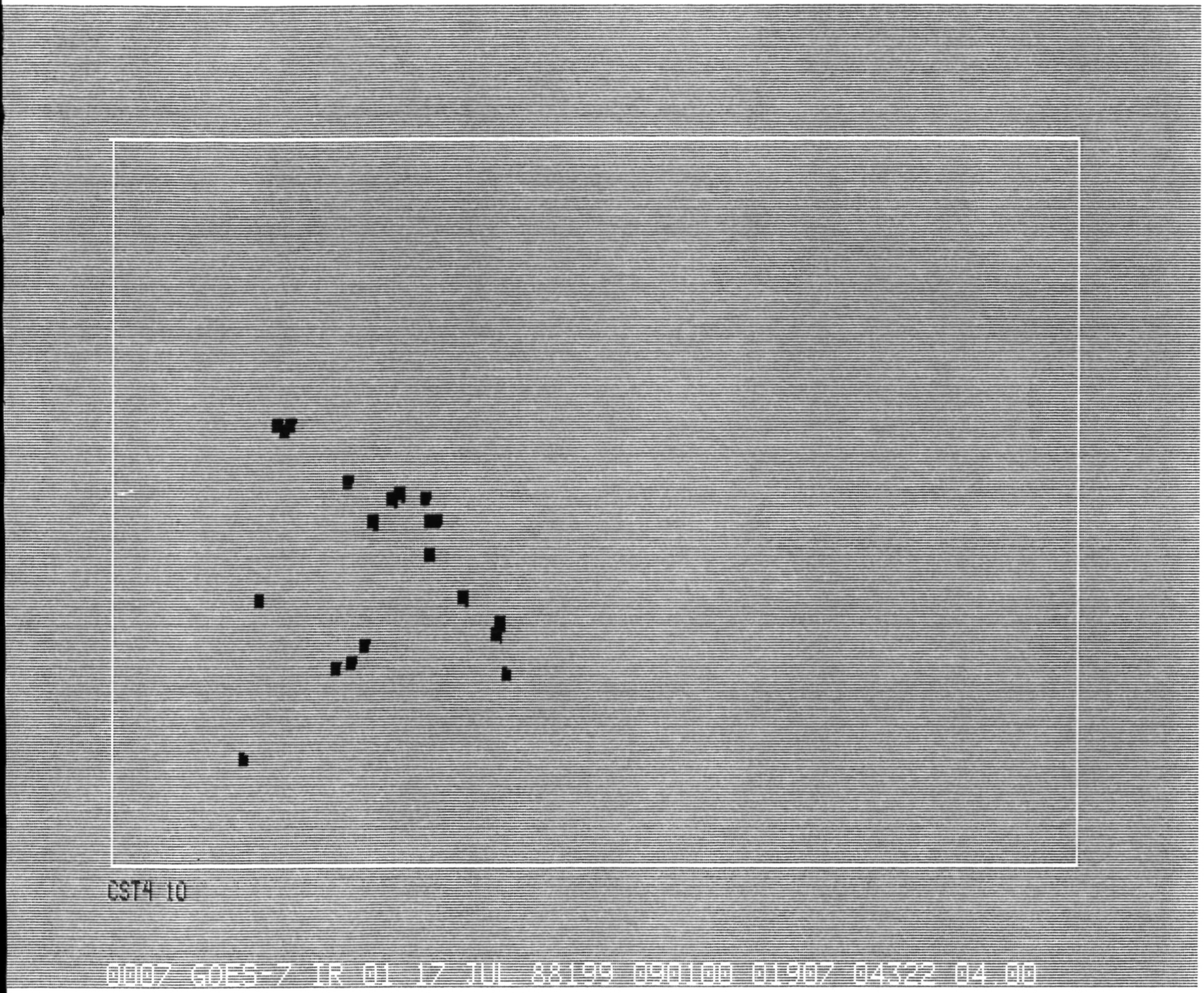


Figure 2

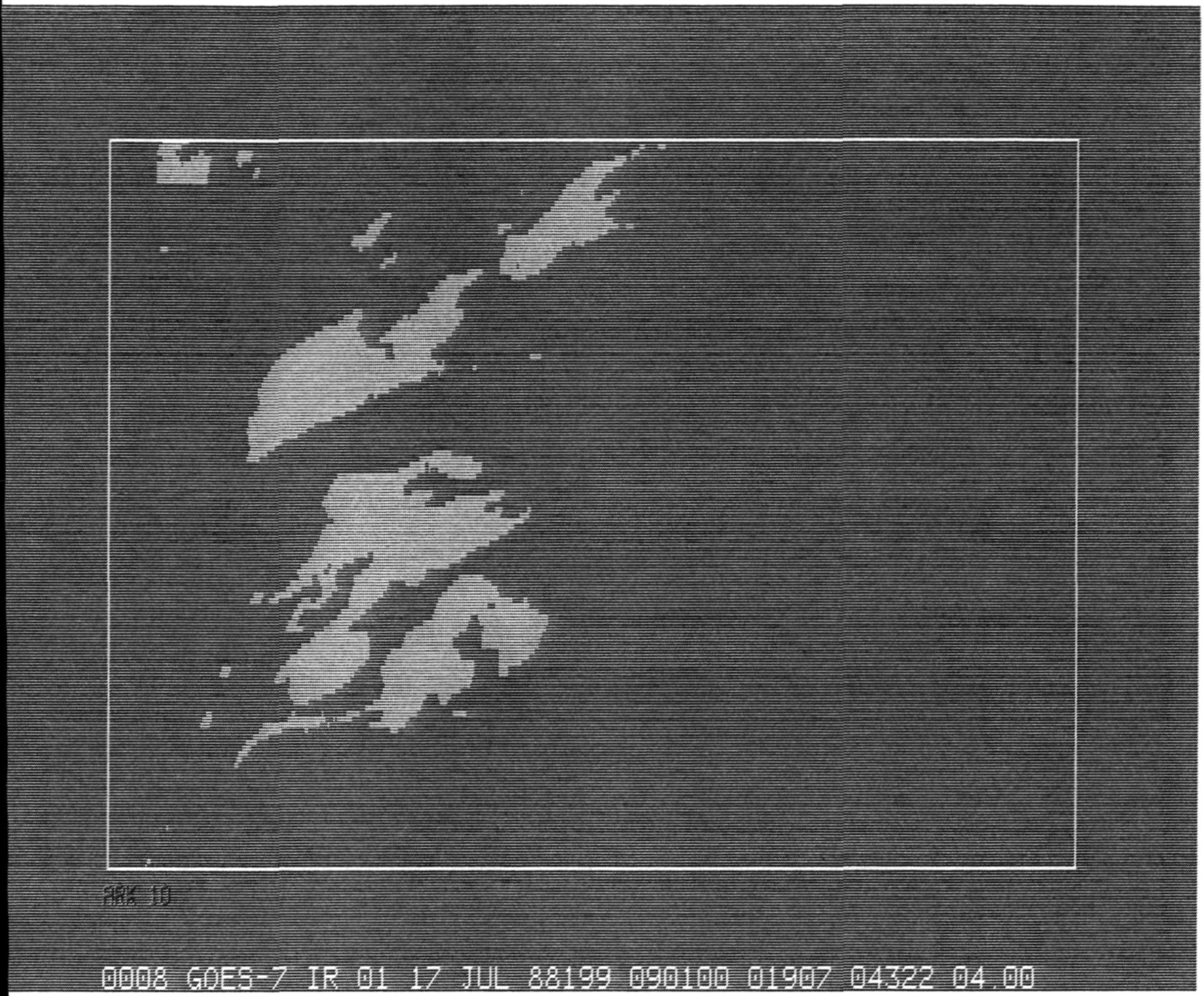


Figure 3

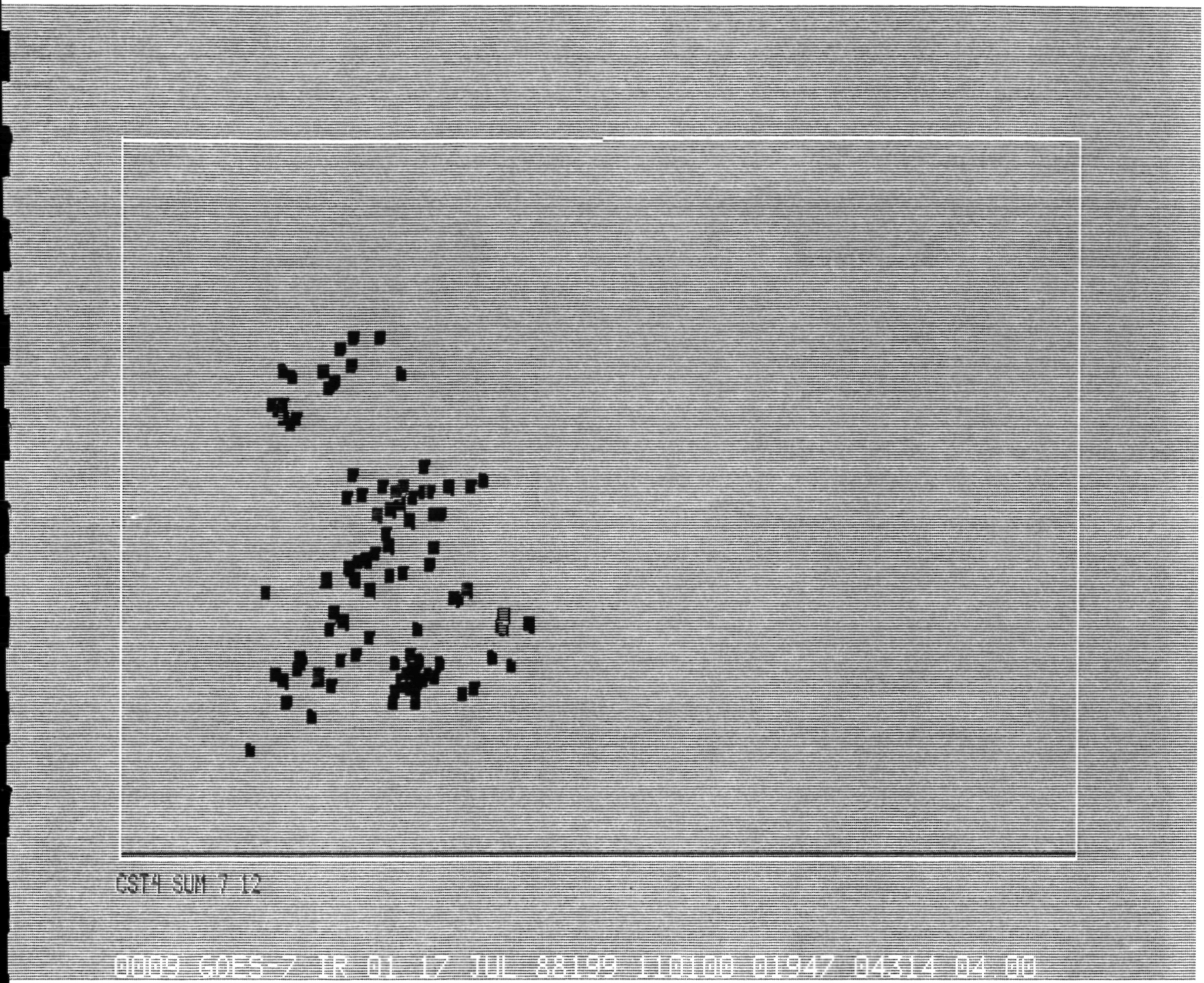

CST4 SUM 7 12
0009 GOES-7 IR 01 17 JUL 88199 110100 01947 04314 04 00

Figure 4

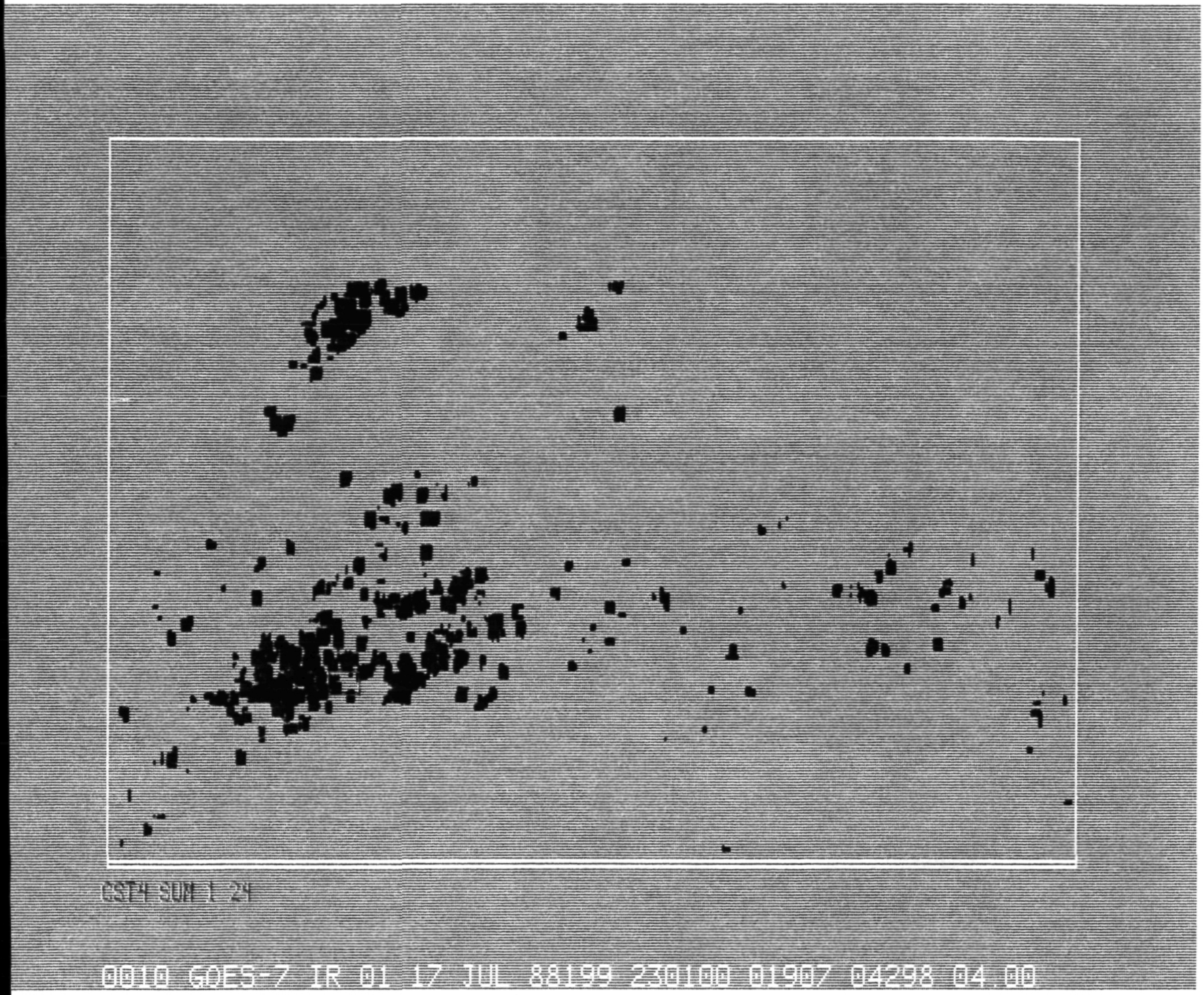

CST4 SUM 1 24
0010 GOES-7 IR 01 17 JUL 88199 230100 01907 04298 04.00

Figure 5

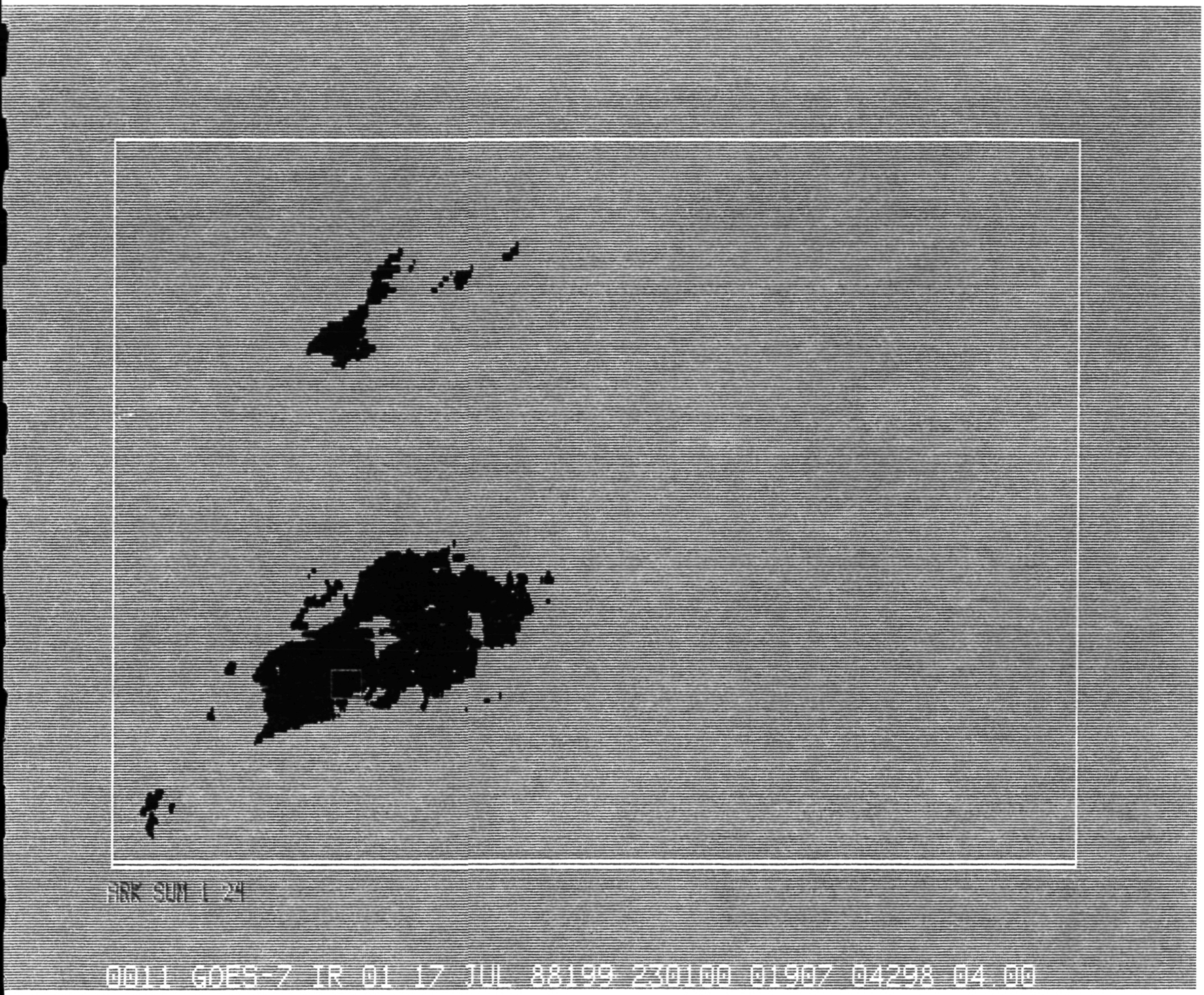

ARK SUM L 24
0011 GOES-7 IR 01 17 JUL 88199 230100 01907 04298 04.00

Figure 6

a

2147	2151	2155	2159	2171	2182	2190	2205	2219	2237	2267	2302	2344	2386	2418
2186	2190	2190	2186	2182	2197	2208	2233	2260	2296	2344	2400	2441	2461	2466
2186	2190	2190	2186	2182	2197	2208	2233	2260	2296	2344	2400	2441	2461	2466
2226	2226	2226	2233	2257	2293	2326	2364	2394	2423	2446	2466	2483	2500	2511
2226	2226	2226	2233	2257	2293	2326	2364	2394	2423	2446	2466	2483	2500	2511
2311	2323	2350	2394	2451	2500	2538	2556	2552	2534	2520	2516	2520	2518	2514
2311	2323	2350	2394	2451	2500	2538	2556	2552	2534	2520	2516	2520	2518	2514

b

210	210	210	20	20	20	20	0	0	0	0	0	0	0	0
20	20	20	20	20	0	0	0	0	0	0	0	0	0	0
20	20	20	20	20	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

c

30	30	30	30	30	30	30	30	30	30	30	30	30	0	0
30	30	30	30	30	30	30	30	30	30	30	0	0	0	0
30	30	30	30	30	30	30	30	30	30	30	0	0	0	0
30	30	30	30	30	30	30	0	0	0	0	0	0	0	0
30	30	30	30	30	30	30	0	0	0	0	0	0	0	0
30	30	30	0	0	0	0	0	0	0	0	0	0	0	0
30	30	30	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7

d

78	78	78	78	78	39	39	39	39	20	20	0	0	0	0
78	78	78	78	78	39	39	39	20	0	0	0	0	0	0
39	39	39	20	20	20	20	20	0	0	0	0	0	0	0
39	39	39	20	20	20	0	0	0	0	0	0	0	0	0
20	20	20	20	20	20	0	0	0	0	0	0	0	0	0
20	20	20	0	0	0	0	0	0	0	0	0	0	0	0
20	20	20	0	0	0	0	0	0	0	0	0	0	0	0

e

157	157	157	157	133	94	290	290	267	447	447	227	243	424	424
345	345	322	322	133	78	78	78	63	227	227	212	227	439	424
306	306	282	78	78	63	63	63	235	227	227	227	227	439	227
698	667	667	455	643	267	227	227	235	235	235	235	220	24	24
682	643	643	455	643	267	227	227	235	235	220	220	204	0	0
643	643	643	620	620	212	212	212	220	220	204	204	204	0	0
455	455	455	620	620	212	188	188	612	384	384	384	204	0	0

f

431	431	431	431	431	431	431	431	431	463	463	408	408	408	376
431	463	463	463	463	463	463	463	431	431	431	345	369	369	314
431	463	463	463	463	463	463	408	376	376	376	345	369	369	314
494	494	463	463	463	463	431	400	400	400	369	369	369	369	369
494	494	463	369	369	369	337	369	369	369	337	306	306	306	306
463	463	463	369	369	369	369	369	369	337	337	306	306	306	306
376	376	376	369	369	369	369	369	369	306	275	275	275	275	275

Figure 7 (completed)

APPENDICES

A. Protocol

API
Image Processing

This is a guide to creating images of daily rainfall for the antecedent rainfall project (1345).

I. Resources
Nine-track, 6250 bpi tape data (APIL1, APIL2, ...; in desk under work station)
UFPa's work station (room 515; address is 128 104 83 32)
Programs: CST4 and ARK; IMGPRO; McIDAS operational software
Areas
mainframe (10304-10351; 7008-7023)
w/s (201-500)
Directory: C:\MCIDAS\DATA

II. Procedure
A. Given, dayx and dayx+1 images on the mainframe; neither one processed.

B. Workstation
1. Power up.
Switch on magneto-optical drive.
Switch on system unit.
Switch on monitor.
Select McIDAS session.

2. Check ir areas.

```
LA 201 224
```

If McIDAS responds, "The E: device is not ready", enter

```
REDIRECT CLEAR
```

If dayx images are not present, go to (3). If present, skip to (6).

3. Log on to mainframe.

```
.LOGON initials 1345 X password
```

4. Load infrared images onto the work station. Reserved blocks are 201-224 or 225-248.

If images are on the mainframe (areas 10304-10327) and are to go to areas 201-224, enter

```
BATCH "SENALL1.BAT
```

If images are to go to areas 225-248, enter

```
BATCH "SENALL2.BAT
```

If images are on a M/O cartridge, load that cartridge. Select OS/2 full screen session. Switch to E: drive and check directory. Find file containing images of interest. Enter

```
COPY *.* C:\MCIDAS\DATA
```

5. Save the ir images on the magneto-optical disk, APIIR. If in McIDAS session, enter

```
OS "COPY AREAxxxx E:\datedir\techniquedir\AREAxxxx
```

where xxxx is the four-digit number of the rain image area on the work station. If in OS/2 session, type

```
COPY C:\MCIDAS\DATA E:\datedir\techniquedir\AREAxxxx
```

6. Check images.

```
LA area1 areaN
```

```
DF area1 areaN
```

```
MAP
```

If more than 10 lines or more than 6 consecutive lines are missing, or if any line is garbled, quit the image and replace it by the nearest neighbor.

```
QA area
```

```
AA sarea darea frame X X X X 256 512
```

If less than these, accept as is. If shot noise is present, filter the image.

```
MA sarea darea frame BAD
```

```
DF sarea frame
```

If the image is offset more than 50 lines or 50 elements from the map, quit the image. If less than 50 but more than 4 lines or 4 elements from the map, measure the offset in lines and/or elements and make the correction through the image definition. (The rules are as follows: image up, subtract lines; image left, subtract elements.)

```
CA area X X X line ele
```

```
DF...
```

If more than 6 images total or 4 consecutive are missing, declare the day to be null and skip to step 16. If 1 to 6 (total) or 1 to 4 (consecutive), repeat rain images on either side of the missing images to fill the gap. Note the number of missing images.

On the image logsheets record any corrections or substitutions.

7. Run rain program ARK.

```
BATCH ARK "RAINn.BAT
```

This command activates a batch file containing 24 repetitions of a radiance-to-rainrate command. "n" is 2 if the infrared images are in areas 201-224; 6, if in 225-248. The command in the batch file is

```
    ARK sarea darea NAME=ARK LAT=35 47 LON=84 110 "ARK-N
```

N is the number of the image (1,2,...24), where 1 corresponds with the 0001 UTC image, 2 with 0101, etc.

This command also activates a batch file containing 24 repetitions of a command sums rainrate. The command in the batch file is

```
IMGPRO X darea AREA=sareai sareaj BAND=1 1 FACTOR=10 10 10
SCALE=0 nn 0 255 ZERO=DATA DATA UNIT=algorithm NAME=SUM
QA=YES "SUM sareai sareaj
```

On a pixel-by-pixel basis this command divides rain rates in source areas i and j by ten, sums them, multiplies the sum by 10, places the result in destination area k, scales eight bit digital count to correspond with the range of summed rainrate, tells the program to use the calibration for algorithm and labels area k as "SUM algorithm".

8. Check daily rain image. Display image and (if needed) list values.

```
DF...
OD
OD L PROD
```

Saturation? Striping? If image is bad, return to II.B.6.

9. Return to II.B.7 and run CST4.

10. Go to OS/2 full screen session.

11. Save daily rain image on a diskette.

```
COPY C:\MCIDAS\DATA\AREA0H91.XXX A:
```

H is 3 for ARK; 4 for CST4. Call diskettes APIDR1, APIDR2, ...

12. Make a HP Paint Jet image 24 h rainfall. (1) Display images (if not on the monitor). (2) Enhance and label images. (3) Print enhanced images.

```
EU REST FOUR
PC T 280 12
ZA 1 6 "XXXX SUM 1 24
SCRDMP EPJ
```

Enhancement four emphasizes the contrast between light rain falls and rain-free areas. Thresholds are 0 9 200 200 0 200 0 200.

13. Save all hourly and six-hourly rain images and the daily rain image on a second magneto/optical disk, APIR.

Insert disk. Switch to E-drive and check directory. (Directories are organized by days and techniques; e.g., 01JULY\CST4.) Go to appropriate directory. Switch back to C-drive and enter copy commands.

```
COPY AREA045* E:
COPY AREA046* E:
COPY AREA047* E:
COPY AREA0493 E:
COPY AREA0495 E:
COPY AREA0497 E:
COPY AREA0499 E:
COPY AREA0491 E:
```

Check destination directory.

14. Repeat (7)-(13) for CST4.

15. Download dayx+1 images (i.e., repeat steps from II.B.3 for dayx+1).

16. Log out.

```
.LOGOFF
```

C. Mainframe

Submit area-restore request. Replace dayx images with dayx+2 images; dayx+1 images (if processed) by dayx+3 images.

APPENDIX B. Code for CST4

```
      SUBROUTINE MAINO
C
C
C ? CST4 - hourly rain rates (ORIGINAL Adler&Negri,JA M 1988)  (BMG)
C ?
C ?   CST4 sarea darea <keywords> "text...
C ?
C ? Parameter(s):
C ?   sarea  |  input  area number (default=current frame)
C ?   darea  |  output area number (no default)
C ?   "text  |  output area directory memo, up to 32 chars
C ? Keyword(s):
C ?   BAND= satellite band/channel number (default=8)
C ?   NAME= rain rate algorithm name, up to 4 chars (default=CST4)
C ?   BLMT= brightness count threshold (default=183)
C ?   RLMT= rain rate threshold and units (default=50 mm/hr)
C ?   FACTOR= input and output calibration internal scaling factor
C ?           (default=1.0 10.0)
C ?   SCALE= (prodlo) (prodhi) (britlo) (brithi) values for calibration
C ?          codicil to scale output rain rates into brightness counts
C ?          (default=0.0 RLMT 0 255)
C ?   LAT= (min) (max) latitude limits (default=current frame)
C ?   LON= (min) (max) longitude limits (default=current frame)
C ? Remarks:
C ?   1) Maximum size limits are 256 lines and 512 elements
C ?   2) Rain rates are stored as tenths of mm/hr
C ?   3) The ouput calibration internal scaling factor can be used to
C ?      adjust the rain rates for the difference between 1 hr and the
C ?      sampling interval of a sequence of images to be summed (e.g.
C ?      using 0.5hr imagery to produce a daily sum will require the
C ?      addition of 48 images instead of 24 1hr images, which will
C ?      result in daily totals which are too large by a factor of two
C
C REMARK:
C    1) INTERNAL SCALING FACTOR DEFAULT IS SET FOR GOES BRIGHTNESS
C       TEMPERATURES WHICH ARE STORED AS SCALED INTEGERS WITH A TENTHS
C       PLACE PRECISION (I.E. THEY ARE STORED AS 4-DIGIT INTEGERS).
C       THIS WILL RESULT IN A TENTHS PLACE PRECISION BEING APPLIED TO
C       THE INTERNAL SCALING OF THE RAIN RATE VALUES (I.E. THE RAIN
C       RATES WILL BE STORED AND DISPLAYED IN UNITS OF 0.1 MM/HR)
C
      IMPLICIT INTEGER (A-B,E-W)
      IMPLICIT CHARACTER*12 (C)
      IMPLICIT REAL*8 (D)
      IMPLICIT REAL*4 (X-Z)
      PARAMETER ( MAXE=512,MAXL=256 )
      DIMENSION IARR(MAXE,MAXL),OSEC(MAXE,MAXL)
      DIMENSION IOUT(MAXE)
      DIMENSION IDIR1(64),IDIR2(64),ICALB(128)
      CHARACTER*160 CTEXT
      CHARACTER*80 CWRITE
C     INCLUDE 'MXCDSZ.INC'
C=========================== DIMXCDSZ ============================

C $   (JR)
C $   THIS INCLUDE FILE IS PART OF THE NAVIGATION SUBSYSTEM
C $
C $   DESCRIPTION:
```

```
C $   DESCRIPTION:
C $      THIS IS THE VALUE OF LARGEST NAVIGATION CODICIL SIZE ALLOWED
C $
C $$  DIMXCDSZ = INCLUDE, NAVIGATION

C DEFINITIONS OF PARAMETERS
C MXCDSZ   : LARGEST ALLOWED CODICIL SIZE

      INTEGER      MXCDSZ

      PARAMETER (MXCDSZ = 5*128)

C======================= END DIMXCDSZ ===========================
      DIMENSION NAVARR(MXCDSZ)
C
C... BEGINNING OF PROGRAM
      CBUG=CKWP('DEBUG',1,'NO')
      CALL SDEST('CST4 --- BEGIN',0)
C
C... INPUT SOURCE AND DESTINATION AREA NUMBERS
      IA1 = IPP(1,0)
      IF (IA1.EQ.0) THEN
         CALL SDEST('An Input Area Must Be Specified',0)
         GO TO 999
      ENDIF
      IA2 = IPP(2,0)
      IF (IA2.EQ.0) THEN
         CALL SDEST('An Output Area Must Be Specified',0)
         GO TO 999
      ENDIF
      IF (IA1.EQ.IA2) THEN
         CALL SDEST('Input/Output Areas Must Be Different',0)
         GO TO 999
      ENDIF
C
C... OPEN SOURCE AREA
      CALL OPNARA(IA1)
      CALL ARAOPT(IA1,1,'SPAC',4)
      CALL ARAOPT(IA1,1,'UNIT',LIT('BRIT'))
C
C... READ SOURCE AREA DIRECTORY
      CALL READD( IA1, IDIR1 )
      IF (IDIR1(1).LT.0) THEN
         CALL SDEST('Input Area Does Not Exist ',IA1)
         GO TO 999
      ENDIF
C
C--- SETUP AREA COORD TRANSFORMATION INFORMATION FROM DIRECTORY ENTIRES
      LCOR = IDIR1(6)
      ECOR = IDIR1(7)
      LSIZ = IDIR1(9)
      ESIZ = IDIR1(10)
      LRES = IDIR1(12)
      ERES = IDIR1(13)
      MAXLIN = LCOR + ((LSIZ-1)*LRES)
      MAXELE = ECOR + ((ESIZ-1)*ERES)
      IZSIZ = IDIR1(11)
      IZRES = IDIR1(14)
      IDATE = IDIR1(46)
      ITIME = IDIR1(47)
      IF (IDATE.EQ.0) THEN
         IDATE = IDIR1(4)
         ITIME = IDIR1(5)
      ENDIF
C
C... GET THE PROPER NAVIGATION FILE.
      IF(NVSET('AREA',IA1).NE.0) THEN
```

```
   [illegible]
         CALL SDEST ('Unable to Get Navigation for Area ',IA1)
         GO TO 999
      ENDIF
C... SET DEFAULT LINE/ELEMENT AND LAT/LON LIMITS
      XLIN1=FLOAT(LCOR)
      XLIN2=FLOAT(MAXLIN)
      XELE1=FLOAT(ECOR)
      XELE2=FLOAT(MAXELE)
      ISTAT=NV1SAE(XLIN1,XELE1,XDUM,XLATNW,XLONNW,0.0)
      ISTAT=NV1SAE(XLIN2,XELE2,XDUM,XLATSE,XLONSE,0.0)
      IF (CBUG(1:1).EQ.'Y') THEN
         CWRITE='LAT= '//CFG(XLATNW)//CFG(XLATSE)
         CALL SDEST(CWRITE,0)
         CWRITE='LON= '//CFG(XLONNW)//CFG(XLONSE)
         CALL SDEST(CWRITE,0)
      ENDIF
C... OVERRIDE DEFAULT VALUES FROM KEYWORD PARAMETERS
      IF (NKWP('LAT').EQ.0.OR.NKWP('LON').EQ.0) GO TO 777
      XLATSE=SNGL(DKWPLL('LAT',1,XLATSE))
      XLATNW=SNGL(DKWPLL('LAT',2,XLATNW))
      XLONSE=SNGL(DKWPLL('LON',1,XLONSE))
      XLONNW=SNGL(DKWPLL('LON',2,XLONNW))
C... NAVIGATE IMAGE LIN/ELE VALUES CORRESPONDING TO THE LAT/LON LIMITS
      ISTAT=NV1EAS(XLATNW,XLONNW,0.0,XLIN1,XELE1,XDUM)
      ISTAT=NV1EAS(XLATSE,XLONSE,0.0,XLIN2,XELE2,XDUM)
      ISTAT=NV1EAS(XLATSE,XLONNW,0.0,XLIN3,XELE3,XDUM)
      ISTAT=NV1EAS(XLATNW,XLONSE,0.0,XLIN4,XELE4,XDUM)
C... SELECT MIN AND MAX IMAGE LIN/ELE LIMITS
      XLIN1=AMIN1(XLIN1,XLIN2,XLIN3,XLIN4)
      XLIN2=AMAX1(XLIN1,XLIN2,XLIN3,XLIN4)
      IF (XLATNW*XLATSE.GT.0.0) THEN
C... AREA TO BE GRIDDED DOES NOT CROSS EQUATOR
         XELE1=AMIN1(XELE1,XELE2,XELE3,XELE4)
         XELE2=AMAX1(XELE1,XELE2,XELE3,XELE4)
      ELSE
C... AREA TO BE GRIDDED DOES CROSS EQUATOR
         ISTAT=NV1EAS(0.0,XLONNW,0.0,XLINE,XELE1,XDUM)
         ISTAT=NV1EAS(0.0,XLONSE,0.0,XLINE,XELE2,XDUM)
      ENDIF
C... CONVERT IMAGE LIN/ELE COORDS INTO TV LIN/ELE CORRDS
777   CONTINUE
      L1=1+(XLIN1-FLOAT(LCOR))/FLOAT(LRES)
      L2=1+(XLIN2-FLOAT(LCOR))/FLOAT(LRES)
      E1=1+(XELE1-FLOAT(ECOR))/FLOAT(ERES)
      E2=1+(XELE2-FLOAT(ECOR))/FLOAT(ERES)
C... TEST THAT LIN/ELE LIMITS ARE WITHIN AREA LIN/ELE BOUNDS
      IF (L1.LT.1) L1=1
      IF (L2.GT.LSIZ) L2=LSIZ
      IF (1+(L2-L1).GT.MAXL) L2=L1+(MAXL-1)
      IF (E1.LT.1) E1=1
      IF (E2.GT.ESIZ) E2=ESIZ
      IF (1+(E2-E1).GT.MAXE) E2=E1+(MAXE-1)
C... SET ELEMENT LENGTH FROM OFFSETS
      LN=(L2-L1)+1
      EN=(E2-E1)+1
      IF (CBUG(1:1).EQ.'Y') THEN
         CWRITE='LAT= '//CFG(XLATNW)//CFG(XLATSE)
         CALL SDEST(CWRITE,0)
         CWRITE='LON= '//CFG(XLONNW)//CFG(XLONSE)
         CALL SDEST(CWRITE,0)
         CWRITE='SATLIN= '//CFG(XLIN1)//CFG(XLIN2)
         CALL SDEST(CWRITE,0)
         CWRITE='SATELE= '//CFG(XELE1)//CFG(XELE2)
         CALL SDEST(CWRITE,0)
         CWRITE='TVLIN='//CFI(L1)//CFI(L2)//CFI(LN)
         CALL SDEST(CWRITE,0)
```

```
            CALL SDEST(CWRITE,0)
            CWRITE='TVELE='//CFI(E1)//CFI(E2)//CFI(EN)
            CALL SDEST(CWRITE,0)
         ENDIF
C
C... INPUT THRESHOLD LIMITS
         IRLMT = IKWP('RLMT',1,50)
         CRUNI = CKWP('RLMT',2,'mm/h')
C
C... INPUT OUTPUT AREA CALIBRATION LIMITS
         XSI  = SNGL(DKWP('FACTOR',1,1.0D0))
         XSF  = SNGL(DKWP('FACTOR',2,10.0D0))
         XNLO = SNGL(DKWP('SCALE',1,0.0D0))*XSF
         XNHI = SNGL(DKWP('SCALE',2,DFLOAT(IRLMT)))*XSF
         XBLO = SNGL(DKWP('SCALE',3,0.0D0))
         XBHI = SNGL(DKWP('SCALE',4,255.0D0))
         XSCA = (XBHI-XBLO)/(XNHI-XNLO)
C
C... INPUT BAND NUMBER
         IBAND = IKWP('BAND',1,8)
C
C... READ INPUT INFRARED TEMPERATURES
         DO 120 NL = L1, L2
            NLIN=1+(NL-L1)
            CALL ZEROW(MAXE,IOUT)
            CALL REDARA(IA1,NL-1,E1-1,EN,IBAND,IOUT)
            DO 100 NE = E1, E2
               NELE=1+(NE-E1)
               IARR(NELE,NLIN)=IOUT(NELE)/XSI
100         CONTINUE
120      CONTINUE
CCC
         CALL CST4(IARR,OSEC,EN,LN,ISUMPX,ISUMRA)
CCC
C
C... CREATE DESTINATION AREA DIRECTORY
         CALL CQFLD(CTEXT)
         CALL MOVW(64,IDIR1,IDIR2)
         IDIR2(6) = NINT(XLIN1)
         IDIR2(7) = NINT(XELE1)
         IDIR2(9) = LN
         IDIR2(10) = EN
         IDIR2(11) = 1
         IDIR2(14) = 1
         IDIR2(15) = 0
         CALL GETDAY(IDIR2(17))
         CALL GETTIM(IDIR2(18))
         IDIR2(19) = 1
         CALL MOVCW(CTEXT(1:32),IDIR2(25))
         IDIR2(33) = IA2
         IDIR2(36) = 0
         IDIR2(49) = 0
         IDIR2(50) = 0
         IDIR2(51) = 0
         IDIR2(52) = LIT('PRD ')
         IDIR2(53) = LIT('BRIT')
C---PC
         IDIR2(34) = IDIR2(34) + 512
         IDIR2(63) = IDIR2(35) + 512
C---PC
C
C... MAKE AND OPEN OUTPUT AREA
         CALL MAKARA(IA2,IDIR2)
         CALL OPNARA(IA2)
C
C--- WRITE OUT RAIN RATE ARRAY AND CALCULATE AREAL BULK STATISTICS
         N = 0
```

```
      XSUM = 0.0
      LMAX = 0
      EMAX = 0
      XRMAX=0.0
      DO 220 NL=L1,L2
         NLIN=1+(NL-L1)
         CALL ZEROW(MAXE,IOUT)
         DO 200 NE=E1,E2
            NELE=1+(NE-E1)
            XRAIN = FLOAT(OSEC(NELE,NLIN))
            IF (XRAIN.GT.0.0) THEN
               N = N + 1
               XSUM = XSUM + XRAIN
            ENDIF
            IF (XRAIN.GT.XRMAX) THEN
               XRMAX = XRAIN
               LMAX = NL
               EMAX = NE
            ENDIF
            XRAIN = XRAIN*XSF
            IRAIN = NINT(XBLO+(XRAIN-XNLO)*XSCA)
            IOUT(NELE) = MIN0(MAX0(0,IRAIN),255)
 200     CONTINUE
         CALL PACK(ESIZ,IOUT,IOUT)
         CALL WRTARA(IA2,NLIN-1,IOUT)
 220  CONTINUE
C
C--- CLOSE INPUT AND OUTPUT AREAS
      CALL CLSARA(IA1)
      CALL CLSARA(IA2)
C
C--- COPY NAVIGATION CODICIL FROM INPUT AREA TO OUTPUT AREA
C---PC
CCC   CALL DELCOD(IA2)
CCC   CALL COPCOD(IA1,IA2)
         NAVSIZ = IDIR2(63) - IDIR2(35)
         CALL ARAGET(IA1,IDIR1(35),NAVSIZ,NAVARR)
         CALL ARAPUT(IA2,IDIR2(35),NAVSIZ,NAVARR)
C---PC
C
C--- WRITE CALIBRATION CODICIL FOR OUTPUT AREA
      CNAME = CKWP('NAME',1,'CST4')
      CALL ZEROW(128,ICALB)
      ICALB(1) = LIT(CNAME)
      ICALB(2) = NINT(XNLO)
      ICALB(3) = NINT(XNHI)
      ICALB(4) = NINT(XBLO)
      ICALB(5) = NINT(XBHI)
C---PC
      CALL ARAPUT(IA2, IDIR2(63), 512, ICALB)
CCC   CALL NXOLOK('AREA',IA2,LIT('CALB'),ICALB)
C---PC
      CALL STAMP(IA2)
      CALL SDEST('Output Area Written ',IA2)
C
C... DISPLAY RAIN RATE STATISTICS OVER IMAGE AREA
      KMS   = LRES*ERES
      LES   = LN*EN
      XATLN=FLOAT(LCOR+((LMAX-1)*LRES))
      XATEL=FLOAT(ECOR+((EMAX-1)*ERES))
      ISTAT=NV1SAE(XATLN,XATEL,XDUM,XLATMX,XLONMX,0.0)
      ISUMST=NINT(XSUM)-ISUMRA
      ISUMPS=N-ISUMPX
      CALL SDEST( ' ', 0 )
      CWRITE='Source      Area = '//CFU(IA1)
      CALL SDEST(CWRITE,0)
      CALL SDEST(CWRITE,0)
      CWRITE='Date    /   Time = '//CFU(IDATE)//CFU(ITIME)
      CALL SDEST(CWRITE,0)
      CWRITE='Destination Area = '//CFU(IA2)
      CALL SDEST(CWRITE,0)
      CALL SDEST( ' ', 0 )
      CALL SDEST( 'Rain Rate Summary Statistics:', 0 )
      CWRITE='Number of Pixels, Total       ='//CFI(LES)
      CALL SDEST(CWRITE,0)
      CWRITE='Number of Pixels, No Rainfall ='//CFI(LES-N)
      CALL SDEST(CWRITE,0)
      CWRITE='Number of Pixels, Rainfall    ='//CFI(N)
      CALL SDEST(CWRITE,0)
      CWRITE='Number of Pixels, Convective  ='//CFI(ISUMPX)
      CALL SDEST(CWRITE,0)
      CWRITE='Number of Pixels, Stratiform  ='//CFI(ISUMPS)
      CALL SDEST(CWRITE,0)
      CALL SDEST( ' ', 0 )
      CWRITE='Regional Coverage, Total      ='//CFI(LES*KMS)//' KM**2'
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Coverage, No Rainfall='//CFI((LES-N)*KMS)//
     &                                                      ' KM**2'
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Coverage, Rainfall   ='//CFI(N*KMS)//' KM**2'
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Coverage, Convective ='//CFI(ISUMPX*KMS)//
     &                                                       ' KM**2'
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Coverage, Stratiform ='//CFI(ISUMPS*KMS)//
     &                                                       ' KM**2'
      CALL SDEST(CWRITE,0)
      CALL SDEST( ' ', 0 )
      CWRITE='Regional Rain Rate, Rainfall  = '//CFG(XSUM)//CRUNI
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Rain Rate, Convective= '//CFI(ISUMRA)//CRUNI
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Rain Rate, Stratiform= '//CFI(ISUMST)//CRUNI
      CALL SDEST(CWRITE,0)
      CWRITE='Regional Rain Rate, Maximum   = '//CFG(XRMAX)//CRUNI
      CALL SDEST(CWRITE,0)
      CWRITE='                    Latitude  = '//CFG(XLATMX)
      CALL SDEST(CWRITE,0)
      CWRITE='                    Longitude = '//CFG(XLONMX)
      CALL SDEST(CWRITE,0)
C
C... END OF PROGRAM
      CALL SDEST( ' ', 0 )
 999  CALL SDEST( 'CST4 --- DONE', 0 )
      RETURN
      END
C-----
      SUBROUTINE CST4 (IARR,OSEC,IDIM,JDIM,ISUMPX,ISUMRA)
C
C       AREAS OF LOCAL MAXIMA IN COUNT (MINIMA IN TEMPERATURE)
C       ARE ASSUMED TO BE THUNDERSTORMS, AND IF A CIRRUS TEST IS PASSED,
C       THESE AREAS ARE CONVERTED TO RAINRATE AND RAINAREA VIA
C       ALDER AND MACK, JCAM 1984 SCHEME.
C
C       VARIABLE            DESCRIPTION
C
C       IARR     :   INPUT ARRAY (INFRARED DATA)
C       OSEC     :   OUTPUT ARRAY (RAINRATE)
C       IMAX     :   STORES THE COUNT VALUES OF THE MAXIMUM POINTS
C       ILOC     :   STORES THE I-CENTROID LOCATION OF THE MAXIMA
C       JLOC     :   STORES THE J-CENTROID LOCATION OF THE MAXIMA
C       IAREA    :   STORES THE NUMBER OF PIXELS IN THE MAXIMA
C       HIST     :   HISTGM OF THE ENTIRE IMAGE. USED TO LOCATE THE
```

```
C                    COLDEST (HIGHEST) NON-ZERO COUNT.
C       IHIST   :   UNUSED IN THIS ROUTINE
C       TMIN    :   DIGITAL COUNT CONVERTED TO BLACKBODY TEMP
C       TSMALL  :   TMIN CONVERTED TO SMALL SCALE TEMP
C       SLOPE   :   DIFFERENCE BETWEEN TMIN AND THE AVG OF 6 SURROUNDING POINTS
C       RATE    :   ASSIGNED RAINRATE MM/HR
C       NPIX    :   NUMBER OF RAINING PIXELS
C       IPASS   :   EITHER 0 OR 1 DEPENDING ON WHETHER THE POINT IS CLASSIFIED
C                   THUNDERSTORM OR CIRRUS
C       TMODE   :   THE STRATIFORM RAIN THRESHOLD TEMPERATURE
C       ITHR1   :   CLOUD THRESHOLD, 183 DIGITAL COUNTS OR 235 K
C
C       A MAXIMUM OF 600 MINIMA CAN BE STORED
C
      PARAMETER( MAXE=512, MAXL=256, MAXH=256, MAXM=600, MAXC=8000)
      INTEGER IARR(MAXE,MAXL),OSEC(MAXE,MAXL)
      INTEGER HISTO(MAXH)
      INTEGER IHIST(MAXH)
      INTEGER IAREA(MAXM),ILOC(MAXM),JLOC(MAXM),IMAX(MAXM),IPASS(MAXM)
CST4
      INTEGER MASK(MAXE,MAXL),ICLDTM(MAXM),ILOCX(MAXC),JLOCX(MAXC)
      INTEGER NAREA(MAXC),IMAXX(MAXC),IMODE(MAXC)
CST4
      REAL*4 RSLOPE(MAXM)
      CHARACTER*80 CTXT,CWRITE
      CHARACTER*12 CFI,CFG,CKWP,CBUG
C
      DATA ITHRSA/183/
      DATA ILOW,IHI,IACC/0,255,0/
      DATA ASLOPE/4.0/
C
      CALL ZEROW(MAXM,IMAX)
      CALL ZEROW(MAXM,ILOC)
      CALL ZEROW(MAXM,JLOC)
      CALL ZEROW(MAXM,IAREA)
CST4
      CALL ZEROW(MAXC,NAREA)
      CALL ZEROW(MAXM,ICLDTM)
CST4
      CALL ZEROW(MAXE*MAXL,OSEC)
C
      ITHR1 = IKWP('BLMT',1,ITHRSA)
C
C      TMIN PROCESSING (ISOLAT LOCAL MAXIMA IN COUNTS)
C
      CBUG=CKWP('DEBUG',1,'NO')
      MXORMN=1
CST4
      IPS=1
      IPE=512
      JLS=1
      JLE=256
CST4
      CALL ZEROW(MAXH,HISTO)
      CALL HISTGM(IARR,IDIM,JDIM,ILOW,IHI,IACC,HISTO,MAXCNT,MINC,
     +                                                MODE,MOST,ITOT)
      IF (CBUG(1:1).EQ.'Y'.OR.CBUG(1:1).EQ.'A') THEN
      CALL SDEST('MODE='//CFI(MODE)//'  MOST='//CFI(MOST)//'  ',ITOT)
      CALL SDEST('MINC ='//CFI(MINC)//'  MAXCNT ='//CFI(MAXCNT),0)
      ENDIF
      IF (CBUG(1:1).EQ.'Y') CALL SDEST('START MAXF',0)
      CALL MAXF(IARR,IDIM,JDIM,ITHR1,HISTO,MXORMN)
      IF (CBUG(1:1).EQ.'Y') CALL SDEST('END MAXF;START ISOLAT',0)
      CALL ISOLAT
     +       (IARR,IDIM,JDIM,ITHR1,IE,IAREA,ILOC,JLOC,IMAX,OSEC,
     +                                              IPS,IPE,JLS,JLE)
```

```
      IF (CBUG(1:1).EQ.'Y'.OR.CBUG(1:1).EQ.'A') THEN
        CALL SDEST('THRESHOLD = ', ITHR1 )
      ENDIF
      IF (CBUG(1:1).EQ.'A') THEN
        CWRITE='  CLOUD   AREA    ILOC    JLOC    IMAX    TMIN'//
     &                           '  SLOPE   PASS?   IMODE   NMODE'
        CALL SDEST(CWRITE,0)
      ENDIF
C
C      CST4 PROCESSING
C
C      SLOPE ESTIMATE TO SCREEN THIN CIRRUS
C
      CALL ZEROW(MAXC,IMODE)
      NUMMOD=0
      CALL ZEROW(MAXH,HISTO)
      CALL IRESET(IARR,IDIM,JDIM)
CST4
      IPASSE=0
CST4
      DO 32 I=1,IE
C
C        SLOPE ESTIMATE TO SCREEN THIN CIRRUS
C
        CALL SLOPE(IARR,IDIM,JDIM,ILOC(I),JLOC(I),RSLOPE(I))
        TMIN=TBB(IMAX(I))
C
C        IPASS=1: THUNDERSTORM.  IPASS=0: CIRRUS
C
        IPASS(I)=IDISCR(TMIN,RSLOPE(I))
        IF (RSLOPE(I).LE.ASLOPE.AND.IPASS(I).EQ.1) THEN
          CALL CALMOD(IARR,IDIM,JDIM,ILOC(I),JLOC(I),
     &                                ITHR1,IMOD,HISTO,NMOD)
          IF (CBUG(1:1).EQ.'A') THEN
            CTXT=' '
            WRITE(CTXT,30) I,IAREA(I),ILOC(I),JLOC(I),IMAX(I),
     &                         TMIN,RSLOPE(I),IPASS(I),IMOD,NMOD
  30        FORMAT(5I7,2F7.1,3I7)
            CALL SDEST(CTXT,0)
          ENDIF
        ELSE
          IF (CBUG(1:1).EQ.'A') THEN
            CTXT=' '
            WRITE(CTXT,40) I,IAREA(I),ILOC(I),JLOC(I),IMAX(I),
     &                            TMIN,RSLOPE(I),IPASS(I)
  40        FORMAT(5I7,2F7.1,I7,2(5X,'NA'))
            CALL SDEST(CTXT,0)
          ENDIF
        ENDIF
32    CONTINUE
CST4
      ITHR2=IMOD-10
CST4
      IF (CBUG(1:1).EQ.'H') THEN
        CALL SDEST('TMIN ELEMENT COMPOSITE BRIGHTNESS HISTOGRAM:',0)
        DO 888 L=ITHR1+1,MAXH
          CALL SDEST(CFI(L)//CFI(HISTO(L)),0)
888     CONTINUE
      ENDIF
CST4
      CALL IRESET(IARR,IDIM,JDIM)
      CALL ISOLAT(IARR,IDIM,JDIM,ITHR2,IENT,NAREA,ILOCX,JLOCX,IMAXX,
     +        MASK,IPS,IPE,JLS,JLE)
      CALL ZEROW(MAXC,IMODE)
      DO 34 II=1,IENT
        CALL ZEROW(MAXH,IHIST)
```

```
         CALL DOHIST(IARR,MASK,II,IDIM,JDIM,IHIST)
         CALL CALMO2(IHIST,IMODE(II),NUMMOD)
34    CONTINUE

CST4
C
C      CONVECTIVE RAIN ASSIGNMENT
C
      IF (CBUG(1:1).EQ.'A'.AND.NUMMOD.GT.0) THEN
         CWRITE='  CLOUD   AREA   ILOC   JLOC   IMAX   TMIN'//
     &                               '  TMINC   PASS?   NPIX   IRATE'
         CALL SDEST(CWRITE,0)
      ENDIF
      IPASED=0
      ISUMRA=0
      ISUMPX=0
      CALL ZEROW(MAXE*MAXL,OSEC)
      DO 33 I =1,IE
         IF (IPASS(I).EQ.1) THEN
            IF (IMAX(I).LT.ITHR2) GOTO 33
            KKK=MASK(ILOC(I),JLOC(I))
            ICLDTM(KKK)=1
            TEMPER=TBB(IMODE(KKK))
            IDRATE=NRATE(211.0)-NRATE(TEMPER)
            IDAREA=NPOINT(211.0)-NPOINT(TEMPER)
            TMIN=TBB(IMAX(I))
            TC=TSMALL(TMIN)
            IRATE=NRATE(TC)+IDRATE
            IF (CBUG(1:1).EQ.'Y') THEN
                CALL SDEST('I ',I)
                CALL SDEST('   KKK ',KKK)
                CALL SDEST('   IMODE(KKK) ',IMODE(KKK))
                CALL SDEST('   TEMPER '//CFG(TEMPER),0)
                CALL SDEST('   NRATE(TEMPER) ',NRATE(TEMPER))
                CALL SDEST('   IDRATE ',IDRATE)
                CALL SDEST('   IMAX(I) ',IMAX(I))
                CALL SDEST('   TMIN '//CFG(TMIN),0)
                CALL SDEST('   TC '//CFG(TC),0)
                CALL SDEST('   NRATE(TC) ',NRATE(TC))
                CALL SDEST('   IRATE ',IRATE)
            ENDIF
            IF (IRATE.LT.0) IRATE=0
            NPIX=NPOINT(TC)+IDAREA
            IF (NPIX.LT.0) NPIX=0
            CALL ASSIGN(OSEC,IDIM,JDIM,ILOC(I),JLOC(I),IRATE,NPIX,
     &                                                    ISUMR,ISUMP)
            IPASED=IPASED+1
            ISUMRA=ISUMRA+ISUMR
            ISUMPX=ISUMPX+ISUMP
            IF (CBUG(1:1).EQ.'A') THEN
               CTXT=' '
               WRITE(CTXT,31) I,IAREA(I),ILOC(I),JLOC(I),IMAX(I),
     &                                      TMIN,TC,IPASS(I),NPIX,IRATE
31             FORMAT(5I7,2F7.1,3I7)
               CALL SDEST(CTXT,0)
            ENDIF
         ENDIF
33    CONTINUE
      IF ((CBUG(1:1).EQ.'Y'.OR.CBUG(1:1).EQ.'A').AND.NUMMOD.GT.0) THEN
         CALL SDEST(' ',0)
         CALL SDEST(' TOTAL NUMBER OF TMIN DEFINED = ',IE)
         CALL SDEST(' NUMBER OF TMIN PASSING SLOPE TEST = ',IPASED)
         CALL SDEST(' ',0)
         CALL SDEST(' TMIN COMPOSITE MODAL BRIGHTNESS COUNT= ',IMODE)
         CWRITE=' TMIN COMPOSITE MODAL BRIGHTNESS TEMPERATURE= '//
     &                                        CFG(TSMALL(TBB(IMODE)))
```

```
         CALL SDEST(CWRITE,0)
         CALL SDEST(' ',0)
      ENDIF
C
C        STRATIFORM RAIN THRESHOLD IS BASED ON THE MODE TEMP FOR ALL T-MIN
C        WITH SLOPE < ASLOPE (USUALLY 4.0) AND IPASS=1
C
C        STRATIFORM ASSIGNMENT (WILL NOT REPLACE A CONVECTIVE ASSIGNMENT)
C
      ISRATE=2
      CALL STRAT2(IARR,OSEC,ISRATE,IPS,IPE,JLS,JLE,MASK,
     +                                                 IMODE,ICLDTM)
C
      RETURN
      END
C-----
      SUBROUTINE STRAT2(IARR,OSEC,ISRATE,IPS,IPE,JLS,JLE,
     +                  MASK,IMODE,ICLDTM)
      PARAMETER( MAXE=512, MAXL=256, MAXM=600, MAXC=8000 )
      INTEGER IARR(MAXE,MAXL),OSEC(MAXE,MAXL),MASK(MAXE,MAXL)
      INTEGER IMODE(MAXC),ICLDTM(*)
      DO 10 J=JLS,JLE
         DO 10 I=IPS,IPE
            KKK=MASK(I,J)
C... NOT PART OF ANY CLOUD ENTITY
            IF (KKK.EQ.0) GOTO 10
C... ALREADY CONVECTIVE
            IF (OSEC(I,J).GT.0) GOTO 10
C... ANNOTATION AND NOISE
            IF (IABS(IARR(I,J)).GE.240) GOTO 10
C... NO TMIN IN THIS CLOUD
            IF (ICLDTM(KKK).EQ.0) GOTO 10
            IF (IABS(IARR(I,J)).GE.IMODE(KKK)) OSEC(I,J)=ISRATE
10    CONTINUE
      RETURN
      END
C-----
      SUBROUTINE CALMO2(IHIST,IMODE,NUMMOD)
      PARAMETER (MAXH=256)
      INTEGER IHIST(MAXH)
C... TO CALCULATE THE MODE COUNT
      NUMMOD=0
      DO 10 IT=0,255
         NUMMOD=MAX0(NUMMOD,IHIST(IT+1))
         IF (NUMMOD.EQ.IHIST(IT+1)) IMODE=IT
10    CONTINUE
      RETURN
      END
C-----
      SUBROUTINE DOHIST(IARR,MASK,II,IDIM,JDIM,IHIST)
      PARAMETER ( MAXE=512, MAXL=256, MAXH=256 )
      INTEGER IARR(MAXE,MAXL), MASK(MAXE,MAXL), IHIST(MAXH)
C... DO HISTOGRAM OF CLOUD ENTITY II IN ARRAY IN USING CLOUD MASK
      DO 10 J=1,JDIM
      DO 10 I=1,IDIM
         IF (MASK(I,J).NE.II) GOTO 10
         K=IARR(I,J)
         IHIST(1+K)=IHIST(1+K)+1
10    CONTINUE
      RETURN
      END
C-----
      SUBROUTINE IRESET(IARR,IDIM,JDIM)
      PARAMETER( MAXE=512, MAXL=256 )
      INTEGER IARR(MAXE,MAXL)
      DO 10 J=1,JDIM
      DO 10 I=1,IDIM
         IARR(I,J)=IABS(IARR(I,J))
10    CONTINUE
      RETURN
      END
C-----
      FUNCTION IDISCR(TMIN,RSLOPE)
CCCCC DATA T1,S1/217.0,0.568/
      DATA T1,S1/210.0,0.4006/
      IF (RSLOPE.GE.30.0) GOTO 100
      ESLOPE=S1*(TMIN-T1)
      IF (RSLOPE.LT.ESLOPE) GOTO 100
      IDISCR=1
      RETURN
100   IDISCR=0
      RETURN
      END
C-----
      SUBROUTINE ASSIGN(OSEC,IDIM,JDIM,I,J,IRATE,NPIX,ISUMR,ISUMP)
C
C      CONVECTIVE RAIN ASSIGNMENT
C
C      THIS ROUTINE ASSIGNS IRATE TO NPIX PIXELS OF ARRAY OSEC,
C      DOING SO IN A "SPIRAL" PATTERN, AND RECOGNIZING THAT THE
C      SPACING IN THE X DIR IS 1/2 THE Y DIR SPACING
C
      PARAMETER( MAXE=512, MAXL=256 )
C FOR 4X8 IMAGERY
CCCCC PARAMETER (MAXSZ=30)
C FOR 4X4 IMAGERY
      PARAMETER (MAXSZ=42)
      INTEGER OSEC(MAXE,MAXL)
      INTEGER NEXTI(MAXSZ),NEXTJ(MAXSZ)
C FOR 4X8 IMAGERY
CCCCC DATA NEXTI/0,1,-1,0,0,2,-2,-1,1,1,-1,-2,2,2,-2,
CCCCC+                  3,-3,-3,3,3,-3,4,-4,0,0,-1,1,1,-1,-2/
CCCCC DATA NEXTJ/0,0,0,-1,1,0,0,-1,-1,1,1,-1,-1,1,1,
CCCCC+                  0,0,-1,-1,1,1,0,0,-2,2,-2,-2,2,2,-2/
C FOR 4X4 IMAGERY
      DATA NEXTI/0,1,1,0,-1,-1,-1,0,1,2,2,2,2,1,0,-1,
     &           -2,-2,-2,-2,-2,-1,0,1,2,3,3,3,3,3,3,
     &           2,1,0,-1,-2,-3,-3,-3,-3,-3,-3/
      DATA NEXTJ/0,0,1,1,1,0,-1,-1,-1,-1,0,1,2,2,2,2,2,
     &           1,0,-1,-2,-2,-2,-2,-2,-2,-1,0,1,2,3,3,
     &           3,3,3,3,3,2,1,0,-1,-2/
C
      IF(NPIX.GT.MAXSZ) NPIX=MAXSZ
      ISUMP=0
      DO 60 NP=1,NPIX
         IN=I+NEXTI(NP)
         IF (IN.LT.1. OR.IN.GT.IDIM) GO TO 60
         JN=J+NEXTJ(NP)
         IF (JN.LT.1.OR.JN.GT.JDIM) GO TO 60
         OSEC(IN,JN)=IRATE
         ISUMP=ISUMP+1
60    CONTINUE
      ISUMR=ISUMP*IRATE
      RETURN
      END
C-----
      SUBROUTINE SLOPE(IARR,IDIM,JDIM,I,J,RSLOPE)
      PARAMETER( MAXE=512, MAXL=256 )
      INTEGER IARR(MAXE,MAXL)
      DATA XMISS/1.0E+35/
C
C      TO CALCULATE DIFFERENCE BETWEEN VALUE OF T-MIN AND THE
```

```
C       [illegible]
C       AVERAGE OF A SAMPLING OF ITS NEAREST NEIGHBORS
C
C FOR 4X8 IMAGERY
CCCCC JRES=1
C FOR 4X4 IMAGERY
      JRES=2
CCC
      RSLOPE=XMISS
CCC
      JRE1=JRES-1
      IF (I.LE.2.OR.I.GE.(IDIM-1)) RETURN
      IF (J.LE.JRES.OR.J.GE.(JDIM-JRE1)) RETURN
      IF (IARR(I,J).EQ.0) RETURN
CCC
      TMIN=TBB(IABS(IARR(I,J)))
CCC
      P1=TBB(IABS(IARR(I-2,J)))
      P2=TBB(IABS(IARR(I-1,J)))
      P3=TBB(IABS(IARR(I+1,J)))
      P4=TBB(IABS(IARR(I+2,J)))
      P5=TBB(IABS(IARR(I,J-JRE1)))
      P6=TBB(IABS(IARR(I,J+JRE1)))
      P7=TBB(IABS(IARR(I,J-JRES)))
      P8=TBB(IABS(IARR(I,J+JRES)))
CCC
      RSLOPE=((P1+P2+P3+P4+P5+P6+P7+P8)/8.0) - TMIN
CCC
      RETURN
      END
C-----
      SUBROUTINE CALMOD(IARR,IDIM,JDIM,I,J,ITHRES,IMODE,HISTO,NUMMOD)
      PARAMETER( MAXE=512, MAXL=256 )
      IMPLICIT CHARACTER*12 (C)
      INTEGER IARR(MAXE,MAXL)
      INTEGER HISTO(256)
C
C... TO CALCULATE THE MODE COUNT IN A 20 BY 20 AREA AROUND THE T-MIN.
C... I, J ARE LOCATIONS OF T-MIN.
C... IMODE IS MODE COUNT OF SURROUNDING 20 ELE BY 20 LIN AREA (401 PTS)
C     WITHOUT EXCEEDING ARRAY BOUNDS.
C... PIXELS MUST BE COLDER THAN ITHRES TO BE INCLUDED IN CALCULATION.
C... NUMMOD IS THE NUMBER OF POINTS AT MODE.
C... AS POINTS IN IARR ARE COUNTED, THEY ARE NEGATED AND NOT USED AGAIN
C
C FOR 4X8 IMAGERY
CCCCC JRES=5
C FOR 4X4 IMAGERY
      JRES=10
      JS=J-JRES
      JE=J+JRES
      IS=I-JRES
      IE=I+JRES
      IF(JS.LT.1) JS=1
      IF(JE.GT.JDIM) JE=JDIM
      IF(IS.LT.1) IS=1
      IF(IE.GT.IDIM) IE=IDIM
      DO 10 JJ=JS,JE
      DO 10 II=IS,IE
         IT=IARR(II,JJ)
         IF (IT.LE.ITHRES) GO TO 10
         HISTO(IT+1)=HISTO(IT+1)+1
         NUMMOD=MAX0(NUMMOD,HISTO(IT+1))
         IF (NUMMOD.EQ.HISTO(IT+1)) IMODE=IT
         IARR(II,JJ)=-IT
10    CONTINUE
      RETURN
      -------
      END
C-----
      SUBROUTINE MAXF(IARR,IDIM,JDIM,ITHRES,HISTO,MXORMN)
C
C      INPUT (ALL PARAMETERS ARE INPUT):
C
C      IARR...THE ARRAY TO BE SEARCHED
C      IDIM,JDIM...THE DIMENSIONS OF THE ARRAY
C      ITHRES...THE LOWER THRESHOLD (FOR MAXIMA ONLY)
C      HISTO(1-256)...NUMBER OF OCCURRENCES OF COUNT VALUES 0-255
C      MXORMN... LT 0 GIVES MINIMA IN FIELD
C      MXORMN... GE 0 GIVES MAXIMA IN FIELD
C
C      PROGRAM IS LIKE LEVEL SLICE IN THAT IT STARTS AT THE COLD END
C      AND AFTER FINDING COLDEST (OR NEXT) COUNT, IT NEGATES ADJOINING
C      POINTS LESS THAN THAT VALUE. THE LOCATION OF THE NEGATED
C      POINTS IS STORED IN A CIRCULAR BUFFER. EACH POINT ADJOINING THOSE
C      POINTS IS NEGATED IF IT IS LESS THAN OR EQUAL TO THAT POINT.
C
C      POINTS ARE ALSO NEGATED IF THEY TOUCH A BOUNDARY.
C
C      POINTS LESS THAN THRESHOLD ARE NEGATED
C
      PARAMETER( MAXE=512, MAXL=256, MAXN=40000 )
      INTEGER HISTO(256)
      INTEGER ITOGO(MAXN),JTOGO(MAXN)
      INTEGER IARR(MAXE,MAXL)
C
      IT=ITHRES
      IF(MXORMN.LT.0) IT=1
      DO 200 ITIMES=IT,255
            IDX=ITIMES
C
C      REVERSE THE LOOP FOR MAXIMA SEARCH
C
            IF(MXORMN.GE.0) IDX=255-ITIMES+IT
            IF(HISTO(IDX+1).EQ.0) GO TO 200
            NTOGO=0
C
            DO 100 J=1,JDIM
            DO 100 I=1,IDIM
               ITEM1=IARR(I,J)
               IF(ITEM1.LE.0) GO TO 100
               IF(ITEM1.GE.IT) GO TO 1
               IARR(I,J)=-IARR(I,J)
               GO TO 100
1              CONTINUE
               IF(ITEM1.NE.IDX) GO TO 100
               IDEC=1
               CALL SEARCH(IARR,I,J,ITEM1,NTOGO,ITOGO,JTOGO,
     +                                 IDEC,IDIM,JDIM,MXORMN)
100         CONTINUE
      IF (NTOGO.GT.MAXN) THEN
      CALL SDEST('#1 TMIN ELEMENTS FOUND EXCEEDS 40000  ',NTOGO)
      NTOGO=MAXN
      ENDIF
C
C             NEGATE POINTS WHO ARE NEIGHBORS OF PREVIOUSLY NEGATED POINTS
C
            NDONE=0
29          IF(NDONE.GE.NTOGO) GO TO 200
            NDONE=NDONE+1
            I=ITOGO(NDONE)
            J=JTOGO(NDONE)
            ITEM1=-IARR(I,J)
            IDEC=2
```

```
           ---- -
           CALL SEARCH(IARR,I,J,ITEM1,NTOGO,ITOGO,JTOGO,
     +                      IDEC,IDIM,JDIM,MXORMN)
           GO TO 29
200      CONTINUE
      IF (NTOGO.GT.MAXN) THEN
      CALL SDEST('#2 TMIN ELEMENTS FOUND EXCEEDS 40000  ',NTOGO)
      NTOGO=MAXN
      ENDIF
C
C      NEGATE MAXIMA THAT TOUCH BOUNDARIES
C
      NTOGO=0
      DO 1100 J=1,JDIM
      DO 1100 I=1,IDIM
           IF(I.GT.1.AND.I.LT.IDIM.AND.J.GT.1.AND.J.LT.JDIM)
     +           GO TO 1100
           ITEM1=IARR(I,J)
           IF(ITEM1.LE.0) GO TO 1100
           IARR(I,J)=-IARR(I,J)
           IDEC=3
           CALL SEARCH(IARR,I,J,ITEM1,NTOGO,ITOGO,JTOGO,
     +                      IDEC,IDIM,JDIM,MXORMN)
1100  CONTINUE
      IF (NTOGO.GT.MAXN) THEN
      CALL SDEST('#3 TMIN ELEMENTS FOUND EXCEEDS 40000  ',NTOGO)
      NTOGO=MAXN
      ENDIF
C
C      NEGATE POINTS WHO ARE NEIGHBORS OF PREVIOUSLY NEGATED POINTS
C
           NDONE=0
129        IF(NDONE.GE.NTOGO) GO TO 1200
           NDONE=NDONE+1
           I=ITOGO(NDONE)
           J=JTOGO(NDONE)
           ITEM1=-IARR(I,J)
           IDEC=4
           CALL SEARCH(IARR,I,J,ITEM1,NTOGO,ITOGO,JTOGO,
     +                      IDEC,IDIM,JDIM,MXORMN)
           GO TO 129
1200      CONTINUE
      IF (NTOGO.GT.MAXN) THEN
      CALL SDEST('#4 TMIN ELEMENTS FOUND EXCEEDS 40000  ',NTOGO)
      NTOGO=MAXN
      ENDIF
      RETURN
      END
C-----
      SUBROUTINE SEARCH(IARR,I,J,ITEM1,NTOGO,ITOGO,JTOGO,
     +                        IDEC,IDIM,JDIM,MXORMN)
      PARAMETER( MAXE=512, MAXL=256, MAXN=40000 )
      INTEGER IARR(MAXE,MAXL)
      INTEGER ITOGO(MAXN),JTOGO(MAXN)
C
      NMAX=MAXN
      I1=I-1
      I2=I+1
      J1=J-1
      J2=J+1
      IF(I1.LT.1) I1=1
      IF(I2.GT.IDIM) I2=IDIM
      IF(J1.LT.1) J1=1
      IF(J2.GT.JDIM) J2=JDIM
      DO 300 JJ=J1,J2
      DO 300 II=I1,I2
           IF(II.EQ.I.AND.JJ.EQ.J) GO TO 300
```

```
            IF([illegible].EQ.0) GO TO 300
            ITEM2=IARR(II,JJ)
            IF(ITEM2.LE.0) GO TO 300
C
C       FOR MAXIMUM SEARCH:
C       1ST TIME, NEGATE NEIGHBOR IF NEIGHBOR IS LT ITEM1
C       2ND TIME, NEGATE NEIGHBOR IF NEIGHBOR IS LE ITEM1
C
      IF(MXORMN.GE.0) THEN
            IF(IDEC.EQ.1.AND.ITEM1.EQ.ITEM2) GO TO 300
            IF(IDEC.EQ.2.AND.ITEM2.GT.ITEM1) GO TO 300
      ELSE
            IF(IDEC.EQ.1.AND.ITEM1.LE.ITEM2) GO TO 300
            IF(IDEC.EQ.2.AND.ITEM2.LT.ITEM1) GO TO 300
      ENDIF
            IARR(II,JJ)=-IARR(II,JJ)
C
C            SAVE LOCATIONS IN CIRCULAR BUFFER
C
            NTOGO=NTOGO+1
            IF(IDEC.EQ.2.AND.NTOGO.GT.NMAX) NTOGO=1
            IF(IDEC.EQ.4.AND.NTOGO.GT.NMAX) NTOGO=1
            ITOGO(NTOGO)=II
            JTOGO(NTOGO)=JJ
300      CONTINUE
      RETURN
      END
C
      FUNCTION TSMALL(TMIN)
C
C       EMPIRICAL CORRECTION FOR FIELD OF VIEW
C       NO CORRECTION BELOW 200 K
C
      DATA SLOPE,YINTER/0.717,56.6/
      TSMALL=SLOPE*TMIN+YINTER
      IF(TMIN.LE.200.0) TSMALL=TMIN
      RETURN
      END
C
      FUNCTION NRATE(TC)
C
C       1-D MODEL RELATION BETWEEN TEMP (HEIGHT) AND RATE
C       FLORIDA RELATIONSHIPS CURRENTLY IN USE
C
      DATA SLOPE,YINTER/-0.2655,74.89/
      NRATE=NINT(SLOPE*TC+YINTER)
      RETURN
      END
C
      FUNCTION NPOINT(TC)
C
C       1-D MODEL RELATION BETWEEN TEMP (HEIGHT) AND RAIN AREA
C       FLORIDA RELATIONSHIPS CURRENTLY IN USE
C
      DATA SLOPE,YINTER/-0.0465,15.27/
C FOR 4X8 IMAGERY
CCCCC DATA PIXARE/24.4/
C FOR 4X4 IMAGERY
      DATA PIXARE/12.2/
      A=SLOPE*TC+YINTER
      NPOINT=NINT(EXP(A)/PIXARE)
      RETURN
      END
C
      FUNCTION TBB(ICOUNT)
C
C       TO CONVERT GOES IR COUNTS TO TBB
```

```
C       [illegible]
C
        INTEGER ICOUNT
        IF(ICOUNT.LE.176) THEN
              TBB=330.0-FLOAT(ICOUNT)/2.0
        ELSE
              TBB=418.0-FLOAT(ICOUNT)
        ENDIF
        RETURN
        END
C-----
        SUBROUTINE HISTGM
       +      (IARR,IDIM,JDIM,ILOW,IHI,IACC,
       2      HISTO,MAXC,MINC,MODE,MOST,ITOT)
C
C
C        THIS ROUTINE WORKS FOR IMAGE TYPE DATA (0-255)
C        AND CONSIDERS DATA IN IARR (OF DIMENSION IDIM,JDIM) WHOSE VALUES
C        ARE BETWEEN ILOW AND IHI INCLUSIVE
C
C        COMPUTE IMAGE HISTGM (HISTO), MAXIMUM, MINIMUM, AND MODE COUNTS
C        (MAXC, MINC, MODE),
C        THE NUMBER OF COUNTS AT THE MODE (MOST) AND TOTAL COUNTS (ITOT)
C
        PARAMETER( MAXE=512,MAXL=256 )
        INTEGER IARR(MAXE,MAXL)
        INTEGER HISTO(256)
C
C        IF IACC=1, ACCUMULATE COUNTS FROM MULTIPLE IMAGES
C
        IF(IACC.EQ.1) GO TO 1
C
C        INITIALIZE COUNTERS
C
        MAXC=0
        MOST=0
        MINC=255
        ITOT=0
C
1       CONTINUE
        DO 40 J=1,JDIM
        DO 40 I=1,IDIM
           IT=IARR(I,J)
           IF(IT.LT.ILOW.OR.IT.GT.IHI) GO TO 40
           HISTO(IT+1)=HISTO(IT+1)+1
           MAXC=MAX0(MAXC,IT)
           MINC=MIN0(MINC,IT)
           MOST=MAX0(MOST,HISTO(IT+1))
           IF(MOST.EQ.HISTO(IT+1)) MODE=IT
           ITOT=ITOT+1
40      CONTINUE
        RETURN
        END
C-----
        SUBROUTINE ISOLAT (ISEC,IDIM,JDIM,ITHRES,IE,IAREA,ILOC,JLOC,
       +                   IMAX,OSEC,IPS,IPE,JLS,JLE)
C
C        SEARCHES ARRAY ISEC AND DEFINES ENTITIES BY A THRESHOLD VALUE.
C
C        INPUT PARAMETERS (FIRST LINE) ARE ISEC (THE ARRAY TO BE
C        "CONTOURED"), IDIM,JDIM (THE DIMENSIONS) AND THE THRESHOLD (ITHR1).
C
C        OUTPUT (2ND LINE) IS:
C        IE (# OF ENTITIES),
C        IAREA (#PIXELS IN EACH ENTITY),
C        ILOC,JLOC (COORD'S OF CENTROID)
C        IMAX (MAXIMUM COUNT PER CLOUD)
```

```
C      IAREA (ENTITY COUNT PER CLOUD),
C      AND IHIST (ENTITY HISTGM).
C      IHIST CAN THEN BE USED TO CALCULATE 10%, 50%  VALUES.
C
C      ADDITIONAL OUTPUT IS THE ARRAY OSEC, A MASK OF THE ORIGINAL ARRAY
C      WITH POINTS LESS THAN ITHR1 IN ISEC SET TO ZERO IN OSEC AND POINTS
C      GREATER OR EQUAL TO ITHR1 IN ISEC SET TO THE CLOUD IDENTIFIER (IE)
C      IN OSEC.
C
      PARAMETER( MAXE=512,MAXL=256,MAXM=600,MAXC=8000)
      INTEGER ISEC(MAXE,MAXL),OSEC(MAXE,MAXL)
      INTEGER ITOGO(MAXC),JTOGO(MAXC)
      INTEGER IAREA(MAXC),ILOC(MAXC),JLOC(MAXC),IMAX(MAXC)
      DIMENSION ISQ(3)
      DATA ISQ/-1,0,1/
      DATA ISCH/MAXC/
C
C      SEARCH FOR CLOUD SYSTEMS
C
      IE=0
      IEND=IDIM-1
      JEND=JDIM-1
      CALL ZEROW(MAXM,IAREA)
      CALL ZEROW(MAXM,ILOC)
      CALL ZEROW(MAXM,JLOC)
      CALL ZEROW(MAXM,IMAX)
      CALL ZEROW(MAXE*MAXL,OSEC)
      IEND = IPE-1
      JEND = JLE-1
      ISTART = IPS+1
      JSTART = JLS+1
      DO 100 JC=JSTART,JEND
      DO 100 IC=ISTART,IEND
            IF(OSEC(IC,JC).GT.0 .OR.
     +                     ISEC(IC,JC).LT.ITHRES) GO TO 100
C
C            BRIGHTNESS GREATER THAN THRESHOLD,INCREMENT CLOUD COUNTER IE
C
30            IE=IE+1
            IF(IE.LE.MAXM) GO TO 34
            CALL EDEST( 'TOO MANY CLOUDS IN THRESHOLD ',MAXM)
            IE=IE-1
            GO TO 150
C
C            STARTING INDEX FOR FRAGMENT OF CLOUD IE
C
34            NTOGO=1
            NDONE=0
C
C            PLACE POINT'S LOCATION IN CIRCULAR BUFFER
C
            ITOGO(1)=IC
            JTOGO(1)=JC
C
C            UPDATE HISTGM
C
            ICV=ISEC(IC,JC)
            IAREA(IE)=IAREA(IE)+1
            ILOC(IE)=ILOC(IE)+IC
            JLOC(IE)=JLOC(IE)+JC
            IMAX(IE)=ICV
C
C            UPDATE OSEC, MARKING POINT AS FOUND
C
            OSEC(IC,JC)= IE
C
29            IF(NDONE.LT.NTOGO) GO TO 99
```

```
            ILOC(IE)=NINT(FLOAT(ILOC(IE))/FLOAT(IAREA(IE)))
            JLOC(IE)=NINT(FLOAT(JLOC(IE))/FLOAT(IAREA(IE)))
            GO TO 100
99            CONTINUE
            NDONE=NDONE+1
            IF(NDONE.GT.ISCH) NDONE=1
C
C            GET CENTER OF NEXT 8-POINT SEARCH
C
            I=ITOGO(NDONE)
            J=JTOGO(NDONE)
C
C            SEARCH THE 8 NEIGHBORS
C
            DO 50 L=1,3
            DO 50 K=1,3
                  IF(K.EQ.2.AND.L.EQ.2) GO TO 50
                  ID=I+ISQ(K)
                  JD=J+ISQ(L)
                  IF(ID.LT.1.OR.ID.GT.IDIM) GO TO 50
                  IF(JD.LT.1.OR.JD.GT.JDIM) GO TO 50
                  ICV=ISEC(ID,JD)
                  IF(OSEC(ID,JD).GT.0.OR.ICV.LT.ITHRES) GO TO 50
C
C                  BRIGHTNESS GREATER THAN THRESHOLD,
C                  SO IT’S ANOTHER POINT OF CLOUD IE
C
                  NTOGO=NTOGO+1
                  IF(NTOGO.GT.ISCH) NTOGO=1
C
C                  SAVE LOCATION IN CIRCULAR BUFFER
C
                  ITOGO(NTOGO)=ID
                  JTOGO(NTOGO)=JD
C
C                  UPDATE HISTGM AND # OF PIXELS
C
48                CONTINUE
                  IAREA(IE)=IAREA(IE)+1
                  ILOC(IE)=ILOC(IE)+ID
                  JLOC(IE)=JLOC(IE)+JD
                  IMAX(IE)=MAX0(IMAX(IE),ICV)
C
C                  UPDATE OSEC, MARKING POINT AS FOUND
C
                  OSEC(ID,JD)= IE
50             CONTINUE
      GO TO 29
100            CONTINUE
150      RETURN
      END
```

APPENDIX C. Correction for Summed Rain-Image Directories

This procedure assumes the images are stored on a McIDAS work station. By means of the "change area" (CA) command, simply replace the existing sum-image directory by the individual-image directory. Sum images are 0X93, 0X95, 0X97, 0X99 and 0X91; these numbers correspond, respectively, to sums over images 1-6, 7-12, 13-18, 19-24 and 1-24. For CST1, X=2; for CST4, X=4; for ARK, X=3. Individual image directories are identical for infrared images and their corresponding rain images, which is to say the information required to correct a sum-image directory can be obtained either from the appropriate infrared image or its equivalent (individual) rain image.

Correct the 1-6 sum image by means of the first infrared image; the 7-12 sum image by means of the seventh infrared image; the 13-18 sum image by means of the thirteenth infrared image and the 19-23 sum image by means of the ninteenth infrared image. Correct the 1-24 sum image (the daily rain image, whether ARK or CST) by means of the first infrared image.

As an example, consider the following case.

> Area 491 holds the sum 1-24 image for julian day 186 (4 July). It acquired its directory from area 248 (note that 491 and 248 directories are identical). Area 491 acquired its navigation from area 225, which has different line coordinates (lcor). In general, the element coordinates (ecor) also will be different. We give area 491 compatible directory and navigation information by changing its lcor, through the CA command. McIDAS automatically echoes the corrected directory, which, after the correction, is identical with that of area 225.

```
  area   ss  yyddd hhmmss lcor ecor lres eres zres lsiz esiz zsiz    bands
  ---- ---- ----- ------ ---- ---- ---- ---- ---- ---- ---- ----    -----
   491   33  88186 230100 2215 4558    4    4    1  256  512    1 1...........
LA:  Done
LA 248
  area   ss  yyddd hhmmss lcor ecor lres eres zres lsiz esiz zsiz    bands
  ---- ---- ----- ------ ---- ---- ---- ---- ---- ---- ---- ----    -----
   248   33  88186 230100 2215 4558    4    4    1  256  512    2 .......8....
LA:  Done
LA 225
  area   ss  yyddd hhmmss lcor ecor lres eres zres lsiz esiz zsiz    bands
  ---- ---- ----- ------ ---- ---- ---- ---- ---- ---- ---- ----    -----
   225   33  88186    100 2191 4558    4    4    1  256  512    2 .......8....
LA:  Done
CA 491 X X X 2191 4558
  area   ss  yyddd hhmmss lcor ecor lres eres zres lsiz esiz zsiz    bands
  ---- ---- ----- ------ ---- ---- ---- ---- ---- ---- ---- ----    -----
   491   33  88186 230100 2191 4558    4    4    1  256  512    1 1...........
     proj:    0 created: 92044 231829  memo: "SUM 492 499
     type:PRD       cal type:BRIT
     area offsets:  data= 2816 navigation=  256 calibration=  768
     doc length:    0   cal length:    0   lev length:    0 PREFIX=   0
```