(2)

Naval Ocean Research and
Development Activity
Stennis Space Center, Mississippi 39529-5004

NORDA Technical Note 388
July 1988

KRMS GEOSAT - LIMEX '87
Data Products

D. T. Eppler
L. D. Farmer
Oceanography Division
Ocean Science Directorate

ABSTRACT

The K_a-band Radiometric Mapping System (KRMS) was used to acquire high resolution passive microwave imagery of sea ice in the East Greenland and Labrador Seas during March 1987. Images obtained in the East Greenland Sea supported NORDA's GEOSAT ice index validation experiment; Labrador Sea images supported the Labrador Ice Margin Experiment (LIMEX). In addition, we acquired several image sets that show open ocean in the North Atlantic Ocean and East Greenland Sea regions, glacial ice along the east central margin of the Greenland ice cap near Scoresby Sound, and land along the Avalon Peninsula of Nova Scotia.

ACKNOWLEDGMENTS

This work was supported by the Oceanographer of the Navy (OP-096, project element 63704N), through the Satellite Applications Technology (SAT) program, A. E. Pressman, program manager. The authors thank Bruce Heydlauff of the Naval Weapons Center, China Lake, California, who provided consummate assistance in analog to digital conversion of KRMS data.

KRMS GEOSAT--LIMEX '87
Data Products

DUANE T. EPPLER AND L. DENNIS FARMER

INTRODUCTION

The K_a-band Radiometric Mapping System (KRMS) was used to acquire high resolution passive microwave imagery of sea ice in the East Greenland and Labrador Seas during March 1987. Images obtained in the East Greenland Sea supported NORDA's GEOSAT ice index validation experiment (Chase and Holyer, 1988); Labrador Sea images supported the Labrador Ice Margin Experiment (LIMEX) (McNutt et al., 1988). In addition, we acquired several image sets that show open ocean in the North Atlantic Ocean and East Greenland Sea regions, glacial ice along the east central margin of the Greenland ice cap near Scoresby Sound, and land along the Avalon Peninsula of Nova Scotia. Farmer et al. (1988) provide navigation logs, track plots, KRMS system logs, and representative image samples for these data.

Approximately 17 hours of imagery was recorded on analog tape. Most analog data subsequently were converted to digital format and recorded on computer compatible tapes (CCTs) at a density of 1600 bpi. The purpose of this report is to describe this digital product, to provide an index to tapes in the data archive, and to present an algorithm and a set of associated programs that were written to allow correlation between GEOSAT footprints and KRMS image segments that were collected along GEOSAT ground tracks.

DIGITAL TAPE PRODUCTS

Data were digitized from analog tapes on an AT-compatible computer running under DOS at facilities provided by Naval Weapons Center (Code 3251) in China Lake, California. Raw digital files were copied into a XENIX-based system for subsequent processing and for production of two different sets of archive tapes at the NORDA Polar Oceanography Branch Office in Hanover, New Hampshire. The two sets of tapes contain the same radiometric information, but differ both in terms of the format in which the information is recorded and in terms of ancillary data recorded in conjunction with KRMS radiometers. One set of tapes was produced using the XENIX "tar" function and contains both pixel intensities and a code that signifies the KRMS antenna used to acquire the data. The other set of tapes was produced using the XENIX "dd" function and contains only pixel intensities. In subsequent discussion we use these function names (tar and dd) to distinguish one set of archive tapes from the other. We provide indices to tar tapes and dd tapes in Appendices A and B.

The primary difference between tar tapes and dd tapes is the format in which data are recorded.

tar format:

tar is a XENIX/UNIX tape archiver function that copies entire directories to backup media. Data archived with tar are compressed and the file directory structure typical of UNIX and XENIX systems is preserved. Users running under XENIX (and probably UNIX) should be able to restore KRMS image files directly from these tar tapes. However, data stored in

this way typically are illegible to non-UNIX/XENIX systems; users running under other operating systems probably will find their experience with the tar tapes somewhat less than rewarding.

dd format:

dd is the general purpose file copying utility provided by XENIX. Data in dd files are preserved in a standard format that is compatible with counterpart I/O utilities resident in most operating systems. KRMS data tapes written with dd reside in binary format in unlabelled files with constant record length and block size. Data written with dd are neither compressed, nor stored in a directory structure. KRMS users on non-XENIX systems (Xontech, Arete, CRREL Geological Sciences Branch, NORDA Remote Sensing Branch) have not encountered significant problems reading KRMS dd files, although some systems require byte swapping.

Images recorded on both sets of tapes are stored scan by scan in the sequence scans were acquired by KRMS. One scan consists of radiances measured across the entire 100 degree field of view by one of KRMS's three antennas. Later scans in a file represent data acquired farther downtrack than scans that occur at the beginning of a file. Each image consists of 2048 scans, each scan consists of 512 pixels, and each pixel occupies two 8-bit bytes. The first 1024 bytes in a file, for example, contain unsigned integer values for the 512 pixels that constitute the first scan in the image. Images that are adjacent to one another have been recorded with an overlap of approximately 50 scans. That is to say that approximately the first 50 lines of one image duplicate data in the last 50 lines of the previous image.

Figure 1 shows differences in bit patterns used for pixels on tar and dd tapes. Pixels on tar tapes include a two-bit antenna code that has been stripped from pixels on dd tapes.

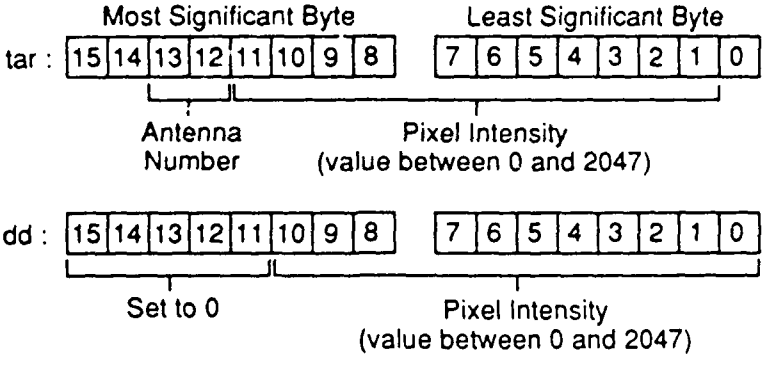

Figure 1. Bit patterns used for pixels recorded on tar and dd archive tapes. On tar tapes, pixel intensity is stored in bits 1-11, and a numeric code that identifies the KRMS antenna used to obtain the data is stored in bits 12 and 13. The antenna code will be an integer 0, 1, or 2. Bit 0 is forced high (set to 1) by the interface to digitizing hardware. Bits 14 and 15 float, but generally are 0 and 1 respectively. On dd tapes, the 11-bit pixel intensity has been shifted to the least significant bits (0-10). Bits 11-15 are set to 0.

This code signifies which of the three KRMS antennas was used to acquire data stored in the pixel. The antenna code is not part of the pixel value and must be masked when images are restored from tar tapes. Integral values recorded on both sets of tapes represent relative intensities sensed by KRMS; these data are not calibrated and, in this form, do not correspond directly to radiometric brightness temperatures. See Farmer et al. (1989) for discussion of methods by which these data can be converted to 33.6-GHz brightness temperatures.

CORRELATION WITH GEOSAT FOOTPRINTS

The primary objective of the March 1987 KRMS mission was to document ice conditions along GEOSAT ground tracks in near real time with respect to GEOSAT overpasses. KRMS was on track collecting imagery during eight of the nine satellite overpasses, and arrived at the ninth track within one hour of the satellite pass. The goal of obtaining a simultaneous KRMS-GEOSAT data set thus was met within limits imposed by GEOSAT ephemeris data and P-3A navigation instruments (Farmer et al., 1988).

In order to utilize KRMS data to verify NORDA's GEOSAT ice index, sections of the KRMS data set that coincide with each GEOSAT footprint must be identified. This was accomplished by correlating KRMS navigation data with satellite ephemeris data according to the following algorithm:

Given:

1. Latitude--longitude coordinates of GEOSAT footprint midpoints.
2. Latitude--longitude-time coordinates of the KRMS P-3A aircraft at approximately five-minute intervals along track.
3. Time coordinates of scans in KRMS image files.

Then:

1. Compute KRMS/P-3 coordinates that correspond to GEOSAT footprint midpoints; set KRMS/P-3 longitude equal to GEOSAT longitude and interpolate latitude and time.
2. Compute KRMS/P-3 times that correspond to beginning and end of each GEOSAT footprint; assume one footprint ends and another begins halfway between two consecutive footprint midpoints.
3. Use these times to compute KRMS scan lines that correspond to beginning and end of each footprint.

Appendix C is a listing of NAVDAT, a program written in C to implement this algorithm. Appendices D, E, and F are listings of ENTERG, ENTERP, and ENTERI, programs written to create files of coordinates of GEOSAT footprint locations, KRMS/P-3 aircraft positions, and KRMS images.NAVDAT requires these as input. Appendix G is output from NAVDAT for selected GEOSAT tracks.

LITERATURE CITED

Chase, J. R. and R. J. Holyer (1988) Estimation of sea-ice type and concentration by linear unmixing of GEOSAT altimeter waveforms. Abstracts of papers submitted to the 1988 IGARS Meeting, Edinburgh, Scotland.

Farmer, L. D., D. T. Eppler, R. E. Burge, and A. W. Lohanick (1988) KRMS GEOSAT-LIMEX'87 quick look report. NORDA Technical Note 358, 47p.

Farmer, L. D., D. T. Eppler, and A. W. Lohanick (1989) Converting digital passive microwave data to kelvin units of brightness temperature. NORDA Technical Note 427 (in prep.).

McNutt, L., S. Argus, F. Carsey, B. Holt, J. Crawford, C. Tang, A. L. Gray, and C. Livingstone (1988) LIMEX '87: the Labrador ice margin experiment, March 1987 -- A pilot experiment in anticipation of RADARSAT and ERS 1 data: *EOS,* 69, (23): 634-643.

APPENDIX A: INDEX TO tar FORMAT TAPES

Notes on column headings

Gains and offsets: those listed here were applied to the analog signal during the digitizing process and are independent of gains and offsets given in logs provided by Farmer et al. (1988). All files digitized with the same gain and offset probably can be calibrated with the same equation. Changes in gain and offset may require that the calibration equation be adjusted.

Number of lines lost: refers to the minimum number of scans that are missing from each image as a result of asynchronous timing between digitizing rates characteristic of the A/D converter and the rate at which the AT computer can copy data.

Tape numbers: analog tape number of the number of the analog mission tape on which data were recorded aboard the P-3A aircraft. Digital tape number refers to the number of the archived tar tape.

DATA SET: G10589

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	82	05:08:17	05:10:45	1.930	5.715	2048	27	3	1
f2	82	05:10:43	05:13:17	1.930	5.715	2048	25	3	1
f3	82	05:13:10	05:15:26	1.930	5.715	2048	25	3	1
f4	82	05:15:21	05:17:39	1.930	5.715	2048	23	3	1
f5	82	05:17:34	05:20:00	1.930	5.715	2048	20	3	1
f6	82	05:19:55	05:22:20	1.930	5.715	2048	21	3	1
f7	82	05:22:15	05:24:41	1.930	5.715	2048	21	3	1
f8	82	05:24:36	05:27:00	1.930	5.715	2048	17	3	1
f9	82	05:26:55	05:29:18	1.930	5.715	2048	22	3	1
f10	82	05:29:13	05:31:34	1.930	5.715	2048	15	3	1
f11	82	05:31:29	05:33:49	1.930	5.715	2048	21	3	2
f12	82	05:33:44	05:36:04	1.930	5.715	2048	19	3	2
f13	82	05:35:59	05:38:21	1.930	5.715	2048	16	3	2
f14	82	05:38:16	05:40:39	1.930	5.715	2048	14	3	2
f15	82	05:40:34	05:43:00	1.930	5.715	2048	12	3	2
f16	82	05:42:55	05:45:20	1.930	5.715	2048	19	3	2

DATA SET: G10589

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f17	82	05:45:18	05:47:49	1.930	5.715	2048	13	3	2
f18	82	05:47:44	05:50:10	1.930	5.715	2048	16	3	2
f19	82	05:50:05	05:52:32	1.930	5.715	2048	18	3	2
f20	82	05:52:27	05:54:53	1.930	5.715	2048	16	3	2
f21	82	05:54:48	05:56:00	1.930	5.715	2048	77	3	2
f21	82	06:08:12	06:10:46	1.930	5.715	2048		3	2

DATA SET: G10590

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	82	07:12:22	07:15:03	1.930	5.69	2048	13	3	3
f2	82	07:14:57	07:19:15	1.930	5.69	2048	16	3	3
f3	82	07:19:10	07:24:45	1.930	5.69	2048	20	3	3
f4	82	07:24:40	07:30:22	1.930	5.69	2048	16	3	3
f5	82	07:30:17	07:35:54	1.930	5.69	2048	20	3	3
f6	82	07:35:49	07:41:23	1.930	5.69	2048	17	3	3
f7	82	07:41:18	07:46:52	1.930	5.69	2048	10	3	3
f8	82	07:46:47	07:52:33	1.930	5.69	2048	17	3	3

DATA SET: G10591

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	82	09:26:30	09:28:44	1.930	5.69	2048	9	3	4
f2	82	09:28:39	09:31:01	1.930	5.69	2048	5	3	4
f3	82	09:30:56	09:33:51	1.930	5.69	2048	22	3	4
f4	82	09:33:46	09:37:00	1.930	5.69	2048	19	3	4
f5	82	09:36:55	09:40:00	1.930	5.69	2048	13	3	4
f6	82	09:39:55	09:43:29	1.930	5.69	2048	13	3	4
f7	82	09:43:24	09:46:56	1.930	5.69	2048	20	3	4
f8	82	09:46:51	09:49:52	1.930	5.69	2048	15	3	4
f9	82	09:49:47	09:52:38	1.930	5.69	2048	14	3	4
f10	82	09:52:33	09:55:24	1.930	5.69	2048	20	3	4

DATA SET: G10603

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	83	05:17:00	05:19:55	1.930	5.69	2048	19	4	5
f2	83	05:19:50	05:22:46	1.930	5.69	2048	19	4	5
f3	83	05:22:41	05:25:31	1.930	5.69	2048	16	4	5
f4	83	05:25:26	05:28:08	1.930	5.69	2048	17	4	5
f5	83	05:28:03	05:30:34	1.930	5.69	2048	20	4	5
f6	83	05:30:29	05:33:00	1.930	5.69	2048	20	4	5
f7	83	05:32:55	05:35:25	1.930	5.69	2048	13	4	5
f8	83	05:35:20	05:37:50	1.930	5.69	2048	17	4	5
f9	83	05:37:45	05:40:14	1.930	5.69	2048	16	4	5
f10	83	05:40:09	05:42:38	1.930	5.69	2048	15	4	5
f11	83	05:42:33	05:45:01	1.930	5.69	2048	9	4	5
f12	83	05:44:56	05:47:25	1.930	5.69	2048	9	4	5
f13	83	05:47:20	05:49:49	1.930	5.69	2048	13	4	5
f14	83	05:49:44	05:52:12	1.930	5.69	2048	23	4	5
f15	83	05:52:07	05:54:35	1.930	5.69	2048	14	4	5
f16	83	05:54:30	05:56:57	1.930	5.69	2048	19	4	5

DATA SET: G10603

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f17	83	05:56:52	05:59:20	1.930	5.69	2048	18	4	6
f18	83	05:59:15	06:01:42	1.930	5.69	2048	16	4	6
f19	83	06:01:37	06:04:02	1.930	5.69	2048	17	4	6
f20	83	06:03:57	06:06:22	1.930	5.69	2048	4	4	6
f21	83	06:06:17	06:08:42	1.930	5.69	2048	15	4	6
f22	83	06:08:37	06:11:05	1.930	5.69	2048	18	4	6
f23	83	06:11:00	06:14:22	1.930	5.69	2048	366	4	6

DATA SET: G10604

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	83	06:33:19	06:36:57	1.930	5.69	2048	29	4	7
f2	83	06:36:52	06:39:24	1.930	5.69	2048	20	4	7
f3	83	06:39:10	06:42:55	1.930	5.69	2048	21	4	7
f4	83	06:42:50	06:46:02	1.930	5.69	2048	18	4	7
f5	83	06:45:57	06:48:59	1.930	5.69	2048	31	4	7
f6	83	06:48:54	06:51:55	1.930	5.69	2048	22	4	7
f7	83	06:51:50	06:54:51	1.930	5.69	2048	22	4	7
f8	83	06:54:46	06:57:47	1.930	5.69	2048	16	4	7
f9	83	06:57:42	07:00:34	1.930	5.69	2048	18	4	7
f10	83	07:00:29	07:03:24	1.930	5.69	2048	18	4	7
f11	83	07:03:19	07:06:17	1.930	5.69	2048	27	4	7
f12	83	07:06:12	07:09:08	1.930	5.69	2048	11	4	7
f13	83	07:09:07	07:12:59	1.930	5.69	2048	15	4	7
f14	83	07:11:54	07:14:49	1.930	5.69	2048	20	4	7

DATA SET: G10605

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	83	08:39:44	08:43:32	1.930	5.69	2048	19	4	8
f2	83	08:43:27	08:46:36	1.930	5.69	2048	16	4	8
f3	83	08:44:30	08:47:24	1.930	5.69	2048	15	4	8
f4	83	08:47:10	08:50:12	1.930	5.69	2048	26	4	8
f5	83	08:50:07	08:53:02	1.930	5.69	2048	9	4	8
f6	83	08:52:57	08:55:55	1.930	5.69	2048	20	4	8
f7	83	08:55:50	08:58:54	1.930	5.69	2048	27	4	8
f8	83	08:58:41	09:01:24	1.930	5.69	2048	452	4	8

DATA SET: G10632

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	85	05:50:00	05:52:30	1.930	5.69	2048	14	5	9
f2	85	05:52:25	05:54:48	1.930	5.69	2048	10	5	9
f3	85	05:54:43	05:57:06	1.930	5.69	2048	16	5	9
f4	85	05:57:01	05:59:23	1.930	5.69	2048	13	5	9
f5	85	05:59:18	06:01:41	1.930	5.69	2048	12	5	9
f6	85	06:01:36	06:03:59	1.930	5.69	2048	15	5	9
f7	85	06:03:54	06:06:17	1.930	5.69	2048	18	5	9
f8	85	06:06:12	06:08.35	1.930	5.69	2048	14	5	9
f9	85	06:08:30	06:10:53	1.930	5.69	2048	12	5	9
f10	85	06:10:48	06:13:12	1.930	5.69	2048	17(?)	5	9
f11	85	06:13:07	06:15:30	1.930	5.69	2048	17	5	10
f12	85	06:15:25	06:18:01	1.930	5.69	2048	12	5	10
f13	83	06:17:56	06:20:37	1.930	5.69	2048	14	5	10
f14	85	06:20:32	06:23:14	1.930	5.69	2048	16	5	10
f15	85	06:23:09	06:25:50	1.930	5.69	2048	16	5	10
f16	85	06:25:45	06:28:26	1.930	5.69	2048	13	5	10

DATA SET: G10632

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f17	85	06:28:21	06:31:03	1.930	5.69	2048	18	5	10
f18	85	06:30:58	06:33:40	1.930	5.69	2048	20	5	10
f19	85	06:33:35	06:36:16	1.930	5.69	2048	10	5	10
f20	85	06:36:11	06:38:52	1.930	5.69	2048	6	5	10
f21	85	06:38:47	06:40:14	1.930	5.69	2048	11	5	10

DATA SET: G10633

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	85	07:12:20	07:14:58	1.93	5.69	2048	16	5	11
f2	85	07:14:53	07:17:31	1.93	5.69	2048	19	5	11
f3	85	07:17:26	07:20:01	1.93	5.69	2048	24	5	11
f4	85	07:19:56	07:22:29	1.93	5.69	2048	13	5	11
f5	85	07:22:24	07:25:01	1.93	5.69	2048	30	5	11
f6	85	07:24:56	07:27:33	1.93	5.69	2048	15	5	11
f7	85	07:27:28	07:30:11	1.93	5.69	2048	14	5	11
f8	85	07:30:06	07:32:51	1.93	5.69	2048	16	5	11
f9	85	07:32:46	07:35:31	1.93	5.69	2048	20	5	11
f10	85	07:35:26	07:38:11	1.93	5.69	2048	19	5	11
f11	85	07:38:06	07:40:51	1.93	5.69	2048	19	5	11
f12	85	07:40:46	07:43:31	1.93	5.69	2048	10	5	11

DATA SET: G10634

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	85	09:14:00	09:16:49	1.93	5.69	2048	26	5	12
f2	85	09:16:44	09:19:16	1.93	5.69	2048	21	5	12
f3	85	09:19:11	09:21:39	1.93	5.69	2048	19	5	12
f4	85	09:21:34	09:23:59	1.93	5.69	2048	9	5	12
f5	85	09:23:54	09:26:20	1.93	5.69	2048	16	5	12
f6	85	09:26:15	09:28:41	1.93	5.69	2048	12	5	12
f7	85	09:28:36	09:31:03	1.93	5.69	2048	19	5	12

DATA SET: LIMEX1

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	16:52:30	16:55:15	1.93	5.69	2048	21	2	13
f2	80	16:55:10	16:57:54	1.93	5.69	2048	14	2	13
f3	80	16:57:49	17:00:34	1.93	5.69	2048	14	2	13
f4	80	17:00:29	17:03:14	1.93	5.69	2048	16	2	13
f5	80	17:03:09	17:05:55	1.93	5.69	2048	18	2	13

DATA SET: LIMEX2

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	17:09:00	17:11:47	1.93	5.69	2048	15	2	13
f2	80	17:11:42	17:14:18	1.93	5.69	2048	17	2	13
f3	80	17:14:13	17:16:40	1.93	5.69	2048	12	2	13
f4	80	17:16:35	17:19:03	1.93	5.69	2048	11	2	13
f5	80	17:18:58	17:21:26	1.93	5.69	2048	14	2	13

DATA SET: LIMEX3

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	17:23:00	17:26:02	1.93	5.69	2048	29	2	14
f2	80	17:25:57	17:29:11	1.93	5.69	2048	15	2	14
f3	80	17:29:06	17:32:21	1.93	5.69	2048	19	2	14

DATA SET: LIMEX4

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	17:44:30	17:47:18	1.93	5.69	2048	15	2	14
f2	80	17:47:13	17:50:02	1.93	5.69	2048	76	2	14

DATA SET: LIMEX5

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	17:52:00	17:54:50	1.93	5.69	2048	13	2	14
f2	80	17:54:45	17:57:36	1.93	5.69	2048	13	2	14
f3	80	17:57:31	18:00:23	1.93	5.69	2048	20	2	14

DATA SET: MSCL.gla1

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	82	06:08:19	06:12:56	1.93	5.69	2048	18	3	15
f2	82	06:12:51	06:17:31	1.93	5.69	2048	26	3	15
f3	82	06:17:26	06:22:05	1.93	5.69	2048	22	3	15
f4	82	06:22:00	06:26:39	1.93	5.69	2048	22	3	15
f5	82	06:26:34	06:31:06	1.93	5.69	2048	85	3	15
f6	82	06:30:55	06:36:06	1.93	5.69	2048	30	3	15

DATA SET: MSCL.wat1

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	82	04:17:34	04:23:15	1.93	5.69	2048	18	3	15
f2	82	04:23:10	04:28:51	1.93	5.69	2048	20	3	15
f3	82	04:28:46	04:34:26	1.93	5.69	2048	11	3	15
f4	82	04:34:21	04:36:26	1.93	5.69	2048	10	3	15

DATA SET: MSCL.wat2

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	80	20:16:44	20:23:02	1.93	5.70	2048	32	2	19
f2	80	20:22:57	20:27:15	1.93	5.70	2048	762	2	19

DATA SET: MSCL.ice1

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	83	07:23:13	07:28:26	1.93	5.69	2048	28	4	16
f2	83	07:28:21	07:32:59	1.93	5.69	2048	55	4	16
f3	83	07:32:54	07:36:53	1.93	5.69	2048	29	4	16
f4	83	07:36:48	07:40:45	1.93	5.69	2048	38	4	16
f5	83	07:40:40	07:44:29	1.93	5.69	2048	109	4	16
f6	83	07:44:24	07:48:14	1.93	5.69	2048	68	4	16

DATA SET: MSCL.ice2

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	83	04:59:06	05:02:11	1.93	5.69	2048	20	4	16
f2	83	05:02:06	05:05:10	1.93	5.69	2048	26	4	16
f3	83	05:05:05	05:08:05	1.93	5.69	2048	16	4	16
f4	83	05:08:00	05:11:01	1.93	5.69	2048	26	4	16
f5	83	05:10:56	05:13:53	1.93	5.69	2048	12	4	16
f6	83	05:13:48	05:16:44	1.93	5.69	2048	23	4	16
f7	83	05:16:39	05:19:35	1.93	5.69	2048	20	4	16
f8	83	05:19:30	05:22:25	1.93	5.69	2048	15	4	16

DATA SET: MSCL.ice3

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	86	16:33:00	16:39:01	1.93	5.70	2048	31	5	19
f2	86	16:38:56	16:45:27	1.93	5.70	2048	25	5	19
f3	86	16:45:22	16:52:08	1.93	5.70	2048	25	5	19
f4	86	16:52:03	16:58:06	1.93	5.70	2048	27	5	19
f5	86	16:58:01	17:04:03	1.93	5.70	2048	20	5	19
f6	86	17:03:58	17:10:00	1.93	5.70	2048	25	5	19
f7	86	17:09:55	17:13:25	1.93	5.70	2048	878	5	19

DATA SET: LIMEX.edge

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f1	88	16:03:02	16:05:49	1.845	5.645	2048	16	6	17
f2	88	16:05:44	16:08:33	1.845	5.645	2048	23	6	17
f3	88	16:08:28	16:11:20	1.845	5.645	2048	18	6	17
f4	88	16:11:15	16:13:58	1.845	5.645	2048	18	6	17
f5	88	16:13:53	16:16:36	1.845	5.645	2048	13	6	17
f6	88	16:16:31	16:19:14	1.845	5.645	2048	15	6	17
f7	88	16:19:09	16:21:53	1.845	5.645	2048	22	6	17
f8	88	16:21:48	16:24:30	1.845	5.645	2048	17	6	17
f9	88	16:24:25	16:27:07	1.845	5.645	2048	12	6	17
f10	88	16:27:02	16:29:45	1.845	5.645	2048	17	6	17
f11	88	16:29:40	16:32:22	1.845	5.645	2048	18	6	17
f12	88	16:32:17	16:35:00	1.845	5.645	2048	23	6	17
f13	88	16:34:55	16:37:38	1.845	5.645	2048	15	6	17
f14	88	16:37:33	16:40:16	1.845	5.645	2048	12	6	17
f15	88	16:40:11	16:42:54	1.845	5.645	2048	16	6	17
f16	88	16:42:49	16:45:32	1.845	5.645	2048	20	6	17

DATA SET: LIMEX.edge

IMAGE NAME	JULIAN DAY	GMT START	GMT STOP	GAIN	OFF-SET	NUMBER OF LINES	NUMBER OF LINES LOST	TAPE NO ANA.	TAPE NO DIG.
f17	88	16:45:27	16:48:10	1.845	5.645	2048	19	6	18
f18	88	16:48:05	16:50:36	1.845	5.645	2048	22	6	18
f19	88	16:50:31	16:52:51	1.845	5.645	2048	11	6	18
f20	88	16:52:45	16:56:19	1.845	5.645	2048	25	6	18
f21	88	16:56:14	16:59:45	1.845	5.645	2048	16	6	18
f22	88	16:59:40	17:03:17	1.845	5.645	2048	16	6	18
f23	88	17:03:12	17:06:50	1.845	5.645	2048	27	6	18
f24	88	17:06:45	17:10:20	1.845	5.645	2048	24	6	18
f25	88	17:10:15	17:13:47	1.845	5.645	2048	23	6	18
f26	88	17:13:42	17:16:41	1.845	5.645	2048	19	6	18

APPENDIX B: INDEX TO dd TAPES

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
1	1	G10589.f1	82	05:08:17	05:10:45
	2	G10589.f2	82	05:10:43	05:13:17
	3	G10589.f3	82	05:13:10	05:15:26
	4	G10589.f4	82	05:15:21	05:17:39
	5	G10589.f5	82	05:17:34	05:20:00
	6	G10589.f6	82	05:19:55	05:22:20
	7	G10589.f7	82	05:22:15	05:24:41
	8	G10589.f8	82	05:24:36	05:27:00
	9	G10589.f9	82	05:26:55	05:29:18
1	10	G10589.f10	82	05:29:13	05:31:34
2	1	G10589.f11	82	05:31:29	05:33:49
	2	G10589.f12	82	05:33:44	05:36:04
	3	G10589.f13	82	05:35:59	05:38:21
	4	G10589.f14	82	05:38:16	05:40:39
	5	G10589.f15	82	05:40:34	05:43:00
	6	G10589.f16	82	05:42:55	05:45:23
	7	G10589.f17	82	05:45:18	05:47:49
	8	G10589.f18	82	05:47:44	05:50:10
	9	G10589.f19	82	05:50:05	05:52:32
2	10	G10589.f20	82	05:52:27	05:54:53

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
3	1	G10589.f21	82	05:54:48	05:56:00
	2	G10590.f1	82	07:12:22	07:15:03
	3	G10590.f2	82	07:14:57	07:19:15
	4	G10590.f3	82	07:19:10	07:24:45
	5	G10590.f4	82	07:24:40	07:30:22
	6	G10590.f5	82	07:30:17	07:35:54
	7	G10590.f6	82	07:35:49	07:41:23
	8	G10590.f7	82	07:41:18	07:46:52
3	9	G10590.f8	82	07:46:47	07:52:33
4	1	G10591.f1	82	09:26:30	09:38:44
	2	G10591.f2	82	09:28:39	09:31:01
	3	G10591.f3	82	09:30:56	09:33:51
	4	G10591.f4	82	09:33:46	09:37:00
	5	G10591.f5	82	09:36:55	09:40:00
	6	G10591.f6	82	09:39:55	09:43:29
	7	G10591.f7	82	09:43:24	09:46:56
	8	G10591.f8	82	09:46:51	09:49:52
	9	G10591.f9	82	09:49:47	09:52:38
4	10	G10591.f10	82	09:52:33	09:55:24

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
5	1	G10603.f1	83	05:17:00	05:19:55
	2	G10603.f2	83	05:19:50	05:22:46
	3	G10603.f3	83	05:22:41	05:25:31
	4	G10603.f4	83	05:25:26	05:28:08
	5	G10603.f5	83	05:28:03	05:30:34
	6	G10603.f6	83	05:30:29	05:33:00
	7	G10603.f7	83	05:32:55	05:35:25
	8	G10603.f8	83	05:35:20	05:37:50
	9	G10603.f9	83	05:37:45	05:40:14
5	10	G10603.f10	83	05:40:09	05:42:38
6	1	G10603.f11	83	05:42:33	05:45:01
	2	G10603.f12	83	05:44:56	05:47:25
	3	G10603.f13	83	05:47:20	05:49:49
	4	G10603.f14	83	05:49:44	05:52:12
	5	G10603.f15	83	05:52:07	05:54:35
	6	G10603.f16	83	05:54:30	05:56:57
	7	G10603.f17	83	05:56:52	05:59:20
	8	G10603.f18	83	05:59:15	06:01:42
	9	G10603.f19	83	06:01:37	06:04:02
6	10	G10603.f20	83	06:03:57	06:06:22

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
7	1	G10603.f21	83	06:06:17	06:08:42
	2	G10603.f22	83	06:08:37	06:11:05
	3	G10603.f23	83	06:11:00	06:14:22
	4	G10604.f1	83	06:33:19	06:36:57
	5	G10604.f2	83	06:36:52	06:39:24
	6	G10604.f3	83	06:39:10	06:42:55
	7	G10604.f4	83	06:42:50	06:46:02
	8	G10604.f5	83	06:45:57	06:48:59
7	9	G10604.f6	83	06:48:54	06:51:55
8	1	G10604.f7	83	06:51:50	06:54:51
	2	G10604.f8	83	06:54:46	06:57:47
	3	G10604.f9	83	06:57:42	07:00:34
	4	G10604.f10	83	07:00:29	07:03:24
	5	G10604.f11	83	07:03:19	07:06:17
	6	G10604.f12	83	07:06:12	07:09:08
	7	G10604.f13	83	07:09:03	07:11:59
8	8	G10604.f14	83	07:11:54	07:14:49

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
9	1	G10605.f1	83	08:39:44	08:43:32
	2	G10605.f2	83	08:43:27	08:46:36
	3	G10605.f3	83	08:44:30	08:47:24
	4	G10605.f4	83	08:47:10	08:50:12
	5	G10605.f5	83	08:50:07	08:53:02
	6	G10605.f6	83	08:52:57	08:55:55
	7	G10605.f7	83	08:55:50	08:58:54
9	8	G10605.f8	83	08:58:41	09:01:24
10	1	G10632.f1	85	05:50:00	05:52:30
	2	G10632.f2	85	05:52:25	05:54:48
	3	G10632.f3	85	05:54:43	05:57:06
	4	G10632.f4	85	05:57:01	05:59:23
	5	G10632.f5	85	05:59:18	06:01:41
	6	G10632.f6	85	06:01:36	06:03:59
	7	G10632.f7	85	06:03:54	06:06:17
	8	G10632.f8	85	06:06:12	06:08:35
	9	G10632.f9	85	06:08:30	06:10:53
10	10	G10632.f10	85	06:10:48	06:13:12

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
11	1	G10632.f11	85	06:13:07	06:15:30
	2	G10632.f12	85	06:15:25	06:18:01
	3	G10632.f13	85	06:17:56	06:20:37
	4	G10632.f14	85	06:20:32	06:23:14
	5	G10632.f15	85	06:23:09	06:25:50
	6	G10632.f16	85	06:25:45	06:28:26
	7	G10632.f17	85	06:28:21	06:31:03
	8	G10632.f18	85	06:30:58	06:33:40
	9	G10632.f19	85	06:33:35	06:36:16
11	10	G10632.f20	85	06:36:11	06:38:52
12	1	G10632.f21	85	06:38:47	06:40:14
	2	G10633.f1	85	07:12:20	07:14:58
	3	G10633.f2	85	07:14:53	07:17:31
	4	G10633.f3	85	07:17:26	07:20:01
	5	G10633.f4	85	07:19:56	07:22:29
	6	G10633.f5	85	07:22:24	07:25:01
	7	G10633.f6	85	07:24:56	07:27:33
	8	G10633.f7	85	07:27:28	07:30:11
	9	G10633.f8	85	07:30:06	07:32:51
12	10	G10633.f9	85	07:32:46	07:35:31

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
13	1	G10633.f10	85	07:35:26	07:38:11
	2	G10633.f11	85	07:38:06	07:40:51
	3	G10633.f12	85	07:40:46	07:43:31
	4	G10634.f1	85	09:14:00	09:16:49
	5	G10634.f2	85	09:16:44	09:19:16
	6	G10634.f3	85	09:19:11	19:21:39
	7	G10634.f4	85	09:21:34	09:23:59
	8	G10634.f5	85	09:23:54	09:26:20
	9	G10634.f6	85	09:26:15	09:28:41
13	10	G10634.f7	85	09:28:36	09:31:03
14	1	LIMEX1.f1	80	16:52:30	16:55:15
	2	LIMEX1.f2	80	16:55:10	16:57:54
	3	LIMEX1.f3	80	16:57:49	17:00:34
	4	LIMEX1.f4	80	17:00:29	17:03:14
	5	LIMEX1.f5	80	17:03:09	17:05:55
	6	LIMEX2.f1	80	17:09:00	17:11:47
	7	LIMEX2.f2	80	17:11:42	17:14:18
	8	LIMEX2.f3	80	17:14:13	17:16:40
	9	LIMEX2.f4	80	17:16:35	17:19:03
14	10	LIMEX2.f5	80	17:18:58	17:21:26

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
15	1	LIMEX3.f1	80	17:23:00	17:26:02
	2	LIMEX3.f2	80	17:25:57	17:29:11
	3	LIMEX3.f3	80	17:29:06	17:32:21
	4	LIMEX4.f1	80	17:44:30	17:47:18
	5	LIMEX4.f2	80	17:47:13	17:50:02
	6	LIMEX5.f1	80	17:52:00	17:54:50
	7	LIMEX5.f2	80	17:54:45	17:57:36
15	8	LIMEX5.f3	80	17:57:31	18:00:23
16	1	MSCL.gla1.f1	82	06:08:19	06:12:56
	2	MSCL.gla1.f2	82	06:12:51	06:17:31
	3	MSCL.gla1.f3	82	06:17:26	06:22:05
	4	MSCL.gla1.f4	82	06:22:00	06:26:39
	5	MSCL.gla1.f5	82	06:26:34	06:31:06
16	6	MSCL.gla1.f6	82	06:30:55	06:36:06

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
17	1	MSCL.wat1.f1	82	04:17:34	04:23:15
	2	MSCL.wat1.f2	82	04:23:10	04:28:51
	3	MSCL.wat1.f3	82	04:28:46	04:34:26
	4	MSCL.wat1.f4	82	04:34:21	04:36:26
	5	MSCL.wat2.f1	80	20:16:44	20:23:02
	6	MSCL.wat2.f2	80	20:22:57	20:27:15
	7	MSCL.wat3.f1*	82	09:46:51	09:49:52
	8	MSCL.wat3.f2*	82	09:49:47	09:52:38
17	9	MSCL.wat3.f3*	82	09:52:33	09:55:24

*These 3 files are identical to G10591.f8 through G10591.f10.

18	1	MSCL.ice1.f1	83	07:23:13	07:28:26
	2	MSCL.ice1.f2	83	07:28:21	07:32:59
	3	MSCL.ice1.f3	83	07:32:54	07:36:33
	4	MSCL.ice1.f4	83	07:36:48	07:40:45
	5	MSCL.ice1.f5	83	07:40:40	07:44:29
18	6	MSCL.ice1.f6	83	07:44:24	07:48:14

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
19	1	MSCL.ice2.f1	83	04:59:06	05:02:11
	2	MSCL.ice2.f2	83	05:02:06	05:05:10
	3	MSCL.ice2.f3	83	05:05:05	05:08:05
	4	MSCL.ice2.f4	83	05:08:00	05:11:01
	5	MSCL.ice2.f5	83	05:10:56	05:13:53
	6	MSCL.ice2.f6	83	05:13:48	05:16:44
	7	MSCL.ice2.f7	83	05:16:39	05:19:35
19	8	MSCL.ice2.f8	83	05:19:30	05:22:25
20	1	MSCL.ice3.f1	86	16:33:00	16:39:01
	2	MSCL.ice3.f2	86	16:38:56	16:45:27
	3	MSCL.ice3.f3	86	16:45:22	16:52:08
	4	MSCL.ice3.f4	86	16:52:03	16:58:06
	5	MSCL.ice3.f5	86	16:58:01	17:04:03
	6	MSCL.ice3.f6	86	17:03:58	17:10:00
20	7	MSCL.ice3.f7	86	17:09:55	17:13:25

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
21	1	LIMEX.edge.f1	88	16:03:02	16:05:49
	2	LIMEX.edge.f2	88	16:05:44	16:08:33
	3	LIMEX.edge.f3	88	16:08:28	16:11:20
	4	LIMEX.edge.f4	88	16:11:15	16:13:58
	5	LIMEX.edge.f5	88	16:13:53	16:16:36
	6	LIMEX.edge.f6	88	16:16:31	16:19:14
	7	LIMEX.edge.f7	88	16:19:09	16:21:53
	8	LIMEX.edge.f8	88	16:21:48	16:24:30
	9	LIMEX.edge.f9	88	16:24:25	16:27:07
21	10	LIMEX.edge.f10	88	16:27:02	16:29:45
22	1	LIMEX.edge.f11	88	16:29:40	16:32:22
	2	LIMEX.edge.f12	88	16:32:17	16:35:00
	3	LIMEX.edge.f13	88	16:34:55	16:37:38
	4	LIMEX.edge.f14	88	16:37:33	16:40:16
	5	LIMEX.edge.f15	88	16:40:11	16:42:54
	6	LIMEX.edge.f16	88	16:42:49	16:45:32
	7	LIMEX.edge.f17	88	16:45:27	16:48:10
	8	LIMEX.edge.f18	88	16:48:05	16:50:36
	9	LIMEX.edge.f19	88	16:50:31	16:52:51
22	10	LIMEX.edge.f20	88	16:52:45	16:56:19

TAPE NO	FILE NO	FILE NAME	JULIAN DAY	GMT START	GMT STOP
23	1	LIMEX.edge.f21	88	16:56:14	16:59:45
	2	LIMEX.edge.f22	88	16:59:40	17:03:17
	3	LIMEX.edge.f23	88	17:03:12	17:06:50
	4	LIMEX.edge.f24	88	17:06:45	17:10:20
	5	LIMEX.edge.f25	88	17:10:15	17:13:47
23	6	LIMEX.edge.f26	88	17:13:42	17:16:41

APPENDIX C: PROGRAM NAVDAT

```
#include        <math.h>
#include        <stdio.h>

struct footprint {
        float glat;     /* latitude of midpoint of GEOSAT footprint */
        float glon;     /* longitude of midpoint of GEOSAT footprint */
        float p3lat;    /* interpolated P-3 latitude associated with footprint */
        float p3lon;    /* P-3 longitude associated with footprint (always equal to glon) */
        float mfptime;  /* time P-3 was at footprint midpoint (dec. hrs.) */
        float bfptime;  /* P-3 time of beginning of footprint */
        float efptime;  /* P-3 time of end of footprint */
        int bimagen;    /* image number in which footprint begins */
        int blinen;     /* first line of image */
        int eimagen;    /* image number in which footprint ends */
        int elinen;     /* last line of image */
}

main()
{
        navdat();
}

/*********************************************************/
/*                                                       */
/*      FFFFF  CCC  RRRR  EEEEE   A   TTTTT EEEEE       */
/*      F     C   C R   R E      A A    T   E           */
/*      FFF   C     RRRR  EEE   A   A   T   EEE */
/*      F     C   C R R   E     AAAAA   T   E           */
/*      F      CCC  R  R  EEEEE A   A   T   EEEEE       */
/*                                                       */
/*********************************************************/
FILE *fcreate()         /* create a file on disk */
{
        FILE *stream;           /* file pointer to I/O device */
        FILE *fopen();          /* function to open files */

        char fname[20];         /* file name */

        /* enter file name */
        scanf("%s", fname);

        /* open file */
        if ( (stream = fopen(fname, "w+")) != NULL)
                return(stream);

        else {
                printf("FOPENE: fopen() failed\n");
                perror("FOPENE");
                exit(-1);
        }
}

/*************************************************/
/*                                               */
/*      FFFFF  OOO  PPPP  EEEEE N   N EEEEE     */
/*      F     O   O P   P E     NN  N E         */
/*      FFF   O   O PPPP  EEE   N N N EEE       */
/*      F     O   O P     E     N  NN E         */
/*      F      OOO  P     EEEEE N   N EEEEE     */
/*                                               */
/*************************************************/
FILE *fopene()          /* open file on disk */
{
        FILE *stream;           /* file pointer to I/O device */
        FILE *fopen();          /* function to open files */

        char fname[20];         /* file name */

        /* enter file name */
        scanf("%s", fname);

        /* open file */
        if( (stream = fopen(fname, "r")) != NULL)
                return(stream);

        else {
                printf("FOPENE: fopen() failed\n");
                perror("FOPENE");
                exit(-1);
```

```
        }
}

/*****************************************************************/
/*                                                               */
/*       GGG  EEEEE TTTTT FFFFF L      OOO    A   TTTTT         */
/*      G     E       T   F     L     O   O  A A    T           */
/*      G GGG EEE     T   FFF   L     O   O A   A   T           */
/*      G   G E       T   F     L     O   O AAAAA   T           */
/*       GGG  EEEEE   T   F     LLLLL  OOO  A   A   T           */
/*                                                               */
/*****************************************************************/
double getfloat(file)   /* read one 'float' value from disk */

FILE *file;             /* pointer to file */

{
        int nitems;     /* status returned by XENIX fread() */
        double value;   /* value to read from disk */

        fread(&value, 8, 1, file);
/*
        if ( (nitems = fread(&value, 8, 1, file)) != 1) {
                printf("\nGETFLOAT: feof(file) = %d\n", feof(file));
                printf("\nGETFLOAT: error on read from disk\n");
                perror("FREAD");
                exit(-1);
        }
        printf("GETFLOAT: value = %lf\n", value);
*/

        return(value);
}

/*****************************************************************/
/*                                                               */
/*       GGG  EEEEE TTTTT III TTTTT III M   M EEEEE             */
/*      G     E       T    I    T    I  MM MM E                 */
/*      G GGG EEE     T    I    T    I  M M M EEE               */
/*      G   G E       T    I    T    I  M   M E                 */
/*       GGG  EEEEE   T   III   T   III M   M EEEEE             */
/*                                                               */
/*****************************************************************/
getitime(file, p_btime, p_etime, p_sph, p_imn)  /* read image times */

FILE *file;             /* pointer to disk file */
float *p_btime;         /* pointer to beginning time of image */
float *p_etime;         /* pointer to ending time of image */
float *p_sph;           /* pointer to scans-per-hour constant for image */
int *p_imn;             /* pointer to image number */
{
        double getfloat();      /* function to read one 'float' value */

        *p_btime = (float)getfloat(file);
        *p_etime = (float)getfloat(file);
        *p_sph = 2048. / (*p_etime - *p_btime);
        (*p_imn)++;
/*
        printf("GETTIMES: bimtime=%f eimtime=%f imn=%d\n", *p_btime, *p_etime, *p_imn);
*/
}

/*****************************************************************/
/*                                                               */
/*       GGG  EEEEE TTTTT PPPP   SSS  TTTTT   A   TTTTT         */
/*      G     E       T   P   P S       T    A A    T           */
/*      G GGG EEE     T   PPPP   SSS    T   A   A   T           */
/*      G   G E       T   P         S   T   AAAAA   T           */
/*       GGG  EEEEE   T   P      SSS    T   A   A   T           */
/*                                                               */
/*****************************************************************/
getpstat(p3a, p3b, p_hpdlon, p_dlatph)  /* generate P-3 constants */

float p3a[3];           /* first set of P-3 coordinates */
float p3b[3];           /* next set of P-3 coordinates */
float *p_hpdlon;        /* pointer to hours-per-degree-longitude constant */
float *p_dlatph;        /* pointer to degrees-latitude-per-hour constant */

{
        /* compute hours per degree longitude for interval between p3 coords */
        *p_hpdlon = fabs( (double)( (p3b[2] - p3a[2]) / (p3b[1] - p3a[1]) ) );

        /* now compute degrees latitude per hour for this same interval */
        *p_dlatph = fabs( (double)( (p3b[0] - p3a[0]) / (p3b[2] - p3a[2]) ) );
}
```

```
/*************************************************/
/*                                               */
/*      N   N   A   V   V DDDD    A   TTTTT      */
/*      NN  N  A A  V   V D   D  A A    T        */
/*      N N N A   A  V V  D   D A   A   T        */
/*      N  NN AAAAA  V V  D   D AAAAA   T        */
/*      N   N A   A   V   DDDD  A   A   T        */
/*                                               */
/*************************************************/
navdat()        /* relate image lines to footprints using navigation data */
{
        struct footprint f[100];        /* structure holding footprint data */

        float bimtime;  /* beginning time for the current image */
        float dlatph;   /* degress latitude per hour constant */
        float eimtime;  /* ending time for the current image */
        char eof;       /* end-of-file flag */
        double fabs();  /* XENIX absolute value function */
        FILE *fcreate();        /* function to create files */
        FILE *fopene(); /* function to open files */
        FILE *geosat;   /* file pointer to GEOSAT coordinate file */
        double getfloat();      /* funtion to read one "float" value from disk */
        float hpdlon;   /* hours-per-degree of longitude constant */
        int i;          /* loop index */
        FILE *images;   /* file pointer to file containing image times */
        int imn;        /* current image number */
        int n;          /* loop index on current footprint */
        int nfp;        /* index on f[*] (number of footprints processed) */
        int n1;         /* first footprint corresponding to an image */
        FILE *output;   /* file pointer to output file of interpolated data */
        FILE *p3;       /* file pointer to file containing P-3 coordinates */
        float p3a[3];   /* first set of P-3 lat-lon-time coords from flight logs */
        float p3b[3];   /* next set of P-3 lat-lon-time coords from flight logs */
        float sph;      /* scans per hour constant */

        n = 0;

        /* send directory list to CRT screen */
        printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");
        printf("Current files in /u/eppler/pgms.GEOSAT:\n");
        system("lc");
        printf("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");

        /* open files */
        printf("\nOPEN FILE OF GEOSAT COORDINATES: ");
        geosat = fopene();
        printf("\nOPEN FILE OF P-3 COORDINATES: ");
        p3 = fopene();
        printf("\nOPEN FILE OF IMAGE TIMES: ");
        images = fopene();

        /* create output file */
        printf("\nCREATE OUTPUT FILE: ");
        output = fcreate();

        /* print contents of geosat, p3, and images files */
        prntg();
        prnti();
        prntp();

        printf("\nread first two sets of p3 data\n");
        /* read first two sets of p3 coordinates */
        for (i = 0; i <= 2; i++)
                p3a[i] = (float)getfloat(p3);
        for (i = 0; i <= 2; i++)
                p3b[i] = (float)getfloat(p3);

        /* exit if P-3 longitudes increase along track */
        if (p3a[1] < p3b[1]) {
                printf("\nP-3 longitudes increase.  Run NAVDATI instead.\n");
                exit(0);
        }
/*
        ****** use these lines for NAVDATI ******
        exit if P-3 longitudes decrease along track
        if (p3a[1] > p3b[1]) {
                printf("\nP-3 longitudes decrease.  Run NAVDATD instead.\n");
                exit(0);
        }
        ****************************************
*/
```

```
        printf("read first set of geosat coordinates\n");
        /* read first set of geosat coordinates */
        do {
                f[0].glat = (float)getfloat(geosat);
                f[0].glon = (float)getfloat(geosat);
        } while (p3a[1] < f[0].glon);
/*
        ****** use this line for NAVDATI ******
        } while (f[0].glon < p3a[1]);
        ****************************************
*/

        printf("begin first interpolation loop\n");
        /* interpolate P-3 times and latitudes that correspond with GEOSAT footprints */
        while (!eof) {          /* loop until all P-3 coordinates are processed */
                /* get P-3 interval constants */
                getpstat(p3a, p3b, &hpdlon, &dlatph);

                while (p3b[1] <= f[n].glon) {
/*
        ****** use this line for NAVDATI *****
                while (f[n].glon <= p3b[1]) {
        ****************************************
*/
                        /* interpolate geosat times for footprint midpoints */
                        f[n].mfptime = (hpdlon * (fabs( (double)(f[n].glon - p3a[1]) ) ) ) + p3a[2];

                        /* interpolate P-3 latitude */
                        f[n].p3lat = (dlatph * (fabs ( (double)(f[n].mfptime - p3a[2]) ) ) ) + p3a[0];

                        f[n].p3lon = f[n].glon;

                        /* read next set of geosat coordinates */
                        n++;
                        f[n].glat = (float)getfloat(geosat);
                        f[n].glon = (float)getfloat(geosat);
                        if (feof(geosat))       break;
                }

                /* read next set of p3 coordinates */
                printf("--------------------\n");
                for (i = 0; i <= 2; i++) {
                        p3a[i] = p3b[i];
                        p3b[i] = (float)getfloat(p3);
                }
                printf("--------------------\n");
                if(feof(p3))    break;

        }

        nfp = n - 1;
        for (i = 0; i <= nfp; i++)
                printf("i=%d glat=%f glon=%f plat=%f plon=%f ptime=%f\n", i, f[i].glat, f[i].glon,
                f[i].p3lat, f[i].p3lon, f[i].mfptime);
        /* find P-3 times that correspond to beginning and end of footprints */
        for (n = 1; n <= nfp; n++)
                f[n-1].efptime = f[n].bfptime = ( (f[n].mfptime - f[n-1].mfptime) / 2.) + f[n-1].mfptime;
        f[n].efptime = f[n].mfptime + (f[n].mfptime - f[n].bfptime);
        f[0].bfptime = f[0].mfptime - (f[0].efptime - f[0].mfptime);

        for (n = 0; n <= nfp; n++)
/*
                printf("NAVDAT: n=%d f[n].bfptime=%f f[n].mfptime=%f f[n].efptime=%f\n", n, f[n].bfptime,
                f[n].mfptime, f[n].efptime
);
*/

        /* find the image that contains start of first footprint */
        imn = 0;
        n = 0;
        do {    /* case of image end occurring earlier than footprint */
/*
                printf("NAVDAT: imn=%d f[0].bfptime=%f\n", imn, f[0].bfptime);
*/
                getitime(images, &bimtime, &eimtime, &sph, &imn);
        } while (f[0].bfptime >= eimtime);

do {    /* case of image end occurring earlier than footprint */

        printf("NAVDAT: imn=%d f[0].bfptime=%f\n", imn, f[0].bfptime);

        getitime(images, &bimtime, &eimtime, &sph, &imn);
} while (f[0].bfptime >= eimtime);

/* case of image beginning occurring later than footprint */
while (bimtime >= f[n].efptime) {
        if (++n > nfp) {
```

```
                        printf("\nERROR: footprints outside range of image times\n");
                        exit(-1);
                }
        }
        n1 = n + 1;

        /* assign image bounds to footprints */
        for (n; n <= nfp; n++) {

                /* compute beginning line number for current footprint */
                if (f[n].bfptime > eimtime) {   /* read times for next image */
                        if (feof(images)) {
                                putdata(output, n1, f, nfp);    /* write data to disk */
                                exit(0);
                        }
                        else
                                getitime(images, &bimtime, &eimtime, &sph, &imn);
                }

                f[n].blinen = sph * (f[n].bfptime - bimtime);
                f[n].bimagen = imn;

                /* compute ending line number for current footprint */
                if (f[n].efptime > eimtime) {   /* read times for next image */
                        if (feof(images)) {
                                putdata(output, n1, f, nfp);    /* write data to disk */
                                exit(0);
                        }
                        else
                                getitime(images, &bimtime, &eimtime, &sph, &imn);
                }

                f[n].elinen = sph * (f[n].efptime - bimtime);
                f[n].eimagen = imn;
        }

        /* write data to disk */
        putdata(output, n1, f, nfp);
}

/*********************************************************/
/*                                                       */
/*      PPPP  U   U TTTTT DDDD    A   TTTTT   A          */
/*      P   P U   U   T   D   D  A A    T    A A         */
/*      PPPP  U   U   T   D   D A   A   T   A   A        */
/*      P     U   U   T   D   D AAAAA   T   AAAAA        */
/*      P      UUU    T   DDDD  A   A   T   A   A        */
/*                                                       */
/*********************************************************/
putdata(file, n1, f, nfp)               /* write footprint data to disk */

FILE *file;             /* pointer to output file */
int n1;                 /* first footprint that corresponds to an image */
struct footprint f[100];        /* structure that holds footprint data */
int nfp;                /* number of footprints */

{
        int i;
        int n;          /* loop counter */
        double value;

        i = 1;

/* print heading */
printf("\f");
printf("\n       GEOSAT FOOTPRINT      P-3 EQUIVALENT    P-3 FOOTPRINT TIMES   START     STOP    \n");

printf(" -N-  LATITUDE LONGITUDE   LATITUDE LONGITUDE   START  MIDPOINT  STOP IMAGE LINE IMAGE LINE\n\n");

/* print data and write it to disk */
for (n = n1; n <= nfp; n++) {

        /* print data */
        printf("%3d   ", i++);
        printf("%6.2f    %6.2f     ", f[n].glat, f[n].glon);
        printf("%6.2f    %6.2f    ", f[n].p3lat, f[n].p3lon);
        printf("%8.4f %8.4f %8.4f   ", f[n].bfptime, f[n].mfptime, f[n].efptime);
        printf("%2d   %4d    ", f[n].bimagen, f[n].blinen);
        printf("%2d   %4d\n", f[n].eimagen, f[n].elinen);

        /* write data to disk */
        value = (double)f[n].glon;
        fwrite(&value, 8, 1, file);
        value = (double)f[n].glat;
```

```
        fwrite(&value, 8, 1, file);
        value = (double)f[n].p3lat;
        fwrite(&value, 8, 1, file);
        value = (double)f[n].p3lon;
        fwrite(&value, 8, 1, file);
        value = (double)f[n].mfptime;
        fwrite(&value, 8, 1, file);
        value = (double)f[n].bfptime;
        fwrite(&value, 8, 1, file);
        value = (double)f[n].efptime;
        fwrite(&value, 8, 1, file);
        fwrite(&f[n].bimagen, 2, 1, file);
        fwrite(&f[n].blinen, 2, 1, file);
        fwrite(&f[n].eimagen, 2, 1, file);
        fwrite(&f[n].elinen, 2, 1, file);
}
```

APPENDIX D: PROGRAM ENTERG

```
#include        stdio.h

main()
{
        enterg();
}

/*************************************************/
/*                                               */
/*      EEEEE N   N TTTTT EEEEE RRRR   GGG       */
/*      E     NN  N   T   E     R   R G          */
/*      EEE   N N N   T   EEE   RRRR  G GGG      */
/*      E     N  NN   T   E     R R   G   G      */
/*      EEEEE N   N   T   EEEEE R  R   GGG       */
/*                                               */
/*************************************************/
enterg()        /* create a disk file of GEOSAT footprint coordinates */
{
        FILE *fcreate();        /* function to create a file on disk */
        FILE *geosat;           /* pointer to disk file of coordinates */
        float lat;              /* latitude coordinate */
        float lon;              /* longitude coordinate */
        int n;                  /* running count of coordinates entered */

        n = 1;

        /* create disk file */
        printf("\nCREATE OUTPUT FILE FOR GEOSAT COORDINATES:\n");
        geosat = fcreate();

        /* enter coordinates from keyboard */
        printf("\n\nENTER COORDINATES AT PROMPT (enter 999. to stop)\n");
        for (;;) {
                printf("\nCOORDINATE PAIR %d:\n", n);

                printf("\tLATITUDE = ");
                scanf("%f", &lat);
                if (lat == 999.)        exit(0);

                printf("\tLONGITUDE = ");
                scanf("%f", &lon);
                if (lon == 999.)        exit(0);

                putfloat(lat, geosat);
                putfloat(lon, geosat);
                n++;
        }
}

/*********************************************************/
/*                                                       */
/*      FFFFF  CCC  RRRR  EEEEE   A   TTTTT EEEEE       */
/*      F     C   C R   R E      A A    T   E           */
/*      FFF   C     RRRR  EEE   A   A   T   EEE */
/*      F     C   C R R   E     AAAAA   T   E           */
/*      F      CCC  R  R  EEEEE A   A   T   EEEEE       */
/*                                                       */
/*********************************************************/
FILE *fcreate()         /* create a file on disk */
{
        FILE *stream;           /* file pointer to I/O device */
        FILE *fopen();          /* function to open files */

        char fname[20];         /* file name */
        /* send directory list to CRT screen */
        printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");
        printf("Current files in /u/eppler/pgms.GEOSAT:\n");
        system("lc");
        printf("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");

        /* enter file name */
        printf("Well, matey, which file will it be?:");
        scanf("%s", fname);

        /* open file */
        if ( (stream = fopen(fname, "w+")) != NULL)
                return(stream);

        else {
                printf("FOPENE: fopen() failed\n");
                perror("FOPENE");
                exit(-1);
```

```
        }
}

/*********************************************************************/
/*                                                                   */
/*      PPPP  U   U TTTTT FFFFF L      OOO    A   TTTTT             */
/*      P   P U   U   T   F     L     O   O  A A    T               */
/*      PPPP  U   U   T   FFF   L     O   O A   A   T               */
/*      P     U   U   T   F     L     O   O AAAAA   T               */
/*      P      UUU    T   F     LLLLL  OOO  A   A   T               */
/*                                                                   */
/*********************************************************************/
putfloat(value, file)           /* write floating point value to disk */

double value;           /* value to write to disk */
FILE *file;             /* pointer to disk file */

{
        int nitems;     /* status returned by XENIX fwrite() */

        if ( (nitems = fwrite(&value, 8, 1, file)) != 1) {
                printf("\nPUTFLOAT: error on write to disk;\n");
                perror("FWRITE");
                exit(-1);
        }
}
```

APPENDIX E: PROGRAM ENTERP

```
#include <stdio.h>

main()
{
        enterp();
}

/*************************************************/
/*                                               */
/*      EEEEE N   N TTTTT EEEEE RRRR  PPPP       */
/*      E     NN  N   T   E     R   R P   P      */
/*      EEE   N N N   T   EEE   RRRR  PPPP       */
/*      E     N  NN   T   E     R R   P          */
/*      EEEEE N   N   T   EEEEE R  R  P          */
/*                                               */
/*************************************************/
enterp()        /* create a disk file of P-3 footprint coordinates */
{
        FILE *fcreate();        /* function to create a file on disk */
        FILE *p3;               /* pointer to disk file of coordinates */
        float lat;              /* latitude coordinate */
        float lon;              /* longitude coordinate */
        int n;                  /* running count of coordinates entered */
        float time;             /* time coordinate */

        n = 1;

        /* create disk file */
        printf("\nCREATE OUTPUT FILE FOR P3 COORDINATES:\n");
        p3 = fcreate();

        /* enter coordinates from keyboard */
        printf("\n\nENTER COORDINATES AT PROMPT (enter 999. to stop)\n");
        for (;;) {
                printf("\nCOORDINATE TRIPLET %d:\n", n);

                printf("\tLATITUDE = ");
                scanf("%f", &lat);
                if (lat == 999.)        exit(0);

                printf("\tLONGITUDE = ");
                scanf("%f", &lon);
                if (lon == 999.)        exit(0);

                printf("\tTIME = ");
                scanf("%f", &time);
                if (time == 999.)       exit(0);

                putfloat(lat, p3);
                putfloat(lon, p3);
                putfloat(time, p3);
                n++;
        }
}

/*************************************************************/
/*                                                           */
/*      FFFFF  CCC  RRRR  EEEEE   A   TTTTT EEEEE           */
/*      F     C   C R   R E      A A    T   E               */
/*      FFF   C     RRRR  EEE   A   A   T   EEE  */
/*      F     C   C R R   E     AAAAA   T   E               */
/*      F      CCC  R  R  EFEEE A   A   T   EEEEE           */
/*                                                           */
/*************************************************************/
FILE *fcreate()         /* create a file on disk */
FILE *stream;           /* file pointer to I/O device */
FILE *fopen();          /* function to open files */

char fname[20];         /* file name */

/* send directory list to CRT screen */
printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");
printf("Current files in /u/eppler/pgms.GEOSAT(\n");
system("lc");
printf("++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");

/* enter file name */
printf("Well, matey, which file will it be?:");
scanf("%s", fname);
```

```
        /* open file */
        if ( (stream = fopen(fname, "wt")) != NULL)
                return(stream);

        else {
                printf("FOPENE: fopen() failed\n");
                perror("FOPENE");
                exit(-1);
        }
}

/*****************************************************************/
/*                                                               */
/*      PPPP  U   U TTTTT FFFFF L      OOO    A   TTTTT         */
/*      P   P U   U   T   F     L     O   O  A A    T           */
/*      PPPP  U   U   T   FFF   L     O   O A   A   T           */
/*      P     U   U   T   F     L     O   O AAAAA   T           */
/*      P      UUU    T   F     LLLLL  OOO  A   A   T           */
/*                                                               */
/*****************************************************************/
putfloat(value, file)           /* write floating point value to disk */

double value;           /* value to write to disk */
FILE *file;             /* pointer to disk file */

{
        int nitems;     /* status returned by XENIX fwrite() */

        if ( (nitems = fwrite(&value, 8, 1, file)) != 1) {
                printf("\nPUTFLOAT: error on write to disk;\n");
                perror("FWRITE");
                exit(-1);
        }
}
```

APPENDIX F: PROGRAM ENTERI

```
#include        <stdio.h>

main()
{
        enteri();
}

/************************************************/
/*                                              */
/*      EEEEE N   N TTTTT EEEEE RRRR  III       */
/*      E     NN  N   T   E     R   R  I        */
/*      EEE   N N N   T   EEE   RRRR   I        */
/*      E     N  NN   T   E     R R    I        */
/*      EEEEE N   N   T   EEEEE R  R  III       */
/*                                              */
/************************************************/
enteri()        /* create a disk file of KRMS image start-stop times */
{
        FILE *fcreate();        /* function to create a file on disk */
        FILE *images;           /* pointer to disk file of coordinates */
        float start;            /* image start time */
        float stop;             /* image stop time */
        int n;                  /* running count of images entered */

        n = 1;

        /* create disk file */
        printf("\nCREATE OUTPUT FILE FOR IMAGE TIMES:\n");
        images = fcreate();

        /* enter times from keyboard */
        printf("\n\nENTER TIMES AT PROMPT (enter 999. to stop)\n");
        for (;;) {
                printf("\nTIMES FOR IMAGE %d:\n", n);

                printf("\tSTART TIME = ");
                scanf("%f", &start);
                if (start == 999.)      exit(0);

                printf("\tSTOP TIME = ");
                scanf("%f", &stop);
                if (stop == 999.)       exit(0);

                putfloat(start, images);
                putfloat(stop, images);
                n++;
        }
}

/************************************************************/
/*                                                          */
/*      FFFFF  CCC  RRRR  EEEEE   A   TTTTT EEEEE          */
/*      F     C   C R   R E      A A    T   E              */
/*      FFF   C     RRRR  EEE   A   A   T   EEE */
/*      F     C   C R R   E     AAAAA   T   E              */
/*      F      CCC  R  R  EEEEE A   A   T   EEEEE          */
/*                                                          */
/************************************************************/
FILE *fcreate()         /* create a file on disk */
{
        FILE *stream;           /* file pointer to I/O device */
        FILE *fopen();          /* function to open files */

        char fname[20];         /* file name */
        /* send directory list to CRT screen */
        printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");
        printf("Current files in /u/eppler/pgms.GEOSAT\n");
        system("lc");
        printf("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");

        /* enter file name */
        printf("Well, matey, which file will it be?:");
        scanf("%s", fname);

        /* open file */
        if ( (stream = fopen(fname, "w+")) != NULL)
                return(stream);
```

```
        else {
                printf("FOPENE: fopen() failed\n");
                perror("FOPENE");
                exit(-1);
        }
}

/*********************************************************************/
/*                                                                   */
/*      FFFFF  U   U TTTTT FFFFF L       OOO    A   TTTTT          */
/*      F   F U   U   T   F     L      O   O  A A    T            */
/*      FFFF  U   U   T   FFF   L      O   O A   A   T            */
/*      F     U   U   T   F     L      O   O AAAAA   T            */
/*      F      UUU    T   F     LLLLL   OOO  A   A   T            */
/*                                                                   */
/*********************************************************************/
putfloat(value, file)           /* write floating point value to disk */

double value;           /* value to write to disk */
FILE *file;             /* pointer to disk file */

{
        int nitems;     /* status returned by XENIX fwrite() */

        if ( (nitems = fwrite(&value, 8, 1, file)) != 1) {
                printf("\nPUTFLOAT: error on write to disk;\n");
                perror("FWRITE");
                exit(-1);
        }
}
```

APPENDIX G: INDICES TO KRMS IMAGES THAT CORRESPOND TO GEOSAT FOOTPRINTS FOR SELECTED GEOSAT TRACKS

Track 10589.a

-N-	GEOSAT FOOTPRINT LATITUDE	GEOSAT FOOTPRINT LONGITUDE	P-3 EQUIVALENT LATITUDE	P-3 EQUIVALENT LONGITUDE	P-3 FOOTPRINT TIMES START	P-3 FOOTPRINT TIMES MIDPOINT	P-3 FOOTPRINT TIMES STOP	START IMAGE	START LINE	STOP IMAGE	STOP LINE
1	70.98	347.14	71.04	347.14	5.2228	5.2282	5.2337	3	179	3	770
2	71.00	346.97	71.06	346.97	5.2337	5.2391	5.2449	3	770	3	1382
3	71.01	346.79	71.08	346.79	5.2449	5.2508	5.2575	3	1382	4	87
4	71.03	346.62	71.10	346.62	5.2575	5.2641	5.2708	4	87	4	801
5	71.05	346.45	71.12	346.45	5.2708	5.2775	5.2846	4	801	4	1536
6	71.07	346.27	71.14	346.27	5.2846	5.2917	5.2983	4	1536	5	281
7	71.09	346.10	71.15	346.10	5.2983	5.3050	5.3121	5	281	5	975
8	71.11	345.92	71.17	345.92	5.3121	5.3192	5.3259	5	975	5	1670
9	71.12	345.75	71.19	345.75	5.3259	5.3325	5.3384	5	1670	6	329
10	71.14	345.57	71.21	345.57	5.3384	5.3443	5.3501	6	329	6	924
11	71.16	345.39	71.22	345.39	5.3501	5.3559	5.3614	6	924	6	1499
12	71.18	345.22	71.24	345.22	5.3614	5.3669	5.3727	6	1499	7	117
13	71.19	345.04	71.26	345.04	5.3727	5.3786	5.3840	7	117	7	682
14	71.21	344.87	71.27	344.87	5.3840	5.3895	5.3954	7	682	7	1247
15	71.23	344.69	71.29	344.69	5.3954	5.4012	5.4070	7	1247	7	1827
16	71.24	344.51	71.30	344.51	5.4070	5.4128	5.4188	7	1827	8	452
17	71.26	344.33	71.32	344.33	5.4188	5.4249	5.4310	8	452	8	1075
18	71.28	344.15	71.33	344.15	5.4310	5.4372	5.4430	8	1075	8	1687
19	71.29	343.98	71.35	343.98	5.4430	5.4488	5.4549	8	1687	9	325
20	71.31	343.80	71.36	343.80	5.4549	5.4611	5.4672	9	325	9	959
21	71.32	343.62	71.38	343.62	5.4672	5.4734	5.4795	9	959	9	1593
22	71.34	343.44	71.39	343.44	5.4795	5.4857	5.4918	9	1593	10	254

Track 10589.b

-N-	GEOSAT FOOTPRINT LATITUDE	GEOSAT FOOTPRINT LONGITUDE	P-3 EQUIVALENT LATITUDE	P-3 EQUIVALENT LONGITUDE	P-3 FOOTPRINT TIMES START	P-3 FOOTPRINT TIMES MIDPOINT	P-3 FOOTPRINT TIMES STOP	START IMAGE	START LINE	STOP IMAGE	STOP LINE
1	71.71	338.32	71.76	338.32	5.8201	5.8261	5.8321	18	1239	18	1846
2	71.72	338.13	71.77	338.13	5.8321	5.8382	5.8440	18	1846	19	464
3	71.73	337.94	71.78	337.94	5.8440	5.8498	5.8552	19	464	19	1028
4	71.74	337.76	71.79	337.76	5.8552	5.8607	5.8665	19	1028	19	1592
5	71.75	337.57	71.80	337.57	5.8665	5.8723	5.8780	19	1592	20	195
6	71.76	337.38	71.81	337.38	5.8780	5.8838	5.8896	20	195	20	779
7	71.77	337.19	71.82	337.19	5.8896	5.8954	5.9009	20	779	20	1347

Track 10603

-N-	GEOSAT FOOTPRINT LATITUDE	GEOSAT FOOTPRINT LONGITUDE	P-3 EQUIVALENT LATITUDE	P-3 EQUIVALENT LONGITUDE	P-3 FOOTPRINT TIMES START	P-3 FOOTPRINT TIMES MIDPOINT	P-3 FOOTPRINT TIMES STOP	START IMAGE	START LINE	STOP IMAGE	STOP LINE
1	71.56	349.51	71.57	349.51	5.3439	5.3545	5.3650	2	558	2	1443
2	71.57	349.33	71.57	349.33	5.3650	5.3756	5.3867	2	1443	3	376
3	71.59	349.14	71.58	349.14	5.3867	5.3979	5.4080	3	376	3	1298
4	71.60	348.96	71.54	348.96	5.4080	5.4181	5.4246	3	1298	3	2017
5	71.61	348.78	71.55	348.78	5.4246	5.4310	5.4379	3	2017	4	635
6	71.62	348.59	71.56	348.59	5.4379	5.4447	5.4512	4	635	4	1240
7	71.64	348.41	71.57	348.41	5.4512	5.4576	5.4644	4	1240	4	1845
8	71.65	348.22	71.59	348.22	5.4644	5.4713	5.4777	4	1845	5	499
9	71.66	348.04	71.60	348.04	5.4777	5.4842	5.4910	5	499	5	1148
10	71.67	347.85	71.61	347.85	5.4910	5.4978	5.5037	5	1148	5	1766
11	71.68	347.66	71.62	347.66	5.5037	5.5095	5.5149	5	1766	6	333
12	71.69	347.48	71.63	347.48	5.5149	5.5202	5.5259	6	333	6	870
13	71.70	347.29	71.64	347.29	5.5259	5.5315	5.5369	6	870	6	1408
14	71.72	347.11	71.65	347.11	5.5369	5.5423	5.5479	6	1408	6	1946
15	71.73	346.92	71.66	346.92	5.5479	5.5536	5.5592	6	1946	7	521
16	71.74	346.73	71.67	346.73	5.5592	5.5649	5.5702	7	521	7	1062
17	71.75	346.55	71.68	346.55	5.5702	5.5756	5.5817	7	1062	7	1625
18	71.76	346.36	71.69	346.36	5.5817	5.5878	5.5948	7	1625	8	288
19	71.77	346.17	71.71	346.17	5.5948	5.6018	5.6088	8	288	8	977
20	71.78	345.98	71.72	345.98	5.6088	5.6158	5.6224	8	977	8	1647
21	71.79	345.80	71.73	345.80	5.6224	5.6291	5.6361	8	1647	9	341
22	71.80	345.61	71.74	345.61	5.6361	5.6431	5.6501	9	341	9	1034
23	71.81	345.42	71.75	345.42	5.6501	5.6571	5.6639	9	1034	9	1718
24	71.82	345.23	71.77	345.23	5.6639	5.6707	5.6772	9	1718	10	395
25	71.82	345.04	71.79	345.04	5.6772	5.6836	5.6897	10	395	10	1016
26	71.83	344.86	71.82	344.86	5.6897	5.6958	5.7022	10	1016	10	1636
27	71.84	344.67	71.84	344.67	5.7022	5.7087	5.7151	10	1636	11	295
28	71.85	344.48	71.87	344.48	5.7151	5.7215	5.7280	11	295	11	937
29	71.86	344.29	71.89	344.29	5.7280	5.7344	5.7409	11	937	11	1578
30	71.87	344.10	71.91	344.10	5.7409	5.7473	5.7535	11	1578	12	192
31	71.87	343.91	71.92	343.91	5.7535	5.7598	5.7659	12	192	12	819
32	71.88	343.72	71.92	343.72	5.7659	5.7721	5.7783	12	819	12	1444

33	71.89	343.53	71.92	343.53	5.7783	5.7845	5.7907	12	1444	13	88
34	71.90	343.34	71.93	343.34	5.7907	5.7969	5.8031	13	88	13	701
35	71.90	343.15	71.93	343.15	5.8031	5.8092	5.8154	13	701	13	1313
36	71.91	342.96	71.93	342.96	5.8154	5.8216	5.8278	13	1313	13	1925
37	71.92	342.77	71.91	342.77	5.8278	5.8340	5.8401	13	1925	14	559
38	71.93	342.58	71.92	342.58	5.8401	5.8463	5.8524	14	559	14	1170
39	71.93	342.39	71.93	342.39	5.8524	5.8585	5.8647	14	1170	14	1782
40	71.94	342.20	71.94	342.20	5.8647	5.8708	5.8769	14	1782	15	414
41	71.94	342.01	71.95	342.01	5.8769	5.8831	5.8892	15	414	15	1026
42	71.95	341.82	71.95	341.82	5.8892	5.8953	5.9015	15	1026	15	1637
43	71.96	341.63	71.96	341.63	5.9015	5.9076	5.9137	15	1637	16	271
44	71.96	341.44	71.97	341.44	5.9137	5.9199	5.9263	16	271	16	900
45	71.97	341.24	71.98	341.24	5.9263	5.9327	5.9388	16	900	16	1527
46	71.97	341.05	71.98	341.05	5.9388	5.9449	5.9510	16	1527	17	158
47	71.98	340.86	71.98	340.86	5.9510	5.9571	5.9631	17	158	17	765
48	71.98	340.67	71.99	340.67	5.9631	5.9692	5.9753	17	765	17	1372
49	71.99	340.48	71.99	340.48	5.9753	5.9814	5.9875	17	1372	17	1978
50	71.99	340.29	72.00	340.29	5.9875	5.9936	6.0000	17	1978	18	625
51	72.00	340.09	72.00	340.09	6.0000	6.0064	6.0124	18	625	18	1249
52	72.00	339.90	72.00	339.90	6.0124	6.0184	6.0245	18	1249	18	1855

Track 10604

	GEOSAT FOOTPRINT		P-3 EQUIVALENT		P-3 FOOTPRINT TIMES			START		STOP	
-N-	LATITUDE	LONGITUDE	LATITUDE	LONGITUDE	START	MIDPOINT	STOP	IMAGE	LINE	IMAGE	LINE
1	70.18	337.07	70.25	337.07	6.7210	6.7290	6.7370	4	273	4	887
2	70.15	337.23	70.22	337.23	6.7370	6.7450	6.7519	4	887	4	1458
3	70.13	337.39	70.16	337.39	6.7519	6.7587	6.7651	4	1458	4	1965
4	70.10	337.55	70.18	337.55	6.7651	6.7714	6.7774	4	1965	5	467
5	70.08	337.70	70.21	337.70	6.7774	6.7833	6.7926	5	467	5	1085
6	70.05	337.86	70.07	337.86	6.7926	6.8019	6.8112	5	1085	5	1839
7	70.03	338.02	70.08	338.02	6.8112	6.8205	6.8292	5	1839	6	576
8	70.00	338.18	70.07	338.18	6.8292	6.8378	6.8449	6	576	6	1218
9	69.98	338.34	70.09	338.34	6.8449	6.8520	6.8587	6	1218	6	1781
10	69.95	338.49	70.11	338.49	6.8587	6.8654	6.8725	6	1781	7	352
11	69.93	338.65	70.13	338.65	6.8725	6.8797	6.8868	7	352	7	933
12	69.90	338.81	70.15	338.81	6.8868	6.8939	6.9006	7	933	7	1496
13	69.88	338.96	70.17	338.96	6.9006	6.9073	6.9144	7	1496	8	67
14	69.85	339.12	70.19	339.12	6.9144	6.9216	6.9283	8	67	8	630
15	69.82	339.27	70.21	339.27	6.9283	6.9349	6.9421	8	630	8	1192
16	69.80	339.43	70.23	339.43	6.9421	6.9492	6.9559	8	1192	8	1755
17	69.77	339.58	70.25	339.58	6.9559	6.9626	6.9697	8	1755	9	344
18	69.74	339.74	70.27	339.74	6.9697	6.9768	6.9835	9	344	9	936
19	69.72	339.89	70.29	339.89	6.9835	6.9902	6.9970	9	936	9	1512
20	69.69	340.04	69.83	340.04	6.9970	7.0037	7.0111	9	1512	10	128
21	69.66	340.20	69.85	340.20	7.0111	7.0185	7.0255	10	128	10	733
22	69.64	340.35	69.87	340.35	7.0255	7.0324	7.0394	10	733	10	1318
23	69.61	340.50	69.89	340.50	7.0394	7.0463	7.0532	10	1318	10	1903
24	69.58	340.65	69.91	340.65	7.0532	7.0602	7.0671	10	1903	11	490
25	69.56	340.80	69.94	340.80	7.0671	7.0741	7.0811	11	490	11	1069
26	69.53	340.95	69.70	340.95	7.0811	7.0881	7.0953	11	1069	11	1658
27	69.50	341.10	69.73	341.10	7.0953	7.1025	7.1102	11	1658	12	285
28	69.47	341.26	69.76	341.26	7.1102	7.1178	7.1250	12	285	12	907
29	69.45	341.41	69.79	341.41	7.1250	7.1322	7.1389	12	907	12	1489
30	69.42	341.55	69.82	341.55	7.1389	7.1456	7.1528	12	1489	13	81
31	69.39	341.70	69.85	341.70	7.1528	7.1600	7.1672	13	81	13	687
32	69.36	341.85	69.54	341.85	7.1672	7.1745	7.1819	13	687	13	1300
33	69.33	342.00	69.57	342.00	7.1819	7.1892	7.1966	13	1300	13	1916
34	69.30	342.15	69.60	342.15	7.1966	7.2039	7.2113	13	1916	14	545
35	69.28	342.30	69.63	342.30	7.2113	7.2186	7.2255	14	545	14	1144
36	69.25	342.44	69.65	342.44	7.2255	7.2324	7.2397	14	1144	14	1743

Track 10632

	GEOSAT FOOTPRINT		P-3 EQUIVALENT		P-3 FOOTPRINT TIMES			START		STOP	
-N-	LATITUDE	LONGITUDE	LATITUDE	LONGITUDE	START	MIDPOINT	STOP	IMAGE	LINE	IMAGE	LINE
1	71.06	347.80	71.18	347.80	5.8437	5.8512	5.8591	1	507	1	1268
2	71.08	347.62	71.20	347.62	5.8591	5.8671	5.8746	1	1268	1	2030
3	71.10	347.45	71.22	347.45	5.8746	5.8822	5.8901	1	2030	2	851
4	71.12	347.27	71.24	347.27	5.8901	5.8981	5.9056	2	851	2	1649
5	71.14	347.10	71.26	347.10	5.9056	5.9131	5.9198	2	1649	3	405
6	71.15	346.92	71.27	346.92	5.9198	5.9265	5.9328	3	405	3	1074
7	71.17	346.74	71.29	346.74	5.9328	5.9391	5.9450	3	1074	3	1705
8	71.19	346.57	71.31	346.57	5.9450	5.9510	5.9573	3	1705	4	363
9	71.20	346.39	71.32	346.39	5.9573	5.9636	5.9699	4	363	4	1018
10	71.22	346.21	71.34	346.21	5.9699	5.9762	5.9821	4	1018	4	1654
11	71.24	346.04	71.35	346.04	5.9821	5.9881	5.9944	4	1654	5	314
12	71.25	345.86	71.37	345.86	5.9944	6.0008	6.0078	5	314	5	1003
13	71.27	345.68	71.38	345.68	6.0078	6.0148	6.0218	5	1003	5	1725
14	71.29	345.50	71.38	345.50	6.0218	6.0288	6.0358	5	1725	6	471
15	71.30	345.32	71.39	345.32	6.0358	6.0428	6.0498	6	471	6	1194
16	71.32	345.14	71.40	345.14	6.0498	6.0568	6.0635	6	1194	6	1897
17	71.34	344.97	71.40	344.97	6.0635	6.0701	6.0771	6	1897	7	621

18	71.35	344.79	71.41	344.79	6.0771	6.0841	6.0909	7	621	7	1334
19	71.37	344.61	71.43	344.61	6.0909	6.0977	6.1045	7	1334	7	2038
20	71.38	344.43	71.46	344.43	6.1045	6.1114	6.1182	7	2038	8	765
21	71.40	344.25	71.48	344.25	6.1182	6.1250	6.1318	8	765	8	1468
22	71.41	344.07	71.50	344.07	6.1318	6.1386	6.1455	8	1468	9	195
23	71.43	343.89	71.53	343.89	6.1455	6.1523	6.1595	9	195	9	917
24	71.44	343.70	71.55	343.70	6.1595	6.1667	6.1740	9	917	9	1664
25	71.46	343.52	71.55	343.52	6.1740	6.1812	6.1885	9	1664	10	436
26	71.47	343.34	71.56	343.34	6.1885	6.1958	6.2031	10	436	10	1181
27	71.48	343.16	71.56	343.16	6.2031	6.2104	6.2176	10	1181	10	1927
28	71.50	342.98	71.56	342.98	6.2176	6.2249	6.2322	10	1927	11	700
29	71.51	342.80	71.57	342.80	6.2322	6.2395	6.2466	11	700	11	1445
30	71.52	342.61	71.54	342.61	6.2466	6.2538	6.2595	11	1445	12	119
31	71.54	342.43	71.56	342.43	6.2595	6.2652	6.2708	12	119	12	656
32	71.55	342.25	71.59	342.25	6.2708	6.2765	6.2825	12	656	12	1208
33	71.57	342.06	71.61	342.06	6.2825	6.2885	6.2942	12	1208	12	1760
34	71.58	341.88	71.64	341.88	6.2942	6.2999	6.3056	12	1760	13	305
35	71.59	341.70	71.66	341.70	6.3056	6.3112	6.3172	13	305	13	839
36	71.60	341.51	71.69	341.51	6.3172	6.3232	6.3290	13	839	13	1378
37	71.62	341.33	71.71	341.33	6.3290	6.3348	6.3416	13	1378	13	1955
38	71.63	341.14	71.72	341.14	6.3416	6.3484	6.3549	13	1955	14	576
39	71.64	340.96	71.73	340.96	6.3549	6.3614	6.3682	14	576	14	1181
40	71.65	340.77	71.74	340.77	6.3682	6.3750	6.3815	14	1181	14	1786
41	71.66	340.59	71.75	340.59	6.3815	6.3879	6.3948	14	1786	15	408
42	71.67	340.40	71.76	340.40	6.3948	6.4016	6.4080	15	408	15	1017
43	71.69	340.22	71.77	340.22	6.4080	6.4145	6.4213	15	1017	15	1625
44	71.70	340.03	71.78	340.03	6.4213	6.4282	6.4350	15	1625	16	266
45	71.71	339.84	71.79	339.84	6.4350	6.4418	6.4483	16	266	16	875
46	71.72	339.66	71.79	339.66	6.4483	6.4547	6.4616	16	875	16	1483
47	71.73	339.47	71.80	339.47	6.4616	6.4684	6.4749	16	1483	17	107
48	71.74	339.29	71.81	339.29	6.4749	6.4813	6.4881	17	107	17	712
49	71.75	339.10	71.82	339.10	6.4881	6.4950	6.5018	17	712	17	1331
50	71.76	338.91	71.83	338.91	6.5018	6.5085	6.5153	17	1331	17	1948
51	71.77	338.72	71.83	338.72	6.5153	6.5221	6.5288	17	1948	18	579
52	71.78	338.53	71.84	338.53	6.5288	6.5356	6.5420	18	579	18	1179
53	71.79	338.35	71.85	338.35	6.5420	6.5484	6.5552	18	1179	18	1778
54	71.80	338.16	71.86	338.16	6.5552	6.5620	6.5687	18	1778	19	412
55	71.81	337.97	71.87	337.97	6.5687	6.5755	6.5811	19	412	19	977
56	71.82	337.78	71.87	337.78	6.5811	6.5866	6.5905	19	977	19	1411
57	71.83	337.59	71.87	337.59	6.5905	6.5944	6.5981	19	1411	19	1759
58	71.84	337.41	71.87	337.41	6.5981	6.6019	6.6058	19	1759	20	123

Track 10633

	GEOSAT FOOTPRINT		P-3 EQUIVALENT		P-3 FOOTPRINT TIMES			START		STOP	
-N-	LATITUDE	LONGITUDE	LATITUDE	LONGITUDE	START	MIDPOINT	STOP	IMAGE	LINE	IMAGE	LINE
1	69.29	334.86	69.32	334.86	7.2888	7.2942	7.2993	2	1899	3	414
2	69.26	335.00	69.34	335.00	7.2993	7.3043	7.3098	3	414	3	914
3	69.23	335.15	69.37	335.15	7.3098	7.3152	7.3206	3	914	3	1431
4	69.20	335.30	69.40	335.30	7.3206	7.3261	7.3315	3	1431	3	1948
5	69.17	335.44	69.17	335.44	7.3315	7.3370	7.3437	3	1948	4	555
6	69.14	335.59	69.18	335.59	7.3437	7.3505	7.3573	4	555	4	1210
7	69.11	335.74	69.20	335.74	7.3573	7.3641	7.3705	4	1210	4	1843
8	69.08	335.88	69.22	335.88	7.3705	7.3768	7.3836	4	1843	5	482
9	69.06	336.03	69.24	336.03	7.3836	7.3904	7.3967	5	482	5	1099
10	69.03	336.17	69.26	336.17	7.3967	7.4031	7.4094	5	1099	5	1694
11	69.00	336.31	69.28	336.31	7.4094	7.4158	7.4225	5	1694	6	324
12	68.97	336.46	69.06	336.46	7.4225	7.4292	7.4355	6	324	6	933
13	68.94	336.60	69.09	336.60	7.4355	7.4418	7.4480	6	933	6	1521
14	68.91	336.74	69.11	336.74	7.4480	7.4543	7.4610	6	1521	7	146
15	68.88	336.89	69.13	336.89	7.4610	7.4677	7.4740	7	146	7	734
16	68.85	337.03	69.15	337.03	7.4740	7.4803	7.4866	7	734	7	1301
17	68.82	337.17	69.18	337.17	7.4866	7.4928	7.4994	7	1301	7	1880
18	68.79	337.31	68.90	337.31	7.4994	7.5059	7.5127	7	1880	8	494
19	68.75	337.45	68.94	337.45	7.5127	7.5196	7.5265	8	494	8	1108
20	68.72	337.59	68.97	337.59	7.5265	7.5333	7.5402	8	1108	8	1721
21	68.69	337.73	69.00	337.73	7.5402	7.5471	7.5539	8	1721	9	348
22	68.66	337.87	69.04	337.87	7.5539	7.5608	7.5676	9	348	9	962
23	68.63	338.01	69.07	338.01	7.5676	7.5745	7.5813	9	962	9	1574
24	68.60	338.15	68.70	338.15	7.5813	7.5882	7.5950	9	1574	10	196
25	68.57	338.29	68.73	338.29	7.5950	7.6017	7.6085	10	196	10	802
26	68.54	338.43	68.76	338.43	7.6085	7.6153	7.6221	10	802	10	1409
27	68.51	338.57	68.79	338.57	7.6221	7.6289	7.6352	10	1409	10	1993
28	68.47	338.70	68.82	338.70	7.6352	7.6415	7.6483	10	1993	11	592
29	68.44	338.84	68.84	338.84	7.6483	7.6550	7.6616	11	592	11	1189
30	68.41	338.98	68.52	338.98	7.6616	7.6682	7.6731	11	1189	11	1702
31	68.38	339.11	68.55	339.11	7.6731	7.6780	7.6833	11	1702	12	174
32	68.35	339.25	68.58	339.25	7.6833	7.6886	7.6939	12	174	12	647

Track 10634

-N-	GEOSAT FOOTPRINT LATITUDE	GEOSAT FOOTPRINT LONGITUDE	P-3 EQUIVALENT LATITUDE	P-3 EQUIVALENT LONGITUDE	P-3 FOOTPRINT TIMES START	P-3 FOOTPRINT TIMES MIDPOINT	P-3 FOOTPRINT TIMES STOP	START IMAGE	START LINE	STOP IMAGE	STOP LINE
1	65.75	323.17	65.92	323.17	9.2840	9.2910	9.2981	2	246	2	930
2	65.71	323.28	65.95	323.28	9.2981	9.3051	9.3122	2	930	2	1614
3	65.67	323.39	65.99	323.39	9.3122	9.3192	9.3263	2	1614	3	326
4	65.64	323.50	66.03	323.50	9.3263	9.3333	9.3394	3	326	3	982
5	65.60	323.61	65.71	323.61	9.3394	9.3456	9.3517	3	982	3	1591
6	65.56	323.72	65.75	323.72	9.3517	9.3578	9.3639	3	1591	4	226
7	65.52	323.83	65.79	323.83	9.3639	9.3700	9.3761	4	226	4	847
8	65.48	323.94	65.83	323.94	9.3761	9.3822	9.3883	4	847	4	1469
9	65.44	324.05	65.87	324.05	9.3883	9.3944	9.4000	4	1469	5	84
10	65.40	324.15	65.90	324.15	9.4000	9.4056	9.4117	5	84	5	675
11	65.36	324.26	65.40	324.26	9.4117	9.4179	9.4244	5	675	5	1316

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NORDA Technical Note 388	5. MONITORING ORGANIZATION REPORT NUMBER(S) NORDA Technical Note 388
6. NAME OF PERFORMING ORGANIZATION Naval Ocean Research and Development Activity	7a. NAME OF MONITORING ORGANIZATION Naval Ocean Research and Development Activity
6c. ADDRESS *(City, State, and ZIP Code)* Ocean Science Directorate Stennis Space Center, Mississippi 39529-5004	7b. ADDRESS *(City, State, and ZIP Code)* Ocean Science Directorate Stennis Space Center, Mississippi 39529-5004
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Ocean Research and Development Activity 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)* Stennis Space Center, MS 39529-5004	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. 63207N; PROJECT NO. 63207N; TASK NO. 100B 0101; WORK UNIT NO. 93327A
11. TITLE *(Include Security Classification)* KRMS GEOSAT - LIMEX '87 Data Products	
12. PERSONAL AUTHOR(S) D. T. Eppler and L. D. Farmer	
13a. TYPE OF REPORT Final	13b. TIME COVERED From _____ To _____
14. DATE OF REPORT *(Yr., Mo., Day)* July 1988	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)* passive microwave, KRMS, C programs, LIMEX, sea ice, digital tapes, GEOSAT, index

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Analog imagery acquired by the Navy K_a-band Radiometric Mapping System (KRMS) in support of the March 1987 GEOSAT and LIMEX experiments has been converted to digital form. We provide indices to these digital tapes, and listings of programs used to correlate KRMS navigation data with GEOSAT ephemeris data.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ■ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL D. T. Eppler	22b. TELEPHONE NUMBER *(Include Area Code)* (603) 646-4175	22c. OFFICE SYMBOL Code 332

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

*U.S. Government Printing Office: 1990-700-057-02074