

Fourth Quarterly Progress Report on NASA Contract NAS5-31347

A Planetary Version of PC-McIDAS

Sanjay S. Limaye
Principal Investigator

Space Science & Engineering Center
University of Wisconsin-Madison
1225 West Dayton Street
Madison, Wisconsin 53706
(608)262-9541

15 July 1992

1. Introduction

This is the fourth and final quarterly progress report on NASA Contract NAS5-31347. Since the administration of this contract has been transferred to the Center for Excellence in Data Information Systems (CESDIS) under a new contract (S.C. 550-80), this report represents the progress made during the first year of the three year program. Thus while this report constitutes the final report on Contract NAS5-31347, it describes the work in progress. Future reports will be filed under contract S.C. 550-80.

Work is continuing to meet the proposed objectives of creating a version of McIDAS for planetary applications. The primary objective is to develop a software capability under the commonly used operating systems such as UNIX and OS/2 to quantitatively analyze planetary image data from a variety of sources such as Voyager, Viking and Mariner spacecraft, radar imaging data obtained from Magellan as well as Hubble Space Telescope within the McIDAS environment (Suomi et al., 1983).

In order to utilize the resources of the vast software capabilities of McIDAS, any new data source or type requires that certain requirements be met so that the McIDAS environment can recognize the data attributes for navigation and radiometric calibration, and others deemed useful by the user. Thus the original "raw" data from a source such as CD-ROMs or Experiment Data Record (EDR) tapes and associated information needs to be imported into the McIDAS environment. The requirement for the ingesting software is to ensure that the needs of McIDAS for knowing the navigation and calibration be met. Once the ingesting program is able to do that, any new data type is fully manipulable using the existing McIDAS software. Thus the major software development is in the area of ingesting new data types and in enabling the use of supplementary information accompanying the respective data types.

In this last year we have concentrated on developing the planetary software under the version of UNIX operating system used on the IBM RS-6000 platform (AIX), although some applications have been developed under the mainframe (McIDAS-MVS) as well that will be eventually ported to McIDAS-AIX. The choice of the RS-6000 platform was based entirely on performance and cost effectiveness. In the recent past the performance of OS/2 hardware systems has also increased through the use of more advanced microprocessors (50 or 66 MHz 80486). It is likely that in the next year we will port the software to such platforms.

The core version of McIDAS-AIX within which the development of the planetary package is being carried out was released to outside users by the Space Science and Engineering Center (SSEC) of the University of Wisconsin-Madison, on April 15 1992. A copy of this software is essential for use of the planetary applications being developed under this contract and is licensed by SSEC.

Progress has been made on the software development. There are two basic areas of software development -basic system level enhancements to the data tracking, and applications software. The system level enhancements provide additional information necessary for processing the data. The applications are what a user might actually need to perform the desired analysis using the system. Additional applications programs will now be created or modified from existing ones as necessary so that they all use this common and more useful and capable system level format.

The structure of the image data descriptor block (described in the previous report) for system and applications program use has been finalized and implemented for the Voyager images and is being tested. In the near future additional data/spacecraft types such as Viking/Mars, Magellan/Venus and HST will be introduced.

Images from CD-ROM's produced by NASA's Planetary Data System (PDS) can now also be imported into the McIDAS environment which adhere to the new data structure. Figure 1 shows the on-line command help for the key-in to read the Voyager CD-ROMs. However, some difficulties have been encountered in accessing many of the disks because of an operating system incompatibility with the strict implementation of the ISO-9660 format for the CD-ROM's. This problem has been brought to the attention of the PDS office at JPL. The solution to this difficulty is a patch to the operating system which has been available for the Sun workstations and is being developed for the Silicon Graphics workstations. As yet a similar patch of the AIX operating system used on the RS-6000 workstations is not available, so the newer CD-ROM's distributed by the PDS are not directly accessible. For the moment we have imported the data from such disks directly by FTP from other machines (MS-DOS) for the purpose of testing the programs. The commands written for planetary applications follow the model of the core McIDAS. The 'HELP' offered on-line for the user for accessing the CD-ROM data is illustrated for the VCDROM command in Figure 1.

```

HELP VCDROM

VCDROM
VCDROM -- READS AND DECOMPRESSES CD-ROM FORMAT VOYAGER IMAGES   (RJK)
        AND WRITES THEM OUT IN McIDAS PLANETARY AREA FORMAT

EXAMPLES:
VCDROM FDS =43975.00 43985.00 TARGET =saturn FILTER =BLUE AREA =0
VCDROM FDS =43885.65 TARGET =mimas AREA =nnn
VCDROM keywords <"FILE NAME">
VCDROM <AREA> <"FILE NAME">

Remarks:
CDPATH   = Enter path in the string table using TE command. Should be blank if:

            (1) CD-ROM format image is in /mcidas/data,
            (2) if the path is included in the FILE NAME field, or
            (3) if the normally used FDS keyword option is chosen.

[A PATH IS REQUIRED, EVEN IF IT IS A BLANK FIELD]

FILE NAME = ANY VALID SOURCE FILE NAME, OPTIONAL, if FDS is given]

(Must begin with " and be the last entry on input line)
(a path may be included in FILE NAME if CDPATH

KEYWORDS:

FDS   = NNNNNN.NN NNNNNN.NN
(A single FDS or a range of numbers. A range specified will generate up to 50 areas consecutively
starting with AREA unless AREA =0)

TARGET = name/NAME (Must be specified when FDS numbers are used . TARGET is always
converted to lower case internally)

FILTER = GREEN, BLUE, ORANGE, CLEAR, VIOLET, UV, CH4_JS (5410A), CH4_U (6180A)
CAM    = WIDE or NARROW

AREA  = ANY VALID McIDAS AREA NUMBER (1-9999) [REQUIRED, def=0]. AREA =0 will allow
a CD-ROM search, with no reformatting. Searching through more than 10,000 FDS numbers on a
CD-ROM is time consuming. Choose FDS limits as narrow as possible to reduce waiting time. Use
the "?" command and the "/" pid" command to kill errant searches.)

```

Figure 1. HELP for using the command VCDROM to access Planetary Data System Voyager CD-ROMS.

New applications being developed include multispectral analysis and optical navigation of planetary images. Specific progress in the software development is described below.

Finally, the work performed during this year was described at the second annual Applied Information System Research Program Workshop held in Boulder, Colorado during August 1992.

2. System Level Software Development

The goal of the current effort is to enhance the McIDAS capabilities for handling planetary data. Simply stated, the requirement was minimal "core" software development and concentrated planetary software development. It was apparent early however that some key aspects of the core software needed to be addressed in terms of the ability to carry through seamlessly the additional data descriptors as well as the ability to intelligently display and animate three color composites. A prototype three color display approach has been tested using the McIDAS Wide Word Workstation (WWW), but implementation has been deferred until a suitable hardware capability has been acquired under this program.

Extended Data Descriptor Block

The original data directory employed by McIDAS has proved to be extremely useful in user interaction with the displayed frame. For planetary applications there are other aspects of an image that are also useful to have readily available. For example, in images of planets taken from spacecraft it is useful to know the location of any satellites or their shadows that may also appear in the image. In other instances the location of stars are also useful to know for refining the camera pointing. The original directory did not have room for storing such additional information, therefore an extended Data Description Block that describes a dataset (or "area" in McIDAS parlance) has been developed. The structure is flexible enough to describe different types of information that are likely to be spacecraft/mission as well as target and instrument dependent. For example, Magellan and Voyager data products are derived from two entirely different sensors, synthetic aperture radar and vidicons. Thus, a descriptor block structure for Voyager data can be expected to be different from Magellan, and an identifier in the descriptor block is used to identify the different data types.

One of the key enhancements is in the way the planetary constants are handled. In the past each target body had a single set of polar and equatorial radii assigned. While this sufficed in the exploratory phase, it became soon apparent that more flexibility was needed for certain targets. A case in point is Venus, for which the cloud top radius is some 65 km different from the surface radius. For Titan the difference is nearly five times greater with the added complication that the cloud level radius is color dependent. Since the planetary radii are required in the image navigation process, the software needs to know which reference radius to use without ambiguity. The new structure allows the user to not only specify the reference type, but also to know what value is being used.

The Data Description Block also has room to record whether any associated files have been created and if so, their types and names. The idea being that as the processed data is migrated from one platform to another or to another medium, the associated files are carried along. Such data files might be histograms of the entire data or specific regions, brightness area statistics of certain outlines etc.

The Data Description Block structure is described in greater detail in Appendix I. Appendix II illustrates the use of McIDAS-X to access and process Voyager images from the Planetary Data System CD-ROM's.

Processing History

A utility for entering not only the user specified but also the default required inputs of an application program into the image processing history has been created. This will allow a complete audit trail of the processes that an image data area has been subjected to by adding the program outputs to the audit trail in a similar fashion.

In keeping with the McIDAS tradition of maintaining modular software, the planetary applications are being maintained as individual modules that perform specific tasks which can be daisy chained by a user specified procedure. This feature of can be utilized to pipeline several applications processes with greater ease so that the output of one program is the input of another, allowing considerable streamlining of the analysis process.

Figure 2 shows an example of an audit trail of a Voyager image showing the processing history.

```

LA 6011 FORM=AUDIT
area ss yyddd hhmmss lcor ecor lr er zr lsiz esiz z  bands
-----
6011 47 80309 200434 1 1 1 1 1 1000 1000 11.....
proj: 0 created: 92310 174044 memo: ---CDROM DECOMPRESS---
type:VISR cal type:BRIT
area offsets: data = 9040 navigation = 256 calibration = 2816
doc length: 0 cal length: 0 lev length: 0 PREFIX = 0
valcod: 0 zcor: 1 band-8: NA reel#:*****
-Audit Trail
yyddd hhmmss
-----
PICNO = 1528S1-009 TARGET = SATURN
FDS = 34699.24 CAMERA = NARROW ANGLE SPACECRAFT = VOYAGER
SHUTTER TIME = 80309 200434 BOTSIM
FILTER = BLUE (2) EXP = 1.4400 SEC. GAIN = LOW
LINES = 800 ELEMENTS = 800 SCAN = 3:1 EDIT =

VCDROM: Internal Parameters and Defaults:
SOURCE = /cdrom/saturn/c3469xxx/c3469924.imq
FDS-START = 3469900
FDS-END = 3469950
TARGET = saturn
CAMERA =
FILTER =
SEARCH INDEX = 0
OUTPUT AREA = 6001
92310 173839 VCDROM FDS = 34699.00 34699.50 AREA = 6001 TARGET = SATURN
92310 174033 SEDRIN 6001 X VGR1S
92310 174038 RF 6001
92310 174058 GEOM 6001 6011
92310 174205 PLAEDG 1 X 3
92310 174212 PLAEDG 2 X 3
92310 174218 PLAEDG 3 X 3
92310 174631 PLAEDG 29 X 3
92310 174634 CALCMA
LA: Done

```

Figure 2. Audit trail for an image read from a CD-ROM.

3. Applications Software Development

Development of four major applications has been underway which have different objectives and are new to McIDAS. The first one is a generalized perspective visualization of planetary data, the second is a method for refining planetary image navigation by referencing the stars, the third application is multicolor animation and display, and the fourth one is for multispectral analysis of data. These are described below.

Generalized Planetary Visualization

Frequently it is necessary to show the planetary data in a different form than the one in which it is acquired. An extreme example is the Magellan synthetic aperture data which is obtained in narrow strips that extend from pole to pole, or Pioneer Orbiter images of Venus that are obtained by the orbital translation of the spacecraft scans of the planet creating a very distorted view of the planet. To allow the mapping of such and other data, a generalized mapping utility has been developed that allows the user to define the viewing geometry in terms of range (in planetary radii), azimuth angle, sub-viewer latitude and longitude, tilt and angle of view with respect to the nadir direction. Any data, either an image or a model output that has been navigated then can be mapped into this view readily. This capability can be utilized for mission planning purposes to show the animated view as seen by spacecraft instruments at various times in orbit. Thus the Venus images obtained from Pioneer can be viewed as a globe, the Magellan Mosaiced Image Data Records (MIDR's) can be pasted to form a perspective view, or, the NOAA-AVHRR imagery can be used to provide a perspective view of the earth from a low orbit to mimic the view obtained by the space shuttle astronauts. Some examples of such views are shown in Figure 3.

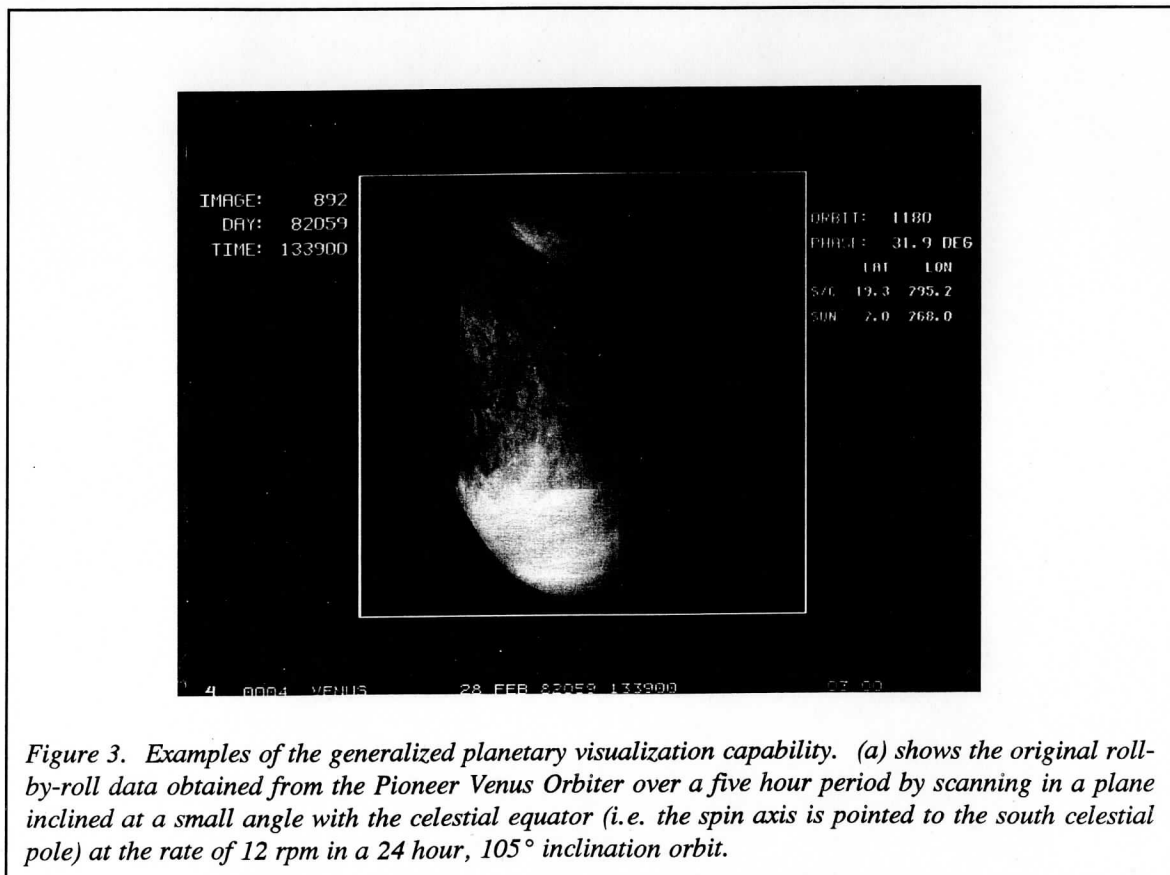


Figure 3. Examples of the generalized planetary visualization capability. (a) shows the original roll-by-roll data obtained from the Pioneer Venus Orbiter over a five hour period by scanning in a plane inclined at a small angle with the celestial equator (i.e. the spin axis is pointed to the south celestial pole) at the rate of 12 rpm in a 24 hour, 105° inclination orbit.

Figure 3. (b) shows the same data but mapped at a range of $6R_V$ with the sub spacecraft point in the southern hemisphere. In the process of mapping the shading due to illumination has been removed by modelling the scattered intensities by the Minnaert law.

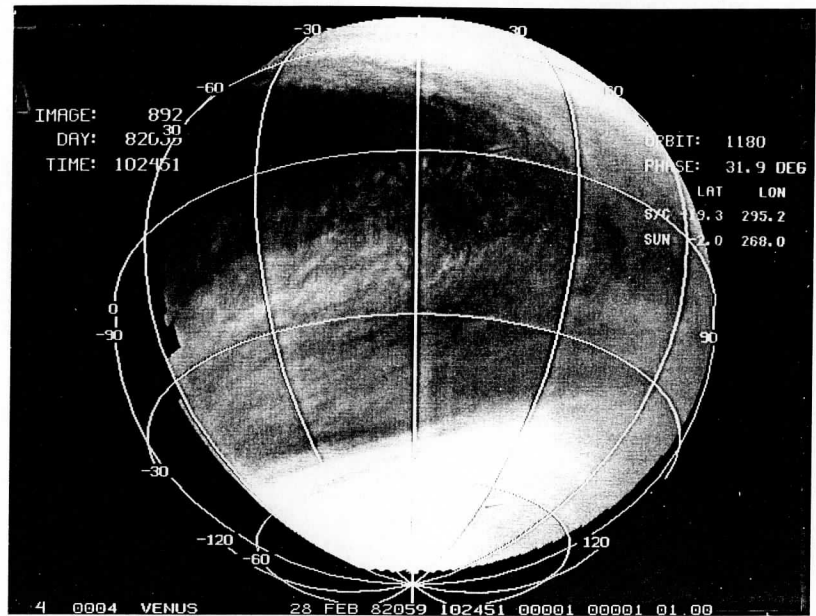
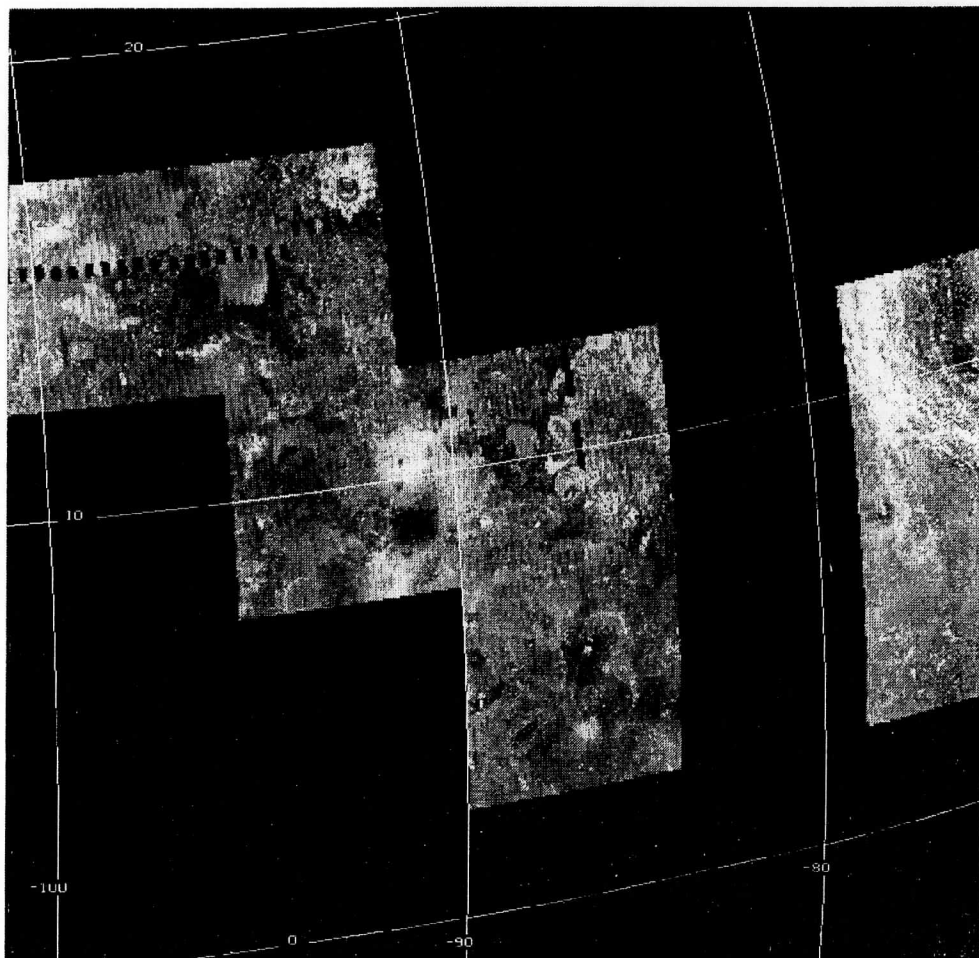


Figure 3. (c) shows the pasting of a few Magellan MIDR's read from CD-ROMs in sinusoidal projection and mapped into a perspective view.



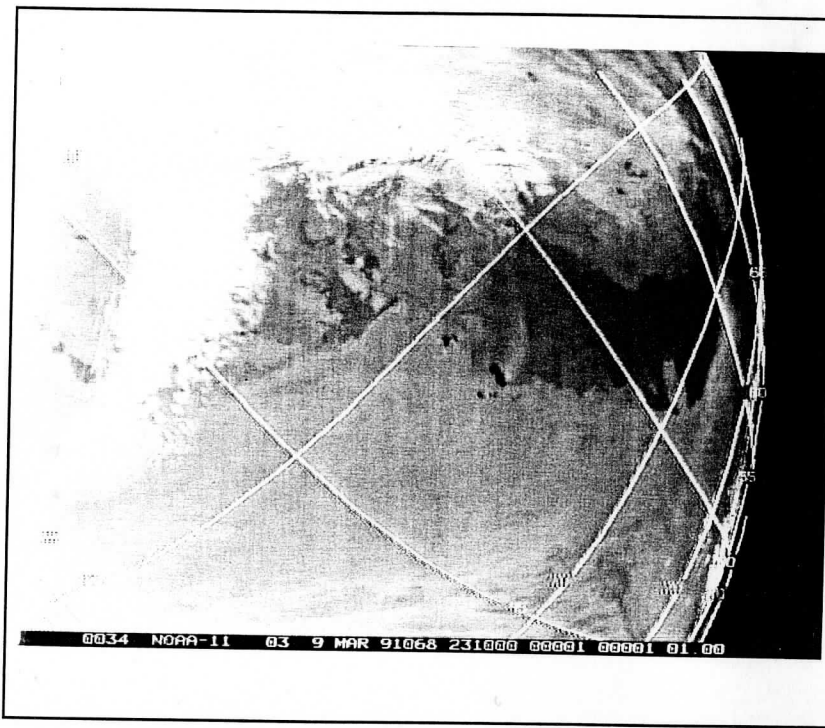


Figure 3. (d) shows a view of the earth as seen from a 180 nm altitude and looking at an angle of 45° to the nadir. The data source is the NOAA 11 polar orbiter and acquired from AVHRR in the 3.4μ channel (Band 3) and shows the oil wells on fire in Kuwait (dark regions).

Optical Navigation

For framing cameras such as those used on Voyager and Galileo missions, the pointing information as obtained from the engineering data is not good enough for precise measurements in the reference frame of the target. Traditionally, therefore the pointing is improved by determining the center of the target image to obtain the correction required to the instrument-to-target reference frame transform. However, there is an additional offset, of an apparent rotation about the optic axis, which is limited by the spacecraft's attitude control system. Although the limit is supposed to be $\pm 0.1^\circ$ for Voyager 1 and 2, in practice the offsets are significantly larger. For the end user the only way to correct for such a roll error is to reference the pointing to reference stars in the same frame, or a simultaneous wide angle frame if available. To address this need programs to precisely determine star locations and to subsequently refine the pointing ('C' matrix) by utilizing the offsets in the three axes have been developed and tested with Voyager Neptune images.

Finding stars in short exposure images taken with vidicons or CCD's is made difficult by the presence of the dark sky background and noise. In longer exposure frames the stars show up as trails due to the movement of the cameras. In any case, the stars rarely show up as a single bright pixel due to the optical transfer function of the optical system which causes "bleeding" and motion of the camera which produces smear. The techniques developed make use of the sky background, a certain brightness threshold and determine the brightness weighted center of the star and an estimated size. Knowledge of the star's position allows the correction to the Voyager camera pointing angle from its image location by correcting for line and element offset and roll angle. Figure 4 shows a summary of the roll angle correction in degrees as determined from measurements of star locations on several Voyager 2 images of Neptune.

3-Channel Display and Animation

Additional work in this area has been deferred for the RS-6000 system until proper graphics card and required support under the vendor's X-windows libraries is available. Given vendor pricing information that has been recently received, it is probable that a different platform is more economical. In the meantime, pseudo three color displays are possible using the multispectral classification approach that has been developed and described below.

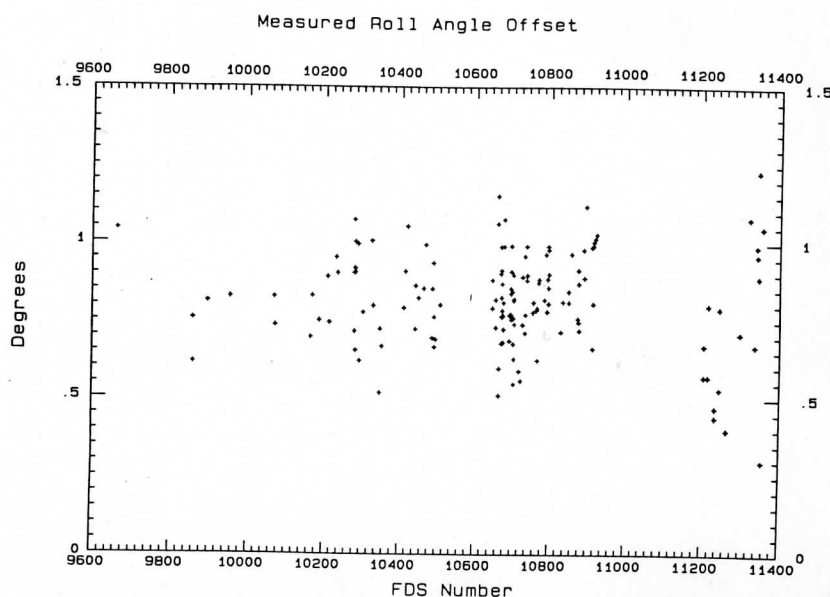


Figure 4. Difference between the measured north angle and the SEDR specified north-angle as obtained by using reference stars.

Multispectral Classification

Two classification programs have been created to run on McIDAS on the IBM RS-6000 computer. The first program performs an unsupervised classification of data from a multispectral image using the "Iterative Self Organizing Data Analysis Technique" or ISO-DATA. (Tou and Gonzalez, 1974). This method begins with an initial set of arbitrary cluster means and classifies the image repetitively, using a minimum spectral distance to the mean to assign classes to each pixel. The arbitrary means of classes will then shift to the means of the clusters in the data. The advantages of this algorithm are that it is not spatially biased to any edge of the data file, and that it produces better results than for other methods such as sequential clustering for data that are not normally distributed. The disadvantages involve the narrow range of input parameters which lead to convergence, and simplistic criteria for cluster splitting and merging.

These disadvantages are addressed by the second classification program, which uses an improved splitting criterion, and Transformed Divergence as a merging criteria to improve accuracy and ease of use. In order to support of this program, a utility program has been written to combine single band McIDAS data areas (for example Voyager images which are acquired one at a time through a single color filter) into a single multi-band area. This allows classification of a region viewed by different satellites at different dates (mapped into a common coordinate system).

In order to compare the results of the multi-spectral program with a widely used commercial product another utility program to convert ERDAS (Environmental Remote sensing

Data Analysis System software package) format files into McIDAS area files and vice versa has been written. This program now converts multi-band images and color look-up table files.

The program is being used to compare classification outputs with classifications produced with the ERDAS package. To test the Isodata program, a Wisconsin Department of Natural Resources project image with ground truth has been classified and compared with the output of the classifier with the ERDAS Isodata routine, (which is not a full implementation as given by Tou and Gonzales). The McIDAS program was shown to produce results at least as good as results from the ERDAS routine when compared with the ground truth. Additional testing of the program with other data sources such as the terrestrial AVHRR and multi-spectral combinations of Voyager planetary images is continuing.

Figure 5 shows one example of a classification of Voyager Jupiter data. The multispectral images were navigated, brightness normalized and mapped before classification.

This work forms part of the Master's degree requirements for Mr. G. Peltzer in the Department of Civil Engineering, University of Wisconsin-Madison.

Figure 5. An example of multispectral classification using four bands. Shown are distinct spectral classes in a region of Jupiter as derived from Voyager images acquired in ultraviolet, blue, orange and green filters.



References

Suomi, V.E., R.J. Fox, S.S. Limaye, and W.L. Smith, 1983: McIDAS III: A Modern Interactive Data Access and Analysis System. *J. Clim. Appl. Meteor.*, **22**, 766-778.

Tou, J. and Gonzalez, R.C. Pattern Recognition Principles, Addison Wesley Publishing Co., 1974.

APPENDIX I

Planetary Image Data Area and Directory Structure under McIDAS-X

This appendix describes access to the new area structure and navigation data for planetary imagery within the McIDAS-X environment. McIDAS-MVS mainframe areas can be transferred to any McIDAS-X or McIDAS-OS/2 workstation using SENAA, but when received by the workstation, those areas have inadequate space for SPICE navigation and radiometric calibration. We have defined a new McIDAS-X area structure, compatible with all versions of McIDAS. This area structure has sufficient room for SPICE navigation and radiometric calibration. A new McIDAS-X keyin, PAA, accepts either EDR or VICAR source data in mainframe McIDAS area formats and converts them to the new planetary McIDAS-X format.

There are several options within McIDAS-X for reformatting imagery from other sources. One option is the program VCDROM, which is used to input Voyager imagery from CD-ROM. A similar McIDAS-X keyin for reformatting Space Telescope images is also planned. Images could be input directly to McIDAS-X from tapes, such as EDRs and VICAR tapes. Such images would have to be reformatted using Unix or OS/2 versions (which as of yet do not exist) of the mainframe background programs PBNEWEDR and PBVICAR2. Those programs may be written if a need arises to read tapes at the workstation level, but the trend is toward acquiring data either directly via TCP/IP or from CD-ROMs.

The new planetary area format is documented below. The format is fully backward compatible with all existing McIDAS workstation software, so the areas may be freely moved from one McIDAS workstation environment to another (from Silicon Graphics, to Sun, to PC, to RS6000). Older planetary applications will require minimal modifications -- notably relinking with a new version of subroutine ACCESS, which contains one additional argument. This additional argument specifies a specific type of navigation data and guarantees that the call will error out if the requested data isn't present in the area. The user will not be able to acquire valid navigation data by accident without knowing specifically where it comes from.

New planetary applications wishing to use SPICE kernel information will have to change the value of one of the old calling arguments. Calls to ACCESS for a new area format have three new parameter values: NFLAG=7,8,9. Those NFLAG values will become the new standards for SEDR- and SPICE-based navigation. Subroutine ACCESS has an additional new option, NFLAG=6, so software can still be made to work with the old format areas for use in debugging. Other values of NFLAG (formerly 1-5) should produce no visible changes in old applications used with new format areas, and will continue to function as before, so the new ACCESS will be fully backward compatible with the old version (provided new areas are used and the PLAN block is specified). Only one version of ACCESS will be maintained in the future for all McIDAS workstation environments.

The mainframe version of Planetary McIDAS kept navigation data in files called "codicils", separate from the image areas. On McIDAS-X and McIDAS OS/2, each planetary area is stored as a single binary disk file containing all the information necessary to display and navigate the image. Each file has the same format, although file lengths are variable with image size and amount of audit trail.

The first 256 bytes of every file contains the area directory block for the image. Each value in the block is a 4-byte word (64 words). This directory contains information about both the area and the image it was created from. It is described in detail later.

The next entry contains the 'NAV ' block for the image. Again, each value in this block is a 4-byte quantity holding the navigation parameters necessary to locate the projection of the image pixels on a planet surface. If no navigation is available for the image the block will be zero filled, or contain the McIDAS standard hex byte string "80808080", meaning "no data". Word 35 of the area directory contains the byte offset of the NAV block within the file. For planetary navigation the new navigation block length is always $5 \times 128 = 640$ words or 2560 bytes.

The next block of bytes contains a 'CAL' entry. This entry will be filled with data if the image is radiometrically calibrated or (in the case of Mariner or Voyager imagery) reseau need to be located in the image to geometrically calibrate it. Like the area directory and NAV blocks, each value in a CAL block is a 4-byte quantity. Word 63 of the area directory is a byte offset to the CAL block's position within the file. For planetary image data the calibration entry consists of three parts, totalling 512+4096+1616=6224 bytes:

1. Part one is a 128 word entry (512 bytes) used to parameterize a function to display the image. Two byte data on the mainframe, for example, uses a three parameter linear transfer function (min, max, scale factor) to convert a range of 16-bit DN values to 63 levels on the display screen. The keyin PDF (planetary display frame) sets up such a transfer function for displaying 2-byte planetary images in McIDAS.

2. Part two consists of 1024 4-byte words (4096 bytes), suitable for up to 10 bits/pixel resolution in a lookup table of 4-byte floating point or scaled integer radiance values. This transfer function is appropriate for doing calculations with physical quantities instead of DN values.

3. Part three is space for a set of 202 REAL*4 line-element reseau locations (1616 bytes). Mariner and Voyager class spacecraft used vidicon sensors with electron beam readout subject to variable distortion from electromagnetic fields the spacecraft passed through. The image geometry was preserved by etching reseau marks on the glass vidicon faceplate so one could reconstruct the exact geometry the telescope optics projected into the focal plane at which the vidicon resided. These reseau locations are found by keyin RF (planetary reseau finder).

The next entry contains the actual digital data values for the image arranged line by line, with no record separators. The length of the 'DATA' block is computed using values within the area directory entry. Word 34 of the area directory is the byte offset to the start of the DATA block within the file. Image data stored in a McIDAS digital area is organized in a two-dimensional array of lines and elements. Each pixel in an area has a line number, starting with zero at the top, and an element number starting with zero on the left side of the line. This line-element number pair defines a coordinate system, called the 'area coordinates', for the elements in the area. Note that the pixel count for areas (not for images or frames) starts at zero, not at one. The line and element counting for images and frames is arbitrary and can be specified in words W6 and W7 of the area directory. All McIDAS applications software must use this convention.

Finally, we have a trailing entry which contains ASCII data that more fully identifies the data source and stores time stamped comments and notes from the user and applications programs. This provides an audit trail of arbitrary length, giving a full history of processing. Because it is at the end of the area file, this entry may grow without limit as the image gets processed.

The first six lines of the audit trail are called the "image identification block", and have the same format for every planetary image, independent of the source. Some Voyager images acquired from EDR's or VICAR tapes have erroneous data in this block, but the information acquired from CD-ROM sourced images (via keyin VCDROM) is correct. All planetary keyins use the image identification block data to identify default keyin values for any displayed image.

```

0 . . . -----
      |  DIRECTORY      |
256 . . . -----
      |  NAVIGATION    |
2816 . . . -----
      |  CALIBRATION   |
9040 . . . -----
      |  IMAGE         |
ENDIMG . . . -----
      |  AUDIT TRAIL   |
EOF . . . -----

```

AREA DIRECTORY

Each area that is extracted from a spacecraft image has associated with it a record called the 'area directory'. The data in the directory are stored as 32-bit (4-byte) two's complement binary integers or as ASCII character data. The length of the directory is 256 bytes (64 words). A list of the directory contents follows:

WORD	NAME	DESCRIPTION
-----	-----	-----
1		Contains zeros if record is valid.
2	Area Format	Always 4 (as of June 1985).
3	SSS	Spacecraft Identification Number.
4	YYDDD	Nominal Year and Julian day of area.
5	HHMMSS	Nominal Time of image.
6	UPPERLEFTLINE	Image line coordinate of area line 0, elem. 0.
7	UPPERLEFTTELE	Image element coordinate of area line 0, elem.0.
8		Not used.
9	NLINES	Number of lines in this digital area.
10	NELES	Number of elements in each line.
11	ELESIZ	Number of bytes/element (1, 2 or 4).
12	LINERES	Line Resolution; spacing in image-lines between consecutive area lines.
13	ELERES	Element Resolution; spacing in image-elements between consecutive area elements.
14	NCHANS	Maximum number of bands/line of area.
15	PRESIZ	Length of line prefix in bytes. Indicated by the sum of W49, W50, W51 (+ 4 if validity code is present, see W36).
16	PROJ	McIDAS user project number under which the area was created.
17	CREATION DATE	Area creation day in YYDDD format.
18	CREATION TIME	Area creation time in HHMMSS format.
19	FILTER MAP	(for Multi-Channel Images) a 32 bit vector. If a bit=1 there are data for that band in the area. The rightmost bit is for band 1.
20-24		Internal use only.
25-32	Memo	(32 ASCII characters available for comments).
33		Number of area containing this directory.
34		Byte offset to the start of the image data within the area file.
35		Byte offset to the start of the navigation block within the area file
36		Validity Code: If these bytes are non-zero they represent a code which must match the first 4 bytes of the line prefix. If the code and prefix bytes are not equal the line does not contain valid data and must be ignored.
37-44	PDL	Exists if the image was made in GOES mode AA or AAA packed byte format.
45		If image is a Mode AA DS ('STEP', 'DWEL') these bytes indicate the origin of Band 8. Not used for GOES mode AAA.
46		Actual image start YYDDD.
47		Actual image start HHMMSS.
48		Actual starting scan.
49		Line prefix documentation section length in bytes.
50		Line prefix calibration section length in bytes.
51		Line prefix level map section length in bytes.
52		Calibration type: 'VISR', 'VAS ', 'ERBE', 'AVHR', ...
53		Calibration Type: Physical units in which the digital data are stored. The calibration type determines how the image is displayed on the screen or processed by various applications. e.g.:

	'BRIT'	8-bit DN values,
	'CAL '	16-bit DN values
	'TEMP'	Degrees Kelvin,
	'RAD '	Watts/cm**2/ster (scaled integer).
54-62		Internal use only.
63		Byte offset to the start of the calibration block within the area file
64		Number of lines in the audit trail.

DATA SOURCE

McIDAS uses a three digit code field to identify the data source (usually a spacecraft) in the area directory:

SSS Code	Sensor Source
-----	-----
0	Non-Image Derived Data
2	Graphics
3	MDR Radar
4	Meteosat Visible
5	Meteosat Infrared
6	Meteosat Water Vapor
7	RADAR
8	Miscellaneous Aircraft Data (MAMS)
12	GMS Visible
13	GMS Infrared
14	ATS 6 Visible
15	ATS 6 Infrared
16	SMS-1 Visible
17	SMS-1 Infrared
18	SMS-2 Visible
19	SMS-2 Infrared
20	GOES-1 Visible
21	GOES-1 Infrared
22	GOES-2 Visible
23	GOES-2 Infrared
24	GOES-3 Visible
25	GOES-3 Infrared
26	GOES-4 Visible (VAS)
27	GOES-4 Infrared & Water Vapor (VAS)
28	GOES-5 Visible
29	GOES-5 Infrared & Water Vapor (VAS)
30	GOES-6 Visible
31	GOES-6 Infrared
32	GOES-7 Visible
33	GOES-7 Infrared
36-40	NOAA Series Satellites
41	TIROS-N
42	NOAA-6
43	NOAA-7
44	NOAA-8
45	NIMBUS Satellites
46	Mariner 10

47	Voyager 1
48	Voyager 2
49	Magellan
50	Hubble Space Telescope
51-55	Digitized Radar
80	ERBE
95	NOAA-9

A similar code is used by the DSN and SPICELIB for identifying the planetary spacecraft. Naturally the codes are different. Negative codes are used by NAIF and the SPICE routines for spacecraft and target bodies. The code assigned to a particular spacecraft is the negative of the code assigned to the same spacecraft by JPL's Deep Space Network (DSN). Integer codes have been assigned for the following spacecraft:

-12	Pioneer 12 (Venus orbiter)
-18	Magellan
-27	Viking 1 orbiter
-30	Viking 2 orbiter
-31	Voyager 1
-32	Voyager 2
-77	Galileo orbiter
-94	Mars observer

Codes have also been assigned for the following barycenters:

0	Solar system barycenter
1	Mercury "
2	Venus "
3	Earth "
4	Mars "
5	Jupiter "
6	Saturn "
7	Uranus "
8	Neptune "
9	Pluto "
10	Sun "

The code for a satellite is normally computed by adding its IAU designation to 100 times the code for its barycenter.

301	Moon
401	Phobos
402	Deimos
501	Io
502	Europa
503	Ganymede
504	Callisto
505	Amalthea
506	Himalia
507	Elara

- 508 Pasiphae
- 509 Sinope
- 510 Lysithea
- 511 Carme
- 512 Ananke
- 513 Leda
- 514 Thebe (1979J2)
- 515 Adrastea (1979J1)
- 516 Metis (1979J3)

- 601 Mimas
- 602 Enceladus
- 603 Tethys
- 604 Dione
- 605 Rhea
- 606 Titan
- 607 Hyperion
- 608 Iapetus
- 609 Phoebe
- 610 Janus (1980S1)
- 611 Epimetheus (1980S3)
- 612 Helene (1980S6)
- 613 Telesto (1980S13)
- 614 Calypso (1980S25)
- 615 Atlas (1980S28)
- 616 Prometheus (1980S27)
- 617 Pandora (1980S26)

- 701 Ariel
- 702 Umbriel
- 703 Titania
- 704 Oberon
- 705 Miranda
- 706 Puck (1985U1)
- 707 Portia (1986U1)
- 708 Rosalind (1986U2)
- 709 Juliet (1986U3)
- 710 Cressida (1986U4)
- 711 Belinda (1986U5)
- 712 Desdemona (1986U6)
- 713 Cordelia (1986U7)
- 714 Ophelia (1986U8)
- 715 Bianca (1986U9)

- 801 Triton
- 802 Nereid

- 901 Charon (1978P1)

A planet is always considered to be the 99th satellite of its own barycenter.

- 199 Mercury
- 299 Venus

- 399 Earth
- 499 Mars
- 599 Jupiter
- 699 Saturn
- 799 Uranus
- 899 Neptune
- 999 Pluto

Spacecraft images occur in several different forms, depending on the original data source. They can be from framing cameras, such as vidicons or solid state detector arrays, or from spin scan cameras such as Pioneer. We describe a raw image as a sequence of 'lines' arranged each below the previous one and numbered from top to bottom, the top line being number 1. Each line consists of a sequence of 'elements' arranged across the line and numbered left to right, the left-most element being number 1.

An element may be an 8,16 or 32 bit quantity, depending on the image source and format. Most raw images are 8 bits, while photometrically corrected images are 16 bits. Raw 10 or 12 bit imagery must be stored in McIDAS areas as two bytes per pixel. Radiometrically corrected images or composite multispectral images may have other formats, but are limited in McIDAS to 4 bytes per pixel. More than 4 bytes requires using the spectral band structure alluded to in the description of the area directory.

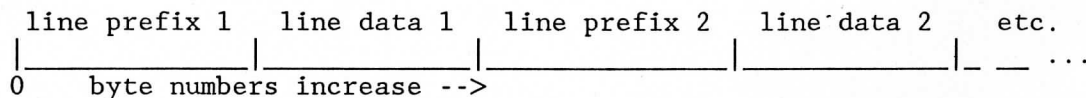
This line/element numbering scheme determines a pair of coordinates for each element, called the 'image coordinates' of the element. This coordinate system is defined only by the spacecraft/camera combination and is independent of how the data are stored. If all of the elements of a raw image were contained in an area, there would be no point in distinguishing between 'image' and 'area' coordinates. What is stored in an area is a rectangular subset or superset of an image, obtained by sampling, averaging, repeating, or editing of lines and elements, or some other pixel mapping process.

In order to map an area to the original image the following formulas are used:

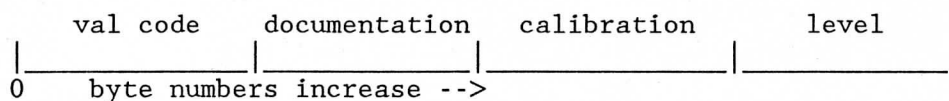
$$\begin{aligned} \text{Image_Line} &= \text{UpperLeftLine} + (\text{Area_Line} * \text{LineRes}) \\ \text{Image_Element} &= \text{UpperLeftEle} + (\text{Area_Element} * \text{EleRes}) \end{aligned}$$

UpperLeftLine is the image line coordinate of the first area line. UpperLeftEle is the image element coordinate of the first area element. When LineRes = EleRes = 1, the area is said to be at 'Resolution 1', or 'Full Resolution'. When, for example, LineRes = EleRes = 4, only every fourth line and element of an image originally at resolution 1 are included in the area. This area is said to be at 'Resolution 4'. Each line in an area has the same total length. This length, in bytes, is always a multiple of four.

The image data in an area may be viewed as a continuous stream of bytes numbered from 0. Within this stream of bytes, the area data are contained line-by-line, with the lines in order, first to last. Each line is further divided into two parts, the 'Line Prefix' and the actual line data (image elements).



The line prefix contains documentation about the image and the particular line:



The size and content of the line prefix depend heavily upon the area type, which in turn is determined by the data source. The area type is given in word W52 of the area directory. Planetary McIDAS areas usually contain the area type "VISR". Regardless of the area type, each line in the area has the same length prefix. This length, in bytes, is given in word W15 of the area directory. It may be zero, and if so, there is no line prefix defined for the area. Images will be displayed by the DF keyin from the first byte of image data in each line in the area.

NAVIGATION

To navigate an image is to associate planet-based coordinates, usually planetocentric latitude and longitude, with the pixels of the image. This is done using highly complex mathematical models of the spacecraft and camera embedded in parameterized navigation software modules.

Navigation parameters for a digital area, when present in the McIDAS system, will be supplied in the 'NAV' block of the area file. Navigation information is used with the McIDAS navigation software to convert image coordinates (line,element) to planet-centered coordinates (latitude, longitude).

The first 128 words in the planetary McIDAS area navigation block are designated as the "current" navigation for the area, for use with any McIDAS compatible navigation software modules, such as:

NVXPLA	Planetary Navigation Framing Cameras
NVXPS	Polar Stereographic Projection
NVXMER	Mercator Projection
NVXLAM	Lambert Conformal Projection
NVXTIR	Polar Orbiter Nadir Scanners
NVXGOE	Geostationary Spin Scanners
NVXREC	Rectilinear latitude-longitude map
NVXSIN	Sinusoidal equal area projection

Images suitable for navigation should be geometrically rectified ("geomed") or be already remapped into one of the standard McIDAS map projections. The first 128 word block must be loaded by a call to the planetary subroutine ACCESS (or NVXINI) prior to doing anything with the geomed image using the NVXPLA module. ACCESS will refuse to load the current navigation block if the block already contains another McIDAS type navigation, such as one of the map projections mentioned above.

Raw images cannot normally be navigated. Exceptions consist of navigation of a raw image by virtual geometric rectification using keyin GEOMTR, or navigation of a raw solid state sensor image from Space Telescope. Such special cases should not use the first 128 word block, but the last 128 word block instead.

The next 384 words are reserved for parameters calculated with the SPICELIB routines. The words fall into five groups: 1) Spacecraft & Instrument Specific (128-191), 2) Central & Picture Body Specific (192-255), 3) Imaging Geometry Specific (256-351), 4) Computed Quantities (384-431), 5) Supplementary Information (432-511). Each group is detailed below using a zero-based word count.

Spacecraft/Instrument Specific Block

WORD -----	ITEM (S/C & Instrument) -----	TYPE ---	COMMENTS -----
128-129	SPICE Block Type/Version	C*8	'SPICEVGR' 'SEDRVGR'
130-131	SPICENAV Type (S/C Data or Mapped)	C*8	Mission Simulation
132	Spacecraft ID (NAIF Code)	I*4	
133	Instrument ID (NAIF Code)	I*4	
134-135	Instrument ID	C*8	ISSNA etc.
136-137	Picture ID-1 (FDS)	C*8	S/C/ Data ID
138-140	Picture ID-2 (PICNO)	C*12	
141	Picture Time (Year)	I*4	
142	Picture Time (Day)	I*4	
143	Month	I*4	
144-145	Picture Time (hh:mm:ss)	C*8	
146	Frame Size (in scan direction)	I*4	Constant
147	Frame Size (in cross-scan direction)	I*4	Constant
148-149	FOV (Degrees in scan direction)	R*8	Constant
150-151	FOV (Degrees in cross-scan direction)	R*8	Constant
152	Focal Length (mm)	R*4	Constant
153	Exposure Time (sec)	R*4	
154-155	Filter ID (name)	C*8	
156	Filter ID (ordinal)	I*4	
157	Filter ID (central wavelength nm)	R*4	
158	HWHM	R*4	
159-160	Units of Above (nm, microns)	C*8	
161-162	Imaging Sequence Mnemonic	C*8	VPHOT etc.
163-164	Imaging Mode Mnemonic	C*8	BOTSIM etc.
165-166	GEOMed or not	C*8	RAW, GEOM, DECONVOL
167-191	Spare		
WORD -----	ITEM (Central & Picture Body) -----	TYPE ---	COMMENTS -----
192-194	Central Body Name	C*12	
195	Central Body (NAIF Code)	I*4	
196-198	Picture Body	C*12	
199	Picture Body (NAIF Code)	I*4	
200	Picture Body Rotational Period (hrs.)	R*4	
201	Picture Body Pole Right Ascension	R*4	
202	Picture Body Pole Declination	R*4	
203	Nominal Picture Body Eq. Radius (km)	R*4	
204	Nominal Picture Body Pol. Radius (km)	R*4	
205	Color Dep. Picture Body Eq. Radius (km)	R*4	
206	Color Dep. Picture Body Pol. Radius (km)	R*4	
207-208	Reference System	C*8	EME50 etc.
209-214	S/C Frame x,y,z of Picture Body	R*8	S/C EME
215-220	Central Body EME x,y,z of S/C Position	R*8	Central Body EME
221-226	Central Body EME relative S/C Velocity	R*8	Central Body EME
227	Scan Platform Clock Angle	R*4	

228	Scan Platform Cone Angle	R*4	
229-234	Earth Direction Unit x,y,z Vector	R*8	S/C EME
235-240	Sun Direction Unit x,y,z Vector	R*8	S/C EME
241-255	Spare		

WORD	ITEM (Imaging Geometry)	TYPE	COMMENTS
-----	-----	---	-----
256	PG Intersect Latitude of S/C x,y,z at Sfc	R*4	Central Body EME
257	PC Intersect Latitude of S/C x,y,z at Sfc	R*4	Central Body EME
258	Orbital Longitude of Picture Body	R*4	EME50
259	PG Sub-Sun Intersect Latitude at Sfc	R*4	Central Body EME
260	PC Sub-Sun Intersect Latitude at Sfc	R*4	Central Body EME
261	Sun Longitude	R*4	Central Body EME
262-263	S/C Range from Picture Body Center	R*8	
264-265	S/C Range to Earth	R*8	
266-267	S/C Range to Sun	R*8	
268-273	S/C x,y,z Velocity in Target Frame	R*8	
274	Smear Direction (deg relative to scan dir)	R*4	
275	Smear Velocity (km/s relative to tgt sfc)	R*4	
276	North Angle (deg relative to scan dir)	R*4	
277-294	3x3 Transform Matrix Inst--> Planet	R*8	
295-312	3x3 Transform Matrix Inst--> EME	R*8	
313-330	3x3 Transform Matrix PicBod EED--> EME	R*8	
331	Optic Axis RA	R*4	
332	Optic Axis Dec	R*4	
333-335	L-Vector (Az,El,Twist)	R*4	
336-338	Pitch, Yaw, and Roll Limit Cycle Angles	R*4	
339	Nadir Angle of Optic Axis	R*4	
340	Azimuth Angle of Optic Axis	R*4	
341-351	Spare		

WORD	ITEM (User Computed Quantities)	TYPE	COMMENTS
-----	-----	---	-----
352-353	User Block Type	C*8	e.g. SSEC-VGR
354-355	Navigation Status	C*8	
356-357	Navigation System Type	C*8	
358-375	Computed Transform Matrix Inst--> PicBod	R*8	
376	Computed North Angle	R*8	
377-394	Computed Transform Matrix Inst--> Planet	R*8	
395-412	Computed Transform Matrix Inst--> EME	R*8	
413-430	Computed Transform Matrix PicBod EED--> EME	R*8	
431-432	Computed Optic Axis RA	R*8	
433-434	Computed Optic Axis Decl	R*8	
435	Minnaert Fit Constant Slope	R*4	
436	Minnaert Fit Constant Intercept	R*4	
437	Minnaert Fit Constant Bias	R*4	
438	Minnaert Fit Constant Max DN	R*4	
439	PicBod Center Line	R*4	
440	PicBod Center Element	R*4	
441	Pixel Size on PicBod Surface	R*4	
442	RMS PicBod Radius Error for Navigation	R*4	
443	Sun Phase Angle at OptAxis/Sfc Intersect	R*4	
444	Best Fit Equatorial Radius of PicBod	R*4	

445	Best Fit Polar Radius of PicBod	R*4
446-447	Image/Navigation Processing Status	C*8
448-459	Spare	

WORD	ITEM (Supplementary Information)	TYPE	COMMENTS
460-461	User Block Type	C*8	SSECSATS, SSESTAR, SSECRING, etc.
462-511	Satellite & Shadow Positions Star Locations Rings		

Words 352-511 are user-specific and can change from one installation to another. The remaining 128 words (512-639) are the "old format" planetary navigation "codicil" corresponding to the separate navigation file entry in the MVS McIDAS system. This last block exists whether an image is raw or geometrically rectified and/or projected. The block is filled when an image is transferred from another area or tape and keyin SEDRIN is called with a SEDR file indicated. If this codicil block is not needed, the user-specific words run from 352-639.

If SEDRIN is called with SPICE kernel files indicated, then the words 128-351 in the navigation block are updated. If the PAA keyin is used, the current navigation is made identical to the codicil, with an empty SPICE block placed in between.

Note that calls to most planetary applications software generally modifies only the last 512 words in the 640 word navigation entry. The first 128 word block is filled in only two ways: 1) by a call to ACCESS, and then only when an image has been geometrically rectified, and 2) by a McIDAS map projection.

Navigation type: 4-byte ASCII. Is normally:

- 'PLAN' for planetary navigation images
- 'PS ' for polar stereographic projection
- 'LAMB' for Lambert Conformal projection
- 'MERC' for Mercator projection
- ' ' (or binary 0), 'ÇÇÇÇ' (or binary 80808080) for no navigation

If the type is 'PLAN', the succeeding words are the same as the McIDAS-MVS planetary navigation codicil format, described later.

To navigate a spacecraft image or area, the parameters contained in the first 128 words of the 'NAV ' block must be moved to an array called 'IARR'. The call ISTAT= NVXINI(1,IARR) will initialize the indicated navigation package (where X has the value 1, 2, or 3). If ISTAT is returned as 0, the initialization was successful. This must be done prior to using any of the routines in the navigation package.

To transform line and element from the spacecraft image to planet coordinates, use the call

```
ISTAT= NVXSAE(XLIN,XELE,0.,XLAT,XLON,XDUMMY).
```

The transformation calls allow for three input and three output arguments, not all of which need to be used. In transforming from line-element to latitude-longitude, the XDUMMY parameter is not needed, and the third dimension of the image (perpendicular to the image plane) is 0. If ISTAT= 0 the transformation was successful. If ISTAT= -1 the transformation was not possible (e.g. arguments were out of range, or not meaningful).

Inputs are

REAL*4 XLIN= image line number
REAL*4 XELE= image element number

Both inputs are based on full resolution. Outputs are

REAL*4 XLAT= latitude of pixel
REAL*4 XLON= longitude of pixel

North and East are positive for 'PLAN' navigation modules, regardless of planet. The map projections are historically based on Earth coordinates and use North and West as positive directions. The user must be careful to avoid inversions when map-projecting geometrically rectified planetary images. Often, the problem can be easily corrected by changing the sign of the spacecraft subpoint longitude and recalculating the C-matrix or calling subroutine CALCMA.

To transform planet coordinates (latitude, longitude) to spacecraft coordinates, call

ISTAT= NVXEAS(XLAT,XLON,0.,XLIN, XELE,XDUMMY).

The arguments and return values are the same as above.

Normally, transformations are made to and from latitude and longitude. If this is not desired, planet-based coordinates may be changed to rectangular coordinates (X,Y,Z). These coordinates have the origin at the planet center with the x-axis passing through the Equator at 0 degrees, the y-axis passing through the Equator at 90 degrees east (+90 deg.) and the z-axis passing through the North Pole. All planetary navigation routines use a right-handed coordinate system.

The call ISTAT=NVXINI(2,'XYZ') will cause the routines NVXSAE and NVXEAS to perform the following:

ISTAT=NVXSAE(XLIN,XELE,0.,X,Y,Z)

Transforms from two dimensional line-element coordinates to a three dimensional vector x,y,z where

XLIN= image line number
XELE= image element number
X,Y,Z are the rectangular coordinates described above.

2) The call:

ISTAT=NVXEAS(X,Y,Z,XLIN,XELE,XDUMMY)

with arguments as above, will transform from x,y,z to line-element.

It is possible to return to latitude longitude coordinates with the call

ISTAT= NVXINI(2,'LL').

The subroutine ACCESS has been modified so it will no longer read old format McIDAS area navigation blocks unless the areas are first modified by new keyin PAA (Planetary area to area mover), or provided the NFLAG variable has been set to the value 6 instead of the usual 5. Changing the NFLAG constant in the subroutine call to the values 6-9 will make all old applications software work with the new planetary area format and the new navigation options. For backward compatibility, leaving NFLAG=1-5 will allow use of the old navigation with the new area format. Using the old version of ACCESS instead of relinking will cause applications to break on new format areas. Great care has been taken to make the shift to the new area format as transparent and backward

compatible as possible while still protecting the programmer from acquiring data in the wrong places. The new format has a great deal more redundancy to accommodate this backward compatibility. Details on the use of calls to the new subroutine ACCESS follow:

CALL ACCESS(IAREA, NFLAG, CSPBLK, IV, ISTAT)

The new version of ACCESS has the additional character variable CSPBLK, which is used to specify the SPICE block format. Some instrument and image identification is stored in the SPICE block as part of the data ingestion by McIDAS-X from CD-ROM. If the user supplied value of CSPBLK doesn't agree with the value in words 128-129, the error value ISTAT=-1 is returned. In the initial implementation, the only valid values of CSPBLK are: 'SPICEVGR' and 'SEDRVGR'.

NFLAG <= 5 works the same as before, but the navigation now comes from the old mainframe codicil existing in the last 128 words of the navigation block. (The user must set CSPBLK='SEDRVGR') The source of navigation data is the same as on the mainframe (SEDR input), and keyin PAA will transfer the mainframe navigation codicil into those 128 words following any SENAA to a temporary intermediate area. Old McIDAS format areas with the first 128 words of navigation in PLAN format will now cause an aborted return with ISTAT=-1, because ACCESS will always be looking in location 512 instead, and will therefore see calibration or image data instead of PLAN navigation.

NFLAG=6 assumes that the current navigation is in PLAN format in a new format area and sets up planetary common blocks using the current navigation in the first 128 words of the navigation block. Missing values in words 0-127 are calculated by ACCESS as needed. This option can also be used to read the PLAN navigation in any old McIDAS format areas (replacing the NFLAG=5 option). (The user must set CSPBLK='SEDRVGR')

NFLAG=7 works like NFLAG=5, except that the "current navigation" (the first 128 words of the navigation block normally used by McIDAS applications) is now set up in PLAN format, with values from the codicil. Besides setting the parameter CSPBLK='SPICEVGR', using NFLAG=7 instead of NFLAG=5 in calls to ACCESS is the only modification needed to make all old planetary applications work with McIDAS-X and with new format areas. This code should eventually be moved to NVXPLA as well, of course, since NVXPLA always uses the first 128 words, but the old NVXPLA will still work if a call to ACCESS with NFLAG=7 is made to update current navigation before using McIDAS NVXPLA PLAN navigation routines.

NFLAG=8 works like NFLAG=7 to update current navigation except that the source of the navigation is now the SPICE kernels and other parameters stored in words 128-351 of the navigation block. Those words in the middle of the navigation block are filled when SPICEIN (the analog of SEDRIN) is called. Computed navigation information in words 352-431 is not used. The codicil data from the mainframe in words 512-639 is not used, and is not updated by SPICEIN. If SEDR data is never used, words 512-639 are undefined, and can have other user-defined purposes.

NFLAG=9 uses calculated and fitted transform quantities in words 352-431 rather than the SPICE kernels to update current navigation. This is equivalent to using information from the image itself rather than the pointing information in the SPICE kernels alone.

SUMMARY

Do SENAA or EDR/VICAR tape read to temporary old format McIDAS areas.

Use keyin PAA to generate new format planetary areas. (new step)

Use keyin SEDRIN after the call to PAA.

In the application, call ACCESS with new NFLAG values...

words 512-639 used	if NFLAG<=5 (old apps with new areas)
words 0-127 used	if NFLAG=6 (old apps with old areas)
words 0-127 and 512-639 used	if NFLAG=7 (NVXPLA with new areas)
words 0-351 used	if NFLAG=8 (SPICE nav, new areas only)
words 0-511 used	if NFLAG=9 (fitted nav, new areas only)

IV array is written back to words 0-127 in area for NFLAG=7,8,9.

(this is a new ACCESS operation, making PLAN nav "current")

New applications with new areas should normally use NFLAG=7,8,9.

Planetary navigation common is always filled in all cases where ACCESS returns ISTAT=0.

If reading images directly from CD-ROM, one should use the McIDAS-X keyin VCDROM, which reads directly into a new planetary area format. Keyin SPICEIN will be used to fill the navigation block, and the new version of ACCESS with calling argument CSPBLK should be used to fill navigation common blocks.

SPICE navigation requires some knowledge of reference frames. Every state vector returned from an ephemeris must be referenced to a recognized inertial reference frame. This requirement ensures that primitive states from different ephemerides can be combined to create more general states. (By forbidding states referenced to dynamically defined frames, this requirement also ensures that the ephemeris data stored in a file cannot be modified by definitions stored outside the file.)

The inertial reference frames used by SPICE are currently of two types:

1. New -- New frames are identical or nearly identical to the frame defined by the Earth mean equator and dynamical equinox of Julian year 2000, adopted by the IAU as the standard reference frame for all publications. The frame defined by the as yet unreleased FK5 star catalog will be in this category.

2. Old -- Old frames are close to the frame defined by the Earth mean equator and dynamical equinox of Besselian year 1950. The frames in this category are often referred to collectively by the name `EME50', although individual frames may be offset from each other by as much as half a second of arc. The name `B1950' is reserved for the frame defined by applying the J2000 precession model to the J2000 frame. The frame defined by the FK4 star catalog is offset from the B1950 frame by about half a second of arc.

The rotation between any two recognized frames can be obtained from subroutine IRFROT in SPICELIB by supplying the indexes of the frames taken from the following list:

Index	Name	Description
1	J2000	Earth mean equator, dynamical equinox of J2000
2	B1950	Earth mean equator, dynamical equinox of B1950
3	FK4	Fundamental Catalog (4)

CALIBRATION

Calibration data are specific to the spacecraft, the instrument, and the calibration procedures and software. We do not provide any such software except for Voyager 1 & 2. Full radiometric correction converts from DN values to watts/cm**2/ster, with a 9-point transfer function for each of the filter positions. The calibration data files occupy about 25 Mbytes of disk space, producing a two-byte per pixel output from single byte raw Voyager images. Simple "field flattening" using a linear two-point transfer function is often adequate for most jobs, and requires far less disk space to operate. It runs fast, removing "hot corners" and improving contrast, and produces a compact one byte per pixel image.

Keyin RF finds reseaus in Mariner and Voyager images, and places a set of nominal or found reseaus at the end of the calibration block. These reseaus are used by the GEOM keyin to produce a remapped image which has geometry very close to that of the focal plane image in the camera.

AUDIT TRAIL

The audit trail is written by every planetary McIDAS keyin using the subroutine PSTAMP, which appends to each output area a record of the day, time, keyin name, and keyin parameters used to modify the area in any way. Those modifications include any changes to the navigation or calibration information. The audit trail is read by using the LA keyin with the keyword FORM=AUDIT.

The first six lines of the audit trail contain the "image identification block", or IID block. This identifies the spacecraft, camera, planet, scan mode, etc. for the image contained in the area. This information is used by all planetary McIDAS keyins (via subroutine IIDBLK) to provide default values for keyin parameters. The user should be aware that this default information is always available to a keyin, and if it has to be changed for some reason, that change can only be input as part of the command line.

A planetary version of the AA keyin, called PAA, will convert any standard McIDAS format area to a new planetary area suitable for the SPICE-oriented navigation and two-byte pixels from radiometric conversion. The new areas remain completely backward compatible with the old as far as McIDAS utilities and core applications software is concerned, but use with some planetary applications requires slightly larger navigation and calibration blocks. PAA is also used to move planetary areas within McIDAS-X.

Using AA instead of PAA to transfer areas may not leave sufficient room for new navigation in the destination areas. AA always errors out without causing any damage. This issue will be resolved in the near future.

PLANETARY NAVIGATION CODICIL FORMAT FOR McIDAS-MVS

Type: LW-File (Area Codicil)
Size: 128 words

The software for planetary imaging navigation on the IBM 4381 is designed to be as compatible as possible with the existing satellite navigation on McIDAS. The codicil structure is fully compatible with the format of NX0NAV, (NX1NAV...).

Planetary flyby missions use a framing camera mounted on a three-axis stabilized spacecraft. The camera exposes a vidicon or solid state imaging array at one shutter time. Any framing camera image, including digitized Earth based telescope images (such as HST) is navigable using this file structure and associated software. Note that the images must be photometrically and geometrically corrected, however, before accurate measurements can be made. This file structure assumes ideal image and camera geometry.

Not all information need necessarily be present, since some is purposely redundant so that more than one way of calculating the transform matrix may be used. Each file entry may exist in three states defined as follows:

1. PRELIMINARY - Words 64-80 are filled with integer values representing scaled or unscaled values transferred from the SEDR file by load module SEDRIN, or input by the user using the keyins NAVPLA or DUPNAV.

2. INTERMEDIATE - Through use of the limb finder (LIMBPT) or user input via NAVPLA, followed by FINDCN, the codicil now contains measured and fitted bright limb points and additional derived quantities dependent on image geometry alone. Only image dependent calculations are possible with data in the codicil at this time, and the data is almost entirely in scaled and packed integers.

3. COMPLETE - All double precision words 7-63 are properly filled, using the CALCMA, SIMULN, FITMOS keyins. The latitude-longitude grid for the planet and the transformation matrix are defined. Calculations using planet coordinate geometry are now possible, using the derived quantities in the codicil.

IMAGE IDENTIFICATION:

1. Navigation type: 'PLAN' (4-bytes EBCDIC)

2. SSSYYDDD: Satellite type= 46 (Mariner 10)

47 (Voyager 1)

48 (Voyager 2)

49 (Galileo)

50 (Hubble Space Telescope)

YYDDD= Standard year and Julian day of shuttering

3. HHMMSS: Spacecraft shutter time (GMT)

4. PICNO: (DDDNNNN) An image identifier consists of two words, the first being a signed number DDDNNNN representing the days to encounter DDD (- is before encounter, + is after encounter) and a number NNNN that increments by one every 48 seconds during that day. The second word identifies the planet.

5. Planet/User ID: (PUUU) Planet initial followed by user's initials. The initial represents the target in the image, so a picture of Titan would be identified by 'T' rather than Saturn's 'S'.

6. FDS: Spacecraft assigned FDS number without embedded decimal point
7. LDATE,NSUBR: Julian day of file entry generation plus identifier of subroutine which made last entry (YYDDSS)
8. LTIME: GMT time of file entry generation (HHMMSS). Words 7-8 permit one to identify when a navigation was done, if it was an individual navigation, tiepoint navigation, or mosaic navigation, and which subroutine made the last entry into the codicil.

CALCULATED DOUBLE PRECISION VALUES:

- 9-26. R(3,3): Rotation matrix which transforms a vector r in planetocentric representation to a vector r' in the spacecraft representation $\{ r' = Rr - b \}$. The transpose of R , called T , does the inverse transform from spacecraft to planet coordinates $\{ r = T(r' + b) \}$.
- 27-32. C(3): Double precision vector b pointing from the spacecraft to the planet center, represented in spacecraft coordinates and normalized to the planet equatorial radius. This defines the translation vector b which accompanies the rotation matrix R for a complete transformation from spacecraft to planet and its inverse.
- 33-34. ROTANG: The orientation of the image scan lines relative to the spin axis of the planet, calculated clockwise from north (in the image plane, about the vector b) to the left pointing scan lines. (e.g. if north is up in the image than $ROTANG > \pi$)
- 35-36. SINALP: Double precision sine of the angle subtended by the planet's semi-major axis at the spacecraft, as determined by the navigation process. This variable determines the distance scale on the planet.
- 37-38. SSPLAT: Double precision planetocentric subspacecraft latitude on the planet's surface.
- 39-40. SSPLON: Double precision subspacecraft longitude on the planet's surface.
- 41-42. SUNLAT: Double precision planetocentric subsolar latitude on the planet's surface.
- 43-44. SUNLON: Double precision subsolar longitude on the planet's surface.
- 45-46. PCLIN: Double precision line origin of spacecraft coordinates in the image (line-element) coordinate system. For single frame navigation, this origin is arbitrarily set at the center of the image frame. For navigation of simultaneous images, the origin is at the center of the wide angle camera frame, but is given in narrow angle frame coordinates.
- 47-48. PCELE: Double precision element origin of the spacecraft coordinate system in image (line-element) coordinates.
- 49-50. RDPLIN: Double precision angle in radians subtended by one scan line at the image center defined by (PCLIN,PCELE).
- 51-52. RDPELE: Double precision angle in radians subtended by one picture element at the image center defined by (PCLIN,PCELE).
- 53-54. SHSTRA: Double precision shrink-stretch angle of rotation in the image plane about the vector $-b$, calculated clockwise from north. Using the planet oblateness and LSPLAT, an observed eccentricity for the planet's figure in the image is calculated. A two-dimensional orthogonal matrix E is calculated, which distorts the image aspect ratio in the two directions defined by the angle SHSTRA so that points on the limb of an oblate sphere (the planet image) appear to be points on a true sphere (the navigation least squares model). After least

squares iteration, the transpose of E is applied to the limb points and planet center to get the fitted planet center for the oblate planet in image coordinates. The E matrix is not used after determination of the fitted planet center, since vector intersections with an oblate sphere model are then directly calculated.

INPUT INTEGER AND SCALED INTEGER VALUES:

55. LP5LAT: P5 point planetographic latitude (optic axis intercept).
56. LP5LON: P5 point longitude. (optic axis intercept -- reseau 101)
57. LSSRAS: Right Ascension of the sun or a star given in HHMMSS.S times 10, at the shutter time when LSUN or LSTAR is defined.
58. LSSDEC: Declination of the sun or a star given in DDMMSS, at the shutter time.
59. LAXRAS: Right Ascension of the planet spin axis given in HHMMSS.S times 10, at shutter time.
60. LAXDEC: Declination of the planet spin axis in DDMMSS, at the shutter time.
61. LTYPE: CAMERA ID (W,N,A, OR B) AND FILTER NUMBER (0-7) ENTERED as left justified EBCDIC, blank filled.
62. LSPTY: Defines type of each of the following two defining vectors for the navigation transform. (LSPTY1*10000 + LSPTY2*10 + LGEO1*1000 + LGEO2) is the packed representation where:
LSPTY=vector type= 10 landmark
 20 subsolar point
 30 subspacecraft point
LGEO=coordinate type= 0 planetocentric
 1 planetographic
Standard input from the SEDR for individually navigated images would normally use LSPTY=301201.
63. LATSP1: Latitude of intersection of vector u with visible planet surface.
64. LONSP1: Longitude of intersection of vector u with visible planet surface.
65. LINSPI/IELSP1: Packed line and element of vector u' in image coordinates (LINSPI*10000 + IELSP1). Word 65 is unused if 63-64 are definition of subspacecraft point and the planet center is to be found from bright limb points.
66. LATSP2: Latitude of intersection of vector z with visible planet surface.
67. LONSP2: Longitude of intersection of vector z with visible planet surface.
68. LINSPI2/IELSP2: Packed line and element of vector v' in image coordinates (LINSPI2*10000 + IELSP2). If word 68 is unused, one of NORANG, LSUN, or LSTAR must be used to orient the planet grid.
69. NORANG: Angle of planet spin axis in image with respect to scan line direction (i.e. defines 'north'). See word 33 for definition of convention. This angle, if used as input, is defined as -ROTANG, but ROTANG is always produced by navigation software as a calculated output quantity. Of 68-71, this is the normal input for Voyager navigation. This angle is input in ten-thousandths of a degree, as a scaled integer (i.e. degrees*10000).

70. LSUN: Direction to the sun in the image coordinate system, measured clockwise from the upward pointing normal to the scan lines. Input is in degrees*10000. (e.g. for Mariner 10, with clock-cone scan platform coordinates, this angle is 270 +/- 0.25 degrees, input as +2700000)
71. LSTAR: Direction to a specified star (Canopus, Vega, Polaris, etc) in image coordinates, measured clockwise from the upward normal to the scan lines. Input as a scaled integer in ten thousandths of a degree. (i.e. degrees*10000)
72. LSPLAT: Planetocentric latitude of spacecraft point, either estimated from predicts or given in the SEDR. Input is in scaled integer degrees*10000.
73. IDIST: Distance from the spacecraft to the center of the planet in kilometers. If word 74 is non-zero the distance used by the navigation software is $IDIST = IDIST * 10^{LEXDST}$.
74. LEXDST: Exponent of 10 to multiply IDIST by for very large (i.e. > 2,000,000,000 km) distances.
75. LINTOT/IELTOT: Image size in lines and elements packed as #lines*10000 + #eles.
76. LINDEG: Angular height of image in millionths of a degree input as a scaled integer degrees*1000000.
77. IELDEG: Angular width of image in millionths of a degree input as a scaled integer degrees*1000000.
78. IRAD: Assigned equatorial radius (semi-major axis) of planet in kilometers at the height of the cloud deck or surface observed at the limb or terminator. This value may vary slightly with choice of filter color or exposure time.
79. ICENLN: Line coordinate of planet center (*1000) to be used for input/output to the planet center finder. This may be used to test alternate hypotheses for the center by observing the changes in the least squares residuals.

See FINDCN keyin for more details.

80. ICENEL: Element coordinate of planet center (*1000)

MEASURED LIMB POINTS, TERMINATOR POINTS, TIE POINTS:

81. LMN: Packed number of limb points (L), terminator points(M), or tie points(N) to follow in immediate succession. Values packed as $L*10000 + M*100 + N$.

- 82-125. Points as follows:

L limb points packed as $line*10000 + element$ followed by:

M terminator points packed as $lin*10000 + ele$ followed by

N tie points in groups of three

as follows:

word 1 = LPICNO of tied frame

word 2 = $line*10000 + element$ of tie point in this frame

word 3 = $line*10000 + element$ of tie point in tied frame LPICNO

126. NAIFID: NAIF ID of the target body (image body to which the R-matrix transforms).

127. ITILT: Tilt angle of camera optic axis away from the vector from the planet center to the spacecraft. Input is in scaled integer degrees*10000

128. IAZIM: Azimuth of the ITILT vector (optic-axis) relative to the spacecraft coordinate z-axis. If ITILT=IAZIM=0 then the spacecraft z-axis is up in a Voyager image and the y-axis is in the negative element (left) direction. Input is in scaled integer degrees*10000

Several derived quantities are calculated by subroutine ACCESS when a navigation codicil is being unpacked into common blocks.

These are:

A(3,3) Intermediate matrix used in TRNSFM to get B(3,3)
D(3) Intermediate vector used in TRNSFM
YFACT=SINALP
XFACT=1/SINALP
COSALP=DSQRT(1 - YFACT**2)
SNALPA=IRAD/(IDIST*10.**LEXDST)
IHGT=IDIST - IRAD
ROT=LSUN
LECC=1000*(eccentricity of planet LPLAN)
LPOL=polar semi-axis

APPENDIX II

This appendix illustrates the use of McIDAS-X for accessing Voyager images from the Planetary Data System CD-ROM's. It is basically a log of the McIDAS-X output that is sent either to the screen or directed to the printer or a log file at user's command. The actual key-in is highlighted in bold characters. Explanatory comments are italicized.

The user wants to process a number of images between two given FDS counts (Flight Data System, basically a system clock that is updated every 48 seconds for Voyager spacecraft) be read from a mounted Voyager PDS CD-ROM. The images are to be stored sequentially in areas beginning with number 6001, and only Saturn images are to be ingested.

```
VCDROM FDS = 34699.00 34699.50 AREA = 6001 TARGET = SATURN
```

VCDROM: BEGIN SEARCH FOR INDICATED IMAGES

DECOMPRESSING DATA...

```
PICNO= 1528S1-009          TARGET= SATURN
FDS= 34699.24      CAMERA= NARROW ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 200434          BOTSIM
FILTER= BLUE (2) EXP= 1.4400 SEC.   GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1  EDIT= 1/1
FINISHED OUTPUT AREA= 6001
```

DECOMPRESSING DATA...

```
PICNO= 1531S1-009          TARGET= SATURN
FDS= 34699.27      CAMERA= WIDE ANGLE  SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 200434          BOTSIM
FILTER= BLUE (1) EXP= 0.3600 SEC.   GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1  EDIT= 1/1
FINISHED OUTPUT AREA= 6002
```

DECOMPRESSING DATA...

```
PICNO= 1535S1-009          TARGET= SATURN
FDS= 34699.31      CAMERA= NARROW ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 201010          BOTSIM
FILTER= VIOLET (1) EXP= 1.9200 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1  EDIT= 1/1
FINISHED OUTPUT AREA= 6003
```

DECOMPRESSING DATA...

```
PICNO= 1538S1-009          TARGET= SATURN
FDS= 34699.34      CAMERA= WIDE ANGLE  SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 201010          BOTSIM
FILTER= VIOLET (3) EXP= 0.9600 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1  EDIT= 1/1
FINISHED OUTPUT AREA= 6004
```

DECOMPRESSING DATA...

```
PICNO= 1541S1-009          TARGET= SATURN
FDS= 34699.37      CAMERA= NARROW ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 201458          NAONLY
```

FILTER= ORANGE (3) EXP= 2.8800 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1 EDIT= 1/1
FINISHED OUTPUT AREA= 6005

DECOMPRESSING DATA...

PICNO= 1546S1-009 TARGET= SATURN
FDS= 34699.42 CAMERA= NARROW ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 201858 BOTSIM
FILTER= BLUE (2) EXP= 1.4400 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1 EDIT= 1/1
FINISHED OUTPUT AREA= 6006

DECOMPRESSING DATA...

PICNO= 1549S1-009 TARGET= SATURN
FDS= 34699.45 CAMERA= WIDE ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 201858 BOTSIM
FILTER= ORANGE (7) EXP= 1.9200 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1 EDIT= 1/1
FINISHED OUTPUT AREA= 6007

DECOMPRESSING DATA...

PICNO= 1553S1-009 TARGET= SATURN
FDS= 34699.49 CAMERA= NARROW ANGLE SPACECRAFT= VOYAGER 1
SHUTTER TIME= 80309 202434 BOTSIM
FILTER= VIOLET (1) EXP= 1.9200 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800 SCAN= 3:1 EDIT= 1/1
FINISHED OUTPUT AREA= 6008

VCDROM: DONE -- 8 FILES DECOMPRESSED
8 AREAS GENERATED

Eight images of Saturn were found on the CD-ROM between the input FDS range and they were stored in Areas 6001 through 6008.

At this point the images can be displayed and subjected to other processing that does not depend on the image geometry or calibrated radiances. Examples of these include low and high pass filtering, blow ups and blow downs etc. However, for any quantitative use, the images have to be calibrated and navigated. Several commands have been created for this purpose. Illustrations of their use follow.

A single Saturn image is first linked with the Supplementary Experiment Data Record (SEDR) information, the image distortion is then determined by finding the Reseau locations, the image is corrected for distortion by digital remapping and finally, navigated by center-finding from measured limb points.

The first step in the processing of the Voyager images is to attach to the images the corresponding navigation information from the SEDR or SPICE files. The navigation information in the SEDR format consists of the spacecraft sub-point latitude and longitude, range to the picture body (which in this case is the same as the target body, Saturn), the position of the sub-solar point, and the north angle, i.e. the angle between the spin axis of Saturn and the line scan direction of the images. The SEDR file ("VGRIS") is assumed to exist on the system. Commands SPICEIN and SEDRIN are created to accomplish these tasks.

SEDRIN 6001 X VGR1S

SEDRIN: BEGIN ACQUISITION OF SEDR DATA FOR AREA 6001
 NO CURRENT PLAN NAVIGATION BLOCK EXISTS FOR AREA 6001

READING SEDR FILE "VGR1S"
 SEARCHING SEDR FILE FOR FDS COUNTS FOLLOWING 3469914

FDS	SHUTTER	TIME	PB	S/C-BODY	P5	LAT-LON	SUB	S/C	LAT-LON	NORTH	PHA
3469914	80309	195633	6	11216583	1.14	215.86	9.95	305.86	315.	13.	
3469918	80309	195945	6	11213630	-9.97	219.43	9.95	307.67	315.	13.	
3469924	80309	200433	6	11209200	-29.09	295.07	9.95	310.37	338.	13.	

CAMERA FOV FROM "CAMERA=" KEYWORD:
 10001000 450216 450325
 (VOYAGER 1 NARROW ANGLE CAMERA)

PLAN NAVIGATION BLOCK FOR AREA6001 92310 174033

1	NAVTYPE	PLAN	SYDDDD	0	HMMSS	0	PICNO	0		
5	USER	lima	FDS	3469924	LDATE	9231002	LTIME	174033		
9	R(3,3)	.000000D+00					.000000D+00			
13		.000000D+00					.000000D+00			
17		.000000D+00			.000000D+00					
21		.000000D+00					.000000D+00			
25		.000000D+00			B(3)		.000000D+00			
29		.000000D+00					.000000D+00			
33	ROTANG	.000000D+00			SINALP		.000000D+00			
37	SSPLAT	.000000D+00			SSPLON		.000000D+00			
41	SUNLAT	.000000D+00			SUNLON		.000000D+00			
45	PCLIN	.000000D+00			PCELE		.000000D+00			
49	RDPLIN	.000000D+00			RDPELE		.000000D+00			
53	SHSTRA	.000000D+00			LP5LAT	-290950	LP5LON	2950690		
57	LSSRAS	0	LSSDEC	0	LAXRAS	0	LAXDEC	0		
61	LTYPE		LSPTYP	301201	LATSP1	99473	LONSP1	496321		
65	LIN-EL	0	LATSP2	37423	LONSP2	372435	LIN-EL	0		
69	NORANG	676486	LSUN	0	LSTAR	0	LSPLAT	99473		
73	IDIST	11209200	LEXDST	0	LIN-EL10001000		LINDEG	450216		
77	IELDEG	450325	IRAD	60330	ICENLN	0	ICENEL	0		
81	LMN	0	POINTS	0		0		0		
85	0		0			0		0		
89		0		0		0		0		
93		0		0		0		0		
97		0		0		0		0		
101		0		0		0		0		
105		0		0		0		0		
109		0		0		0		0		
113		0		0		0		0		
117		0		0		0		0		
121		0		0		0		0		
125		0	NAIFID	699	NADIR	0	NAZIM	0		

SEDRIN: END PROGRAM -- NAVIGATION BLOCKS WRITTEN FOR AREA 6001

The SEDRIN key-in has properly filled in the camera parameters for the Voyager 1 camera and the sub-point information. At this point the system knows about the image characteristics such as spacecraft, camera, picture body, viewing geometry. The geometric and photometric corrections need to be made to the image before proceeding with limb finding for navigation.

First the geometric distortion needs to be removed. To do this, we make use of the reference marks on the vidicon tube whose faceplate and object space locations are known. The knowledge of the image locations of the Reseau marks defines the transformation that will allow the corrections to be made to the image. The Reseau locations are found with the RF key-in. RF makes a first pass through the image to search for the reseau marks in the vicinity of the "nominal" locations. The Reseau marks are expected to be within a reasonable distance of their nominal locations (usually no more than 20 pixels). Usually not all the Reseaus can be located for a variety of reasons. Their locations are interpolated based on the distortion defined by the Reseaus that have been found, by making a second pass to check for the Reseau locations.

RF 6001

Identified NARROW ANGLE camera and SPACECRAFT= 47

SEARCH BOX SIZE 15 X 15

INPUT AREA= 6001

MATCHING RESEAU POINTS

FIRST PASS (Unadjusted)

RESEAU 27	RESLIN=	-.7	RESELE=	.6	TOTAL=	.9
RESEAU 38	RESLIN=	-.6	RESELE=	-.3	TOTAL=	.7
RESEAU 44	RESLIN=	-.8	RESELE=	-.4	TOTAL=	1.0
RESEAU 48	RESLIN=	-.7	RESELE=	-.3	TOTAL=	.8
RESEAU 202	RESLIN=	-1.0	RESELE=	.3	TOTAL=	1.1
RESEAU 53	RESLIN=	-1.1	RESELE=	-.2	TOTAL=	1.1
RESEAU 54	RESLIN=	.1	RESELE=	.8	TOTAL=	.8
RESEAU 66	RESLIN=	.6	RESELE=	.1	TOTAL=	.7
RESEAU 69	RESLIN=	-.8	RESELE=	.1	TOTAL=	.8
RESEAU 75	RESLIN=	-.1	RESELE=	-.7	TOTAL=	.8
RESEAU 76	RESLIN=	-.2	RESELE=	.7	TOTAL=	.7
RESEAU 79	RESLIN=	-.6	RESELE=	.7	TOTAL=	.9
RESEAU 84	RESLIN=	-.8	RESELE=	.0	TOTAL=	.8
RESEAU 133	RESLIN=	-.3	RESELE=	.6	TOTAL=	.7
RESEAU 139	RESLIN=	-.7	RESELE=	-.4	TOTAL=	.8
RESEAU 154	RESLIN=	.1	RESELE=	-.6	TOTAL=	.6
RESEAU 189	RESLIN=	-1.2	RESELE=	-.5	TOTAL=	1.3
RESEAU 13	RESLIN=	.7	RESELE=	-.5	TOTAL=	.8
RESEAU 77	RESLIN=	-.5	RESELE=	-.7	TOTAL=	.9
RESEAU 66	RESLIN=	-.5	RESELE=	-.9	TOTAL=	1.0
RESEAU 126	RESLIN=	.7	RESELE=	.1	TOTAL=	.7
RESEAU 76	RESLIN=	-1.0	RESELE=	-.3	TOTAL=	1.1
RESEAU 91	RESLIN=	.1	RESELE=	-.7	TOTAL=	.7
RESEAU 136	RESLIN=	.5	RESELE=	.7	TOTAL=	.8
RESEAU 151	RESLIN=	.3	RESELE=	-.7	TOTAL=	.7
RESEAU 80	RESLIN=	-1.0	RESELE=	-.5	TOTAL=	1.1
RESEAU 95	RESLIN=	.2	RESELE=	-.6	TOTAL=	.7
RESEAU 15	RESLIN=	.0	RESELE=	-.6	TOTAL=	.6
RESEAU 132	RESLIN=	-.2	RESELE=	-.6	TOTAL=	.6

RESEAU 148	RESLIN=	.0	RESELE=	-.7	TOTAL=	.7
RESEAU 133	RESLIN=	.0	RESELE=	-.6	TOTAL=	.6
RESEAU 75	RESLIN=	-.8	RESELE=	.0	TOTAL=	.8
RESEAU 150	RESLIN=	-.1	RESELE=	-.7	TOTAL=	.7
RESEAU 79	RESLIN=	-1.0	RESELE=	-.1	TOTAL=	1.0
RESEAU 189	RESLIN=	.1	RESELE=	.7	TOTAL=	.8

SECOND PASS (Adjusted to line up with neighbors)

RESEAU 137	RESLIN=	.3	RESELE=	.6	TOTAL=	.7
RESEAU 148	RESLIN=	.7	RESELE=	.0	TOTAL=	.7
RESEAU 154	RESLIN=	-.7	RESELE=	.2	TOTAL=	.7
RESEAU 155	RESLIN=	-.8	RESELE=	.3	TOTAL=	.9
RESEAU 189	RESLIN=	-.7	RESELE=	-.1	TOTAL=	.8
RESEAU 148	RESLIN=	.0	RESELE=	-.7	TOTAL=	.7

RESEAU POINTS FOUND 158

At this point the image and object space locations of the Reseau marks are known and the original can be digitally mapped to the object space by determining the transformation between the image space and object space as defined by the Reseau locations. This remapping is done with the GEOM key-in which first determines the transformation on a regular grid from the irregularly spaced Reseau locations.

GEOM 6001 6011

```

Begin geometric rectification for area= 6001
  Output area= 6011
Output area is square with sides= 1000
REMAP - TRANSFORMATIONS COMPLETE..BEGIN DATA MOVE
--REMAP DONE

```

The resulting image has been stored in ARea 6011 and is devoid of geometric distortions, i.e. the distance is uniform in terms of the image pixels in any direction. This can be verified by plotting the object space location of the reseaus on a graphics frame and examining the correspondence with their signature in the image. The key-in RESEAU accomplishes this task and also lists the Reseau locations.

RESEAU PLOT OBJ

```

PLOTting RESEAU S FOR VOYAGER  NARROW ANGLE CAMERA
THESE ARE OBJECT SPACE RESEAU S IN THE GEOMED IMAGE
PLOTting RESEAU S FOR VOYAGER  NARROW ANGLE CAMERA
  1 ( 25.3, 25.2) ( 20.4, 85.5) ( 25.3,177.5) ( 25.3,269.8) ( 25.3,361.8)
  6 ( 25.3,454.0) ( 25.3,546.0) ( 25.3,638.0) ( 25.3,730.2) ( 25.3,822.3)
 11 ( 20.4,914.6) ( 25.3,974.9) ( 39.5, 39.5) ( 51.3,131.3) ( 51.3,223.6)
 16 ( 51.3,315.8) ( 51.3,407.9) ( 51.3,500.0) ( 51.3,592.0) ( 51.3,684.1)
 21 ( 51.3,776.3) ( 51.3,868.5) ( 39.5,960.7) ( 85.6, 20.5) ( 85.6, 85.5)
 26 ( 85.6,177.5) ( 85.6,269.8) ( 85.6,361.8) ( 85.6,454.0) ( 85.6,546.0)
 31 ( 85.6,638.0) ( 85.6,730.2) ( 85.6,822.3) ( 85.6,914.6) ( 85.6,979.7)
 36 (131.6, 51.2) (131.6,131.3) (131.6,223.6) (131.6,315.8) (131.6,407.9)
 41 (131.6,500.0) (131.6,592.0) (131.6,684.1) (131.6,776.3) (131.6,868.5)
 46 (131.6,948.7) (177.7, 25.2) (177.7, 85.5) (177.7,914.6) (177.7,974.9)
 51 (223.7, 51.2) (223.7,131.3) (223.7,223.6) (223.7,315.8) (223.7,407.9)

```

56 (223.7,500.0) (223.7,592.0) (223.7,684.1) (223.7,776.3) (223.7,868.5)
61 (223.7,948.7) (269.9, 25.2) (269.9, 85.5) (269.9,914.6) (269.9,974.9)
66 (315.8, 51.2) (315.8,131.3) (315.8,223.6) (315.8,315.8) (315.8,407.9)
71 (315.8,500.0) (315.8,592.0) (315.8,684.1) (315.8,776.3) (315.8,868.5)
76 (315.8,948.7) (361.8, 25.2) (361.8, 85.5) (361.8,914.6) (361.8,974.9)
81 (407.9, 51.2) (407.9,131.3) (407.9,223.6) (407.9,315.8) (407.9,407.9)
86 (407.9,500.0) (407.9,592.0) (407.9,684.1) (407.9,776.3) (407.9,868.5)
91 (407.9,948.7) (454.1, 25.2) (454.1, 85.5) (454.1,914.6) (454.1,974.9)
96 (500.0, 51.2) (500.0,131.3) (500.0,223.6) (500.0,315.8) (500.0,407.9)
101 (500.0,500.0) (500.0,592.0) (500.0,684.1) (500.0,776.3) (500.0,868.5)
106 (500.0,948.7) (545.9, 25.2) (545.9, 85.5) (545.9,914.6) (545.9,974.9)
111 (592.1, 51.2) (592.1,131.3) (592.1,223.6) (592.1,315.8) (592.1,407.9)
116 (592.1,500.0) (592.1,592.0) (592.1,684.1) (592.1,776.3) (592.1,868.5)
121 (592.1,948.7) (638.3, 25.2) (638.3, 85.5) (638.3,914.6) (638.3,974.9)
126 (684.3, 51.2) (684.3,131.3) (684.3,223.6) (684.3,315.8) (684.3,407.9)
131 (684.3,500.0) (684.3,592.0) (684.3,684.1) (684.3,776.3) (684.3,868.5)
136 (684.3,948.7) (730.2, 25.2) (730.2, 85.5) (730.2,914.6) (730.2,974.9)
141 (776.4, 51.2) (776.4,131.3) (776.4,223.6) (776.4,315.8) (776.4,407.9)
146 (776.4,500.0) (776.4,592.0) (776.4,684.1) (776.4,776.3) (776.4,868.5)
151 (776.4,948.7) (822.3, 25.2) (822.3, 85.5) (822.3,914.6) (822.3,974.9)
156 (868.5, 51.2) (868.5,131.3) (868.5,223.6) (868.5,315.8) (868.5,407.9)
161 (868.5,500.0) (868.5,592.0) (868.5,684.1) (868.5,776.3) (868.5,868.5)
166 (868.5,948.7) (914.6, 20.5) (914.6, 85.5) (914.6,177.5) (914.6,269.8)
171 (914.6,361.8) (914.6,454.0) (914.6,546.0) (914.6,638.0) (914.6,730.2)
176 (914.6,822.3) (914.6,914.6) (914.6,979.7) (960.6, 39.5) (948.8,131.3)
181 (948.8,223.6) (948.8,315.8) (948.8,407.9) (948.8,500.0) (948.8,592.0)
186 (948.8,684.1) (948.8,776.3) (948.8,868.5) (960.6,960.7) (974.8, 25.2)
191 (979.8, 85.5) (974.8,177.5) (974.8,269.8) (974.8,361.8) (974.8,454.0)
196 (974.8,546.0) (974.8,638.0) (974.8,730.2) (974.8,822.3) (979.8,914.6)
201 (974.8,974.9) (177.7,730.2)

RESEAU: DONE

The image although not corrected for photometric distortions, is now ready to be navigated, i.e. to determine the transformation matrix between the image co-ordinates (line and element numbers of a given pixel) and the planet centered co-ordinates (latitude and longitude). This is accomplished by first finding the limb points. There are commands available that use general conic fits to determine the unconstrained center by determining the shape of the figure from the input limb points, its orientation and size. If the figure of the picture body is known with some confidence, that knowledge can be used to constrain the center finding as is done in the commands PLAEDG and CALCMA. A first guess center location is determined when more than 3 limb points are determined. Assuming the figure of the planet, the center location is refined by using a constrained fit.

PLAEDG key-in allows a user to place a cursor on the planet's bright limb and determine the limb edge points. When 3 or more points are available, the center is printed out as well as the difference between the expected radius to the input points from the calculated center and the actual radius to the limb point. These computations are made in an equivalent space in which the oblateness of the planet is explicitly accounted for and simplified by a sphere of largest radius that will fit within the perspective view of the planet.

PHELP PLAEDG

PLAEDG: Find or list limb points in displayed planet image or area

PLAEDG <AREA> <POINT#> <GRAD> <LINOFF> <IELOFF> LIN= ELE= NUM=
PLAEDG 'L' or 'LIST' (Lists current limb points for current area)
PLAEDG <AREA> BLOB=MINDN MAXDN LSKIP IESKIP NUM=
IF KEYWORD BLOB IS SPECIFIED, CENTER OF BRIGHTNESS
IS USED TO COMPUTE SEARCH LIMITS FOR LIN, ELE

AREA = area (>30) to search for points (def=displayed frame)

POINT# = 0 FIND MULTIPLE limb points in area (default)

> 0 FIND a SINGLE limb point on the line LIN=

or within the cursor if LIN is undefined

(added after last valid limb point found, so

number may be lowered on graphics overlay)

< 0 REMOVE a SINGLE current limb point

GRAD = local gradient value defining limb

(default=50 for 2 byte data, 10 for single byte)

LINOFF = line offset to found limb points

IELOFF = element offset to found limb points

KEYWORDS:

LIN = beg end line bounds (default BEG=50 END=last-50)

ELE = beg end element bounds (def BEG=40 END=neles-3)

NUM = number of limb points to find (default=15)

(maximum is 30, higher numbers interpreted as area)

BLOB = mindn minimum dn bound for brightness center

maxdn maximum dn bound for brightness center

lskip line skip increment for sampling dn's (def=5)

leskip element increment for sampling dn's (def=5)

EXAMPLES: PLAEDG (tries to find 15 limb points for displayed frame)

PLAEDG -3 (removes point 3 in displayed frame)

PLAEDG NUM=30 (tries to find 30 limb points in frame)

PLAEDG 5 LIN=230 (point 5 added, as found on line 230)

PLAEDG 5 (point 5 added within cursor)

PLAEDG 6465 (tries to find 15 limb points for area 6465)

PLAEDG 200 3 (adds point 3 in cursor to area 200)

PLAEDG -30 (remove non-existent point to see trial fit again)

PLAEDG 1 X 3

**** McIDAS-X PLANET LIMB DETECTOR ****

DISPLAYED FRAME IS 2

CORRESPONDING AREA IS 6011

USING CURSOR TO DEFINE SINGLE LIMB POINT ON LINE 71

1- LINE= 71 ELEMENT 274 GRADIENT= 11

Filing Limb Points in area 6011

CHECK FIT -- POINTS INPUT 1

Not enough bright limb points - THREE required

Filing Calculated Planet Center in area 6011

Old Center was at: .000 .000

PLAEDG: DONE FOR AREA 6011

PLAEDG 2 X 3

**** McIDAS-X PLANET LIMB DETECTOR ****

DISPLAYED FRAME IS 2
CORRESPONDING AREA IS 6011
USING CURSOR TO DEFINE SINGLE LIMB POINT ON LINE 127
2- LINE= 127 ELEMENT 277 GRADIENT= 11
Filing Limb Points in area 6011
CHECK FIT -- POINTS INPUT 2
Not enough bright limb points - THREE required
Filing Calculated Planet Center in area 6011
Old Center was at: .000 .000

PLAEDG: DONE FOR AREA 6011

PLAEDG 3 X 3

**** McIDAS-X PLANET LIMB DETECTOR ****

DISPLAYED FRAME IS 2
CORRESPONDING AREA IS 6011
USING CURSOR TO DEFINE SINGLE LIMB POINT ON LINE 190
3- LINE= 190 ELEMENT 288 GRADIENT= 10
Filing Limb Points in area 6011
CHECK FIT -- POINTS INPUT 3
TRIAL FIT:
71 274 -34.822 KM
127 277 65.049 KM
190 288 -30.714 KM
RADIUS ERROR -0.163 KM
ITERATIONS= 8 LINE-ELE: 87.912 908.391
Filing Calculated Planet Center in area 6011
Old Center was at: .000 .000

PLAEDG: DONE FOR AREA 6011

Three limb points are the minimum necessary for the least-squares determination, however, for oblate objects these are insufficient for a good center determination. Typically more than 20 points are necessary to stabilize the solution using the constrained fit.

PLAEDG 29 X 3

**** McIDAS-X PLANET LIMB DETECTOR ****

DISPLAYED FRAME IS 2
CORRESPONDING AREA IS 6011
USING CURSOR TO DEFINE SINGLE LIMB POINT ON LINE 153
#29- LINE= 153 ELEMENT 281 GRADIENT= 11

Filing Limb Points in area 6011

CHECK FIT -- POINTS INPUT 29

TRIAL FIT:

71	274	-134.529 KM
127	277	28.002 KM
190	288	3.537 KM
247	302	56.927 KM
295	318	68.585 KM
344	338	96.215 KM
366	348	135.009 KM
386	360	3.919 KM
419	380	-87.198 KM
452	399	70.453 KM
476	418	-73.675 KM
512	447	-113.479 KM
531	462	-11.596 KM
553	484	-94.544 KM
575	506	-60.170 KM
596	529	-30.440 KM
610	546	-34.892 KM
625	565	-13.952 KM
637	582	-33.859 KM
654	608	-57.752 KM
671	636	-45.174 KM
685	660	29.935 KM
697	682	120.318 KM
709	709	127.456 KM
401	368	24.182 KM
321	329	10.737 KM
266	308	53.411 KM
214	294	-32.563 KM
153	281	5.754 KM

RADIUS ERROR 0.366 KM

ITERATIONS= 7 LINE-ELE: 78.646 906.859

Filing Calculated Planet Center in area 6011

Old Center was at: 78.654 906.868

PLAEDG: DONE FOR AREA 6011

The center location is deemed to be satisfactory in the sense that addition of each new limb point changes the center location by an appreciable amount (less than 0.5 pixel). Figure A1 shows how quickly the center line and element locations converge as a new limb point is added. In the example shown when more than 10 limb points are used, the constrained fit center changes by less than 0.5 pixels and by less than 0.2 pixels after about 24 limb points.

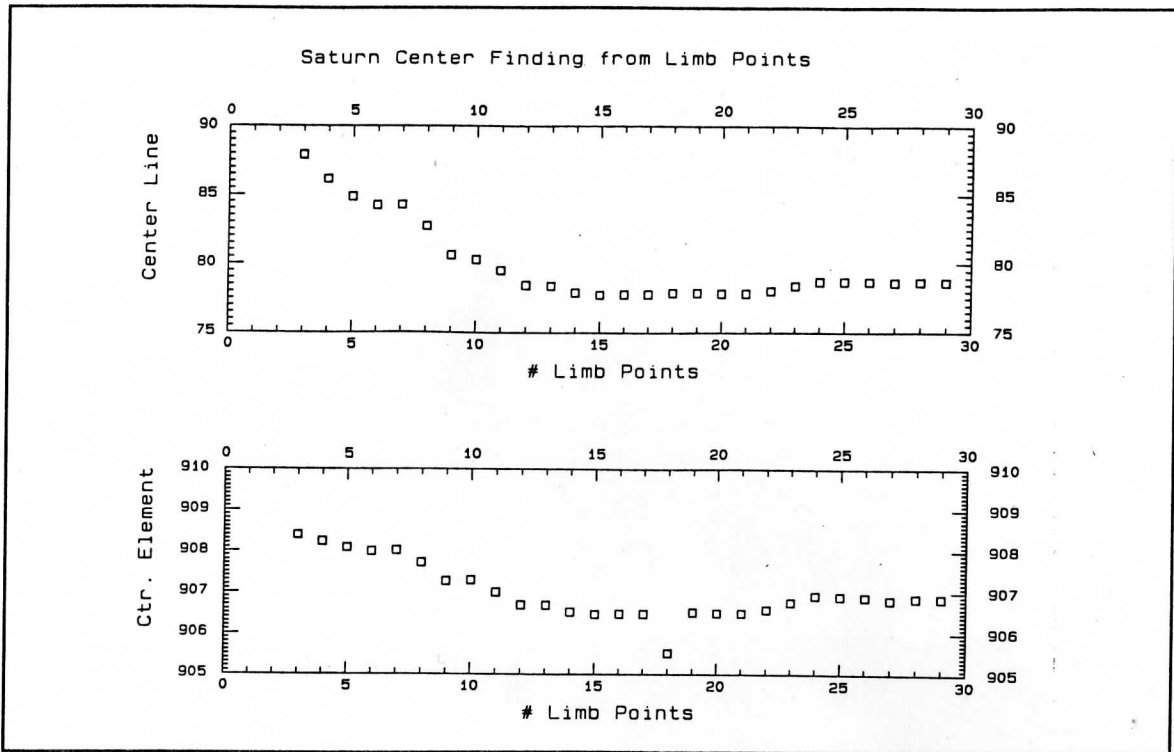


Figure A1. Change in the image center line and element location that corresponds to the sub-spacecraft point as it changes with increasing number of limb points using the constrained fit.

The transformation between the image co-ordinates and the planet co-ordinates can now be computed using the CALCMA key-in. CALCMA writes the necessary information that McIDAS-X recognizes so that all existing commands that deal with the geographical co-ordinates such as map or latitude-longitude grid, frame loads etc., can now operate with the planetary data.

CALCMA

*** PLANETARY NAVIGATION MATRIX CALCULATION ***

CALCULATING PLANETARY NAVIGATION MATRIX FOR AREA 6011

CALCMA: Try without point : 7

Spin Axis +.202E+03 deg

Transform Matrix :

-.99999E+00 +.31978E-02 -.33109E-02
+.18461E-02 -.38029E+00 -.92487E+00
-.42166E-02 -.92486E+00 +.38028E+00

Vector to Spacecraft:

+.20370E+03 +.00000E+00 +.00000E+00

Planet-Spacecraft Vector 11209200.0 km

CALCMA: PLAN NAVIGATION UPDATED FOR AREA= 6011

MAP LALO

MAP: DONE

The processing history of the image can be looked by using the LA keyin with the FORM=AUDIT option.

LA 6011 FORM=AUDIT

```
area  ss  yyddd hhmss  lcor  ecor  lr  er  zr  lsiz  esiz  z  bands
-----
6011  47  80309 200434   1    1  1  1  1  1000 1000 1 1.....
```

```
proj:    0 created: 92310 174044 memo: ---CDROM DECOMPRESS---
type:VISR   cal type:BRIT
area offsets: data= 9040 navigation= 256 calibration= 2816
doc length:  0 cal length:  0 lev length:  0 PREFIX=  0
valcod:      0 zcor:  1 band-8: NA reel#:*****
```

-Audit Trail

yyddd hhmss

```
-----
PICNO= 1528S1-009          TARGET= SATURN
FDS= 34699.24          CAMERA= NARROW ANGLE          SPACECRAFT= VOYAGER
SHUTTER TIME= 80309 200434          BOTSIM
FILTER= BLUE (2) EXP= 1.4400 SEC. GAIN= LOW
LINES= 800 ELEMENTS= 800          SCAN= 3:1 EDIT=
```

VCDROM: Internal Parameters and Defaults:

```
SOURCE      = /cdrom/saturn/c3469xxx/c3469924.imq
FDS-START   = 3469900
FDS-END     = 3469950
TARGET      = saturn
CAMERA      =
FILTER      =
SEARCH_INDEX= 0
OUTPUT_AREA = 6001
92310 173839 VCDROM FDS= 34699.00 34699.50 AREA= 6001 TARGET= SATURN
92310 174033 SEDRIN 6001 X VGR1S
92310 174038 RF 6001
92310 174058 GEOM 6001 6011
92310 174205 PLAEDG 1 X 3
92310 174212 PLAEDG 2 X 3
92310 174218 PLAEDG 3 X 3
92310 174631 PLAEDG 29 X 3
92310 174634 CALCMA
```

LA: Done