ISS Calibration Analysis Support System

Charles Avis
Jan Yoshimizu

California Institute of Technology
Jet Propulsion Laboratory, Mail Stop 168-514
4800 Oak Grove Drive, Pasadena, CA 91109

ABSTRACT

The newest generation of JPL imaging experiments, the Cassini Imaging Science Subsystem, required calibration analysis effort beyond that of its predecessor instruments. This called for streamlining the data reduction process with automation and flexibility while using software inherited from support of the Galileo Solid State Imaging instrument. A set of enhancements was implemented which automated many tasks, tracked their inputs and outputs and generated easily distributable products,

Keywords: instrument calibration, image processing, Cassini

1. INTRODUCTION

The Cassini mission is sending a suite of remote sensing instruments into orbit around Saturn. Among these instruments, a pair of CCD imagers make up the Imaging Science Subsystem (ISS). The Narrow-angle Camera is a new design with 24 filters and a 1024X1024 sensor producing 12-bit data values. The Wide-angle Camera uses spare Voyager optics with J 8 filters, but identical electronics with the Narrow-angle Camera[1].

These devices required testing and calibration prior to launch in order to:

- verify that the design requirements had been met,
- find problems with hardware or software that could be fixed pre-launch,
- characterize the system response over the range of operating conditions,
- determine the various aspects of the instrument signature that appear in the data

The vast array of calibrated test equipment, thermal/vacuum chambers, light sources, and computer hardware and software necessary to produce the raw calibration image data is not covered in this paper. The aspects covered here involve the computer hardware and software necessary to analyze the thousands of test images produced while calibrating each instrument. Emphasis is given to enhancements to the Galileo analysis software in the areas of automation and products.

2. ANALYSIS TASKS

Of the many and varied activities it takes to produce a fully calibrated instrument, the calibration analysis support discussed here is that which involves processing of the image data. The following list outlines the analyses performed on each camera system and was derived from instrument requirements[2].

- Data acquisition, cataloging and archiving
- System gain constant derivation
- Shutter-offset compensation
- Navigation target data distribution
- Linearity and sensitivity analysis
- Pixel-by-pixel radiometric response derivation
- Entropy effect correlation
- Noise spectrum analysis
- Modulation Transfer Function derivation
- Focus analysis
- Geometric distortion function validation
- Point Response Function analysis
- Light leak analysis
- Polarization angle derivation
- Dark-current characterization
- Sensor blemish analysis
- Calibrated flat field production

3. THE FACILITY

The facility at the Jet Propulsion Laboratory (JPL) known today as the Multimission Image Processing Subsystem (M 11'S) has been calibrating JPL's imaging instruments since the days of Mariner 6 and 7. Although the computer platforms are radically different today, many of the basic calibration software applications arc still in usc. For instance, projects still need Fourier Transforms to check noise and derive Modulation Transfer Functions. The calibration reports for Mariner 9[3], Mariner 10[4], Viking Orbiter[5] and Voyager[6] show an evolving capability for calibration of sensor systems. However, it was the previous project, Gal ileo, which produced proven analysis software for characterizing CCD sensors[7]8.

4. BASIS OF THE SYSTEM DESIGN

1 bring the period of support for Cassini 1SS calibration, the M 11'S facility was transitioning from a VAX/VMS environment to a distributed UNIX environment. The facility's new components (both hardware and software) were coming on-line and the old proven VAX/VMS architecture was being decommissioned. The timing of this transition effort relative to the 1SS support period was a major factor in tile design of the calibration capability.

The large task of porting the existing Galileo programs into the developing MIPS UNIX environment in time for Cassini support (including extensive software testing) looked like a budgetary anti schedule problem. The alternate approach was to maintain anti enhance the VAX/VMS capability while augmenting the existing calibration programs with reasonable levels of automation.

5. SYSTEM COMPONENTS

5.1 Proven software

The software baseline going into the Cassini task was that resulting from tbc Galileo calibration of tbc 1980's. The VAX/VMS software performed most of the tasks required for Cassi ni but by most] y non-automated means. This capability covered most of the tasks listed above with the exception of cataloging and archiving, polarizat ion and focus. The plotting capability, however, was inflexible anti hardwired into the applications and each task required extensive setup time.

la addition, the rest of the MI 1'S VICAR image processing system[9] was also part of the VAX/VMS software baseline. The VICAR libraries of applications programs were available for the more gene.ral tasks of image processing, such as image arithmetic, FFTs, statistics, scripting (procedures), and display.

5.2 Hardware

An inexpensive VAX/VMS platform (VAXstat ion 4000/90) was acquired when the M 11'S VAX platforms that Galileo had u sed were replaced by UNI X hard ware. In this way, the VAX/VMS software baseline could be maintained in its working condition anti enhancements begun immediately. In its final configuration, this system's peripherals included:

- 19 Gbytes magnetic disk
- 2 rewritable optical cartridge drives
- 1 CD-ROM drive
- 1 8-mm magnetic tape backup device

5.3 MIPS Services

Adding this dedicated VAX/VMS platform to the MI I'S distributed system made available the. other new MIPS capabilities. The M I i'S fiber-optics network linking the plat fores with each other anti with the outside world was already in place. A database server was already available on tbc network which would serve both UNIX and VMS client machines. So, adding the client database software to this cal i brat ion workstat icm immediatel y established a cataloging capabilit y. Therefore, onl y a minimal effort was required to maintain the baseline of working Galileo softwar c on a networked machine with potential database capability.

6. DATA ACQ1JIS1"1'1ON ANI) FORMAT

An important activity which had a great influence upon the design and operation of the calibration system was the early interact ion of M IPS with the 1SS development team. This allowed the downstream M 11'S users of the calibration data to have input in decisions and requirements. Design choices were made very car] y which greatly influenced the calibration operations and how MIPS got its analysis done.

The ISS development team used the Engineering Ground Support Equipment (EGSE) computer system to support the actual image data taking. It performed the camera commanding and built the image files from the camera's telemetry stream. The data flow from the EGSE to the MIPS workstation (Figure 1) was envisioned to use the existing JPL networks because the two machines were not colocated. However, several EGSE loading and usage issues precluded the regular direct transfer of data to MIPS. Typically, the data was transferred to a second IWS11 (oat which was not interacting with the camera) via optical disk before being transferred to the MIPS workstation via File Transfer Protocol (FTP).

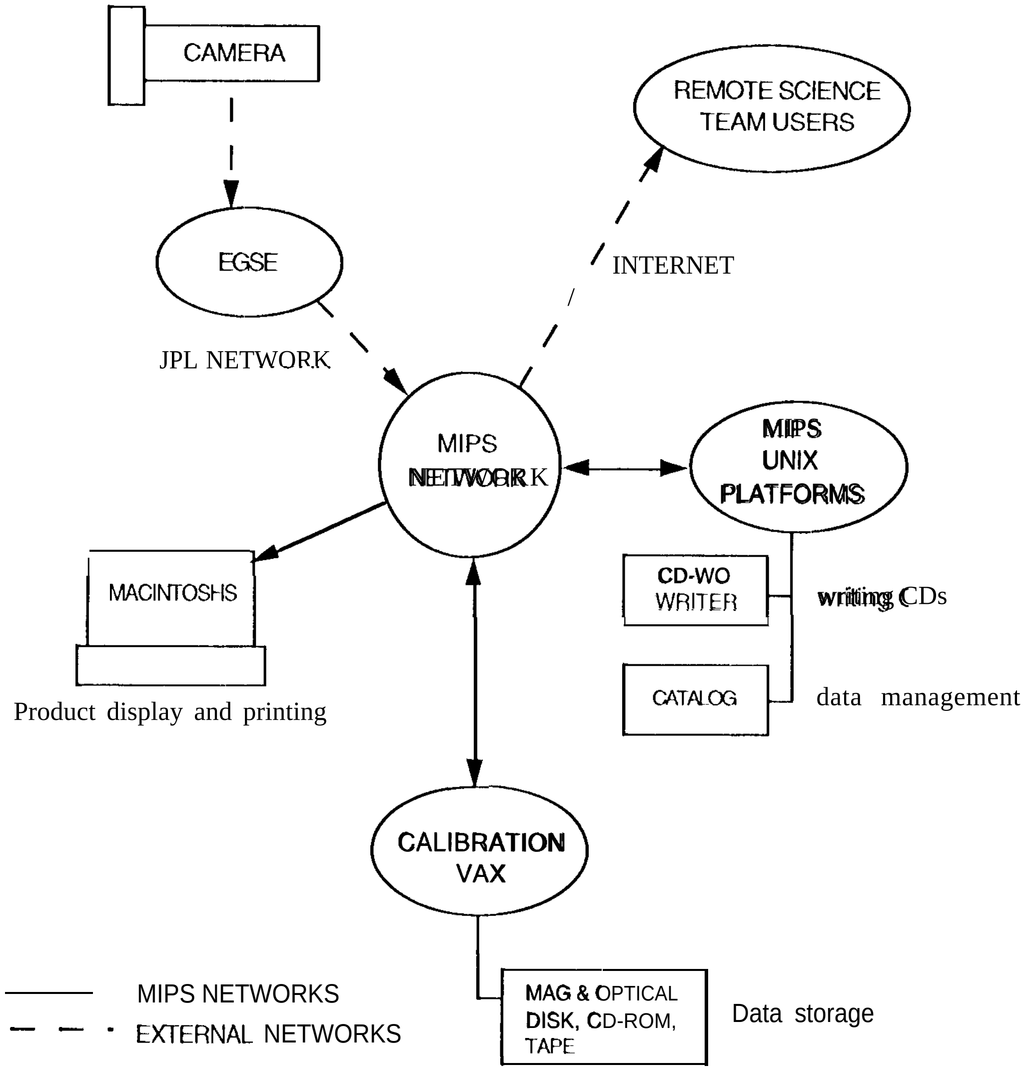

Fig. 1. Networked computers was the key to raw data transfer and storage, data management and product delivery

Particularly critical was the choice. of image format used by the EGSE. The fact that the EGSE was designed to generate images in VICAR format meant that these images were already in a "MIPS-native" format when they arrived on the workstation's disk. The VICAR labels of the images constructed by the EGSE contained ASCII keyword=value pairs similar to Planetary Data System labels. These human- and machine-readable items were part of the image files and would be updated with processing history information as necessary by the VICAR executive. In addition, this executive handled all input/output such that it was invisible to the VAX user that the image files were created on a foreign machine (the EGSE was a SPARCstation10). See Figure 2 for the list of VICAR label items produced by the EGSE at image creation time.

The data-taking process at the EGSE labeled all images as being part of unique observations. For instance, the VICAR label of all Light Transfer data taken at + 25°C in gain state 2 with filters CL1 and CL2 was given an observation_id of "LIGHT_TRANSFER_2 O 5". Similar tests with different filters, gain or temperature would be given different observation- id values. Because of this initial labeling, the independent sets of data resulting from the various tests were easily identified throughout the analysis process. The role of observation information in automation will be discussed further in Section 7.4.

7. CALIBRATION ANALYSIS ENHANCEMENTS FOR 1 S S

The usc of the baseline of working Galileo software allowed time for development of automation turd cataloging capabilities. Most of this development effort went into the following four areas:

- catalog and interface
- improved output products
- process automation
- product archiving

7.1 Catalog

The most critical enhancement of the analysis capabilities was the USC. of an image catalog database. No other improvement caused so much increase in productivity. Previous calibration efforts had no database capabilities. However, the Voyager support at MI I'S enjoyed an image catalog from the start of the Uranus support. These years of experience with image catalogs were used to design a catalog and user interface which would allow tasks to be highly automated.

The M 1 }'S facility had a Sybase database system in place for support of multiple projects. The server (a UNIX platform) serviced requests from various Sybase client machines, including our Cassini calibration VAX, over a local network, In addition, an interactive user interface was in place before the development for Cassini occurred. This meant that, once established, the calibration image catalog tables could be queried and accessed by users immediately. Only the lSS-specific soft ware interface needed to be developed.

7.1.1 Tables

Appropriate catalog tables and fields were defined for the specific requirements of 1SS calibration. Each record in the catalog tables referred to a single data file, whether of image, tablular, or other t ypc. 1 'or image data, almost all information 10 be store.cl in the catalog came directly from the descriptive items of the VICAR label except for the disk location of the file. A table called 'main' contained these characteristics and the location of the raw data. The correspondence between lhc VICAR label and the 'main' fields is shown in Figure 2.

VICAR LABEL ITEM NAME	table field name
IMAGE_NUMBER	sclk
MISSION_PHASE_TYPE	phase
OBSERVATION_ID	observation
IMAGE_TIME	eventyear
	[illegible]
	eventtime
TARGET_NAME	target
INSTRUMENT_ID	camera
INSTRUMENT_MODE_ID	mode
EXPOSURE_DURATION	expos
FILTER1_NAME	filter1
FILTER2_NAME	filter2
GAIN_MODE_ID	gain
LIGHT_FLOOD_STATE_FLAG	lightflood
ANTIBLOOMING_STATE_FLAG	antiblooming
CALIB_LAMP_STATE_FLAG	callamps
CONVERSION_TYPE	conversion
ENCODING_TYPE	encoding
GROUP_BLOCKS	blocks
ALGORITHM	algorithm
BLOCK_TYPE	btype
QUANTIZATION_FACTOR_INDEX	qfactor
COMPRESSION_RATIO	compratio
ILLUMINANT	illuminant
RADIANCE	radiance
OPTICS_TEMPERATURE	opticstemp
FILTER_TEMPERATURE	filtertemp
DETECTOR_TEMPERATURE	ccdtemp
MISSING_LINES	missinglines
COMMENT	comment
	quality
	diskid
	directory
	file
	extension
	archfile
	created
M M	
'W	
M M	

Fig. 2. This table shows the correspondence between the values in an image's VICAR label and its record in the catalog.

A table called 'products' stored all image, tabular and other files produced by the automated procedures. 'l'his product archiving task is discussed further in Section 7.3. A table called 'calfiles' stored all files that later programs would need to do further calibration processing. These files of critical reduced data stored information such as shutter-offse,t, dark-crrrrcnt and radiometric response.

7.1.2 Catalog Usage

Storing, modifying and extracting information from the catalog tables involved only a small sc(of software modules. All these modules were ncw and involved sending commands and queries to the rc.mote. UNIX database. server and receiving response.s.

Tbc program CATALOGER was written to store information on images newly acquired from tbc EGSE. The user merely had to give CATALOGER a text file containing the filenames of the images to catalog (a simple directory listing). CATALOGER would read the VICAR label items and submit that information to the Sybase server for storage in the table 'main'. Similarly, SAVEISS and SAVECAL stored items in the 'products' and 'calfiles' tables.

Modifications to tbc catalog could be done interactively via tbc existing interface program DBVIEW or with the ncw program MODCAT. Any modifications were always done in conjunction with modifications to tbc VICAR labels so that both sets of information remained consistent.

Users could extract information from the catalog tables by using DBVIEW, or through the ncw program CALRPT. CALRPT generated reports containing user-selected fields from user-sclectcd sets of records. The records could bc selected by 'observation_id' or by 'sclk (i e., image_number)' range.

The program GIMME was designed as the single interface between the catalog and the application programs. It returned a text file containing a list of all filenames making up a specified observation. Among the options available, it could sort the list by exposure time, ignore poor quality frames, or return frames only with a given exposure time. In addition, it could return files with similar observation or could return the off-line archive locations of image.s. GIMME's kcy usage in automated procedures is discussed below in Section 7.4.

7.2 New output products

The reduced data of previous projects often had to be extracted from the log files or printouts from various processing jobs. In addition, all the plots were done on a line printer with no flexibility of presentation. A critical enhancement for Cassini calibration involved augmenting many kcy programs with tbc addition of tabular data output tiles. These contained tab-delimited ASCII data values representing the results. The addition of these files alone transformed the output products of calibration from "this is what tbc software can give you" to "what kind of plot would you like?". The format of these products being ASCII tat>-delimited text made thcm essentially gene.ric platform-independent products that could be easily distributed to other team members.

Commercial off-tbc-shelf plotting tools were selected to provide enhanced information display over the Galileo capabilities. These were for running on Macintosh platforms to complement the VAX image-oriented display capabilities. The plotting applications accepted the tab-delimited ASCII tables produced by the analysis programs and procedures.

The applications Spyglass Plot (for tabular data) and Spyglass Transform (for array data) were found to be flexible in data presentation, fast and allow large numbers of points. Their capabilities were more than required and with zero development cost. Operating in a Macintosh environment allowed laser printer output products, postscript plot files for e-mailing, delivery of tables on Macintosh or IBM diskettes and sharing files across a network with the other users.

7.3 Product Archiving

As files of products were generated by the calibration procedures, they were copied from the user's directory to an archive storage location and cataloged by SAVEISS. This relieved the users of much of tbc data tracking and saving process. No storage mix-ups occurred even when the same procedure was run multiple times, because each version of a product was given a unique name constructed from a counter value called the 'linkid'. For example, a gain table was produced each time tbc System Gain procedure was run but each table would have a unique name like GAIN. TBL14 or GAIN. TBL95. In addition, the log or other files from tbc same System Gain job which made GAIN.TBL14 would also have the linkid '14' in their filename. l'bus, all saved files resulting from a given job were associated by the 'linkid' in their filenames. In addition, the catalog maintained the products' location and heritage. The 'linkid' value was incremented for each run and maintained separately for each analysis procedure.

7.4 **Process Automation**

The catalog contained all tbc information necessary to associate image filenames with the various tests ('observations') run by tbc EGSE. Therefore, the arduous and error-prone task of typing filenames into procedures could be eliminated by using an interface between tbc catalog and the procedures. Instead of installing a catalog interface in each program which needed files, tbc ncw program GIMME was used as the single interface point. As described earlier, it would query the catalog for an observation and return a list of files to process. This had the effect of allowing users to select tests to process rather than deciding which images were required for a given analysis task.

The nature of the calibration job meant that each calibration task operated either on a group of images simultaneously or operated the same way on multiple individual images sequentially. For instance, a System Gain task derives a function from a set of images of various exposure times. On the other hand, each image's noise spectrum is produced independently of all other images, but many examples are required. Either way, the analysis procedures which run the application programs require a distinct set of files to process for cacb test,

For those programs needing multiple frames simultaneously, GIMME's list was submitted directly to the application program. It was a minor enhancement of the Galileo programs to allow them to accept lists instead of relying on the user to put all the names on a command line. Example 1 shows tbc basic structure of this type of analysis procedure.

in the example, after getting the user-specified parameters, the procedure accesses a file containing the 'linkid' which will be used to make unique filenames for tbc output products. The 'linkid' value is incremented and stored in the file. Next, GIMME is run to extract from the catalog tbc filenames for the observation being processed. These files arc then processed as a group by a program which makes an output table.

The output product (usually onc or more tab-delimited ASCII table), GIMME's list of files and perhaps other files produced by the procedure arc all given filenames using the 'linkid' to associate them. The ncw program SAVEISS then catalogs these files for future usc with characteristics common to their input images (e.g., gain, temperature, etc.).

Example 1: Processing of Image Groups - A table (e.g., signal vs. noise) is produced for each observation (e.g., images at many exposure times).

PROCEDURE
get parameter values from procedure call (e.g., Observation to process)
get current value of 'linkid' from file
increment 'linkid' and store value back in file

GIMME queries catalog for certain Observation; outputs list of filenames to process

potential preprocessing of files in list (e.g., dark-current subtraction)
application program processes list of files returning onc or more tables of results

copy table to archival location
SAVEISS catalogs tables' location
SAVEISS catalogs list output by GIMME

END-PROCEDURE

For those tasks needing multiple files to be processed sequentially, Examples 2A and 2B show the basic structure of this kind of procedure. The two examples produce distinctly different kinds of tabular output. Again, the procedure begins by accessing tbc parameters and the current value of the 'linkid', followed by GIMME extracting from the catalog the filenames for the observation being processed. These files arc processed sequentially via a loop. Each loop processes the next file in GIMME's list and create.s a record (Example 2A) or a column (Example 2B) in an output table. Programs in the loop are supplied each filename by the procedure as needed.

Example 2A: Processing of Images Sequentially - A scalar value (e.g., mean signal) is returned for each image of an observation. A table of values vs. image number is created.

PROCEDURE
get parameter values from procedure call (e.g., Observation to process)
get current value of 'linkid' from file
increment 'linkid' and store value back in file

GIMME queries catalog for certain Observation; outputs list of filenames to process

LOOP
get next filename from list
get the image number from the VICAR label
application program processes this file returning a scalar value
append this value and the image number to growing columns of values (a two-column table)
END-LOOP

copy table to archival location
SAVEISS catalogs table's location
SAVEISS catalogs list output by GIMME

END-PROCEDURE

Example of resulting table:

IMAGE_NUMBER	VALUE
123456	0.1
123457	0.2
...	...
123460	0.9

Example 2B: Processing of Images Sequentially - A table (e.g., amplitude vs. frequency) is returned for each image of an observation. A multi-column table of values vs. image number is created.

PROCEDURE
get parameter values from procedure call (e.g., Observation to process)
get current value of 'linkid' from file
increment 'linkid' and store value back in file

GIMME queries catalog for certain Observation; outputs list of filenames to process

LOOP
get next filename from list
get the image number from the VICAR label
application program processes this file returning a table of results
concatenate the table's columns to a growing multi-column table
END-LOOP

copy table to archival location
SAVEISS catalogs table's location
SAVEISS catalogs list output by GIMME

END-PROCEDURE

Example of resulting table:

FREQUENCY	IMG123456	IMG123457	. . .	IMG123460
0.0	1000.	1000.	...	1000.
0.1	200.	150.	...	160.
...	...	...	...	...
0.5	10.	9.	...	12.

8. CONCLUSION

The ISS calibration analysis task at MIPS was not developed from scratch. New hardware and networking capabilities were brought together with upgraded versions of proven software. The cataloging d'the raw and processed files enabled process automation anti flexibility to make the effort responsive to the instrument development team.

9. ACKNOWLEDGEMENTS

The authors wish to thank Tom Liverm ore, Cindy Kahn and Bill Harris foi guiding the 1SS team through the instrument development process. Thanks also go to Carolyn Porco's Science Team who spent considerable effort validating the enhanced MIPS analysis software. In addition, David Stokes, Young Kwon and Dimitri Haeff put the system described herein through its paces while calibrating both the Engineering and the Flight Model versions of the 1SS cameras,

10. REFERENCES

1. C. Kahn, *Cassini Imaging Science Subsystem (ISS) Subsystem Description Document*, JPL CAS-5-2036, 1993.
2. W. King, *Cassini Imaging Science Subsystem (ISS) Instrument Calibration Requirements Document - Subsystem Level Calibration*, JPL 699-CAS-4-2036-CAL, 1993.
3. L. M. Snyder, *Mariner 9 TV Subsystem Calibration Report*, JPL 610-202, 1971.
4. M. Benesh and M. Morrill, *MVM'73 TV Subsystem Calibration Report*, JPL 615-148, 1975.
5. M. Benesh and T. Thorpe, *Viking Orbiter 1975 Visual Imaging Subsystem Calibration Report*, JPL 611-125, 1976.
6. M. Benesh and P. Jepsen, *Voyager Imaging Science Subsystem Calibration Report*, JPL 618-802, 1978.
7. H. Breneman and K. Klassen, *Galileo Solid-state Imaging Subsystem Calibration Report*, Part 1, JPL 625-210, 1988.
8. K. Klassen, *Galileo Solid-state Imaging Subsystem Calibration Report, Part 2*, JPL 625-210, 1993.
9. E. Duxbury and D. Jensen, *VICAR User's Guide*, JPL D-4186 Rev. B, 1994.

The research described in this paper was carried out by the Jet Propulsion Laboratory, Califoi nia Institute of Technology, under a contract with the National Aeronautics anti Space Administration.