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 18 filters, but identical electronics with the Narrow-angle Camera.

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's 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:

- 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 are still in use. For instance, projects still need Fourier Transforms to check noise and derive Modulation Transfer Functions. The calibration reports for Mariner 9, Mariner 10\textsuperscript{a}, Viking Orbiter\textsuperscript{b} and Voyager\textsuperscript{c} show an evolving capability for calibration of sensor systems. However, it was the previous project, Galileo, which produced proven analysis software for characterizing CCD sensors\textsuperscript{d}.

4. BASIS OF THE SYSTEM DESIGN

A bring the period of support for Cassini ISS 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 ISS 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 and 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 the Galileo calibration of the 1980's. The VAX/VMS software performed most of the tasks required for Cassini but by mostly non-automated means. This capability covered most of the tasks listed above with the exception of cataloging and archiving, polarimetry and focus. The plotting capability, however, was inflexible and hard wired into the applications and each task required extensive setup time.

In addition, the rest of the M 11'S VICAR imaging system\textsuperscript{e} was also part of the VAX/VMS software baseline. The VICAR library of applications programs were available for the more general tasks of image processing, such as image arithmetic, FFTs, statistics, scripting (procedures), and display.

5.2 Hardware

An inexpensive VAX/VMS platform (VAXstation 4000/90) was acquired when the M 11'S VAX platform that Galileo had used was replaced by UNIX hardware. In this way, the VAX/VMS software baseline could be maintained in its working condition and enhancements begun immediately. In its final configuration, this system's peripherals included:

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

5.3 MIPS Services

Adding this dedicated VAX/VMS platform to the M 11'S distributed system made available the other new MIPS capabilities. The M 11'S fiber-optics network linking the platforms with each other and with the outside world was already in place. A database server was already available on the network which would serve both UNIX and VMS client machines. So, adding the client database software to this calibration workstation immediately established a cataloging capability. Therefore, only a minimal effort was required to maintain the baseline of working Galileo software on a networked machine with potential database capability.

6. DATA ACQUISITION AND FORMAT

An important activity which had a great influence on the design and operation of the calibration system was the early interaction of MIPS with the ISS 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 carefully, which greatly influenced the calibration operations and how MIPS got its analysis done.
The ISS development team used the ISS Ground Support Equipment (GSE) computer system to support the actual image datataking. It performed the camera commanding and built the image files from the camera's telemetry stream. The data flow from the GSE to the MIPS workstation (Figure 1) was envisioned to use the existing JPL networks because the two machines were not colocated. However, several GSE loading and usage issues precluded the regular direct transfer of data to MIPS. Typically, the data was transferred to a second IWS11 (one which was not interacting with the camera) via optical disk before being transferred to the MIPS workstation via File Transfer Protocol (FTP).

Figure 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 GSE. The fact that the GSE 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 GSE 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 GSE; was a SPARCstation10). See Figure 2 for the list of VICAR label items produced by the GSE at image creation time.

The data-taking process at the GSE 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 C1.1 and C1.2 was given an observation_id of "LIGHT_TRANSFER_2_O5". 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 ISS

The use of the baseline of working Galileo software allowed time for development of automation and 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 use 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 M1'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 M1'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 ISS-specific software interface needed to be developed.

7.1.1 Tables

Appropriate catalog tables and fields were defined for the specific requirements of ISS calibration. Each record in the catalog tables referred to a single data file, whether of image, tabular, or other type. For image data, almost all information had to be stored 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 the VICAR label and the 'main' fields is shown in Figure 2.
<table>
<thead>
<tr>
<th>VICAR LABELITEM NAME</th>
<th>table field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE_NUMBER</td>
<td>selk</td>
</tr>
<tr>
<td>MISSION_PHASE_TYPE</td>
<td>phase</td>
</tr>
<tr>
<td>OBSERVATION_ID</td>
<td>observation</td>
</tr>
<tr>
<td>IMAGE_TIME</td>
<td>eventyear</td>
</tr>
<tr>
<td></td>
<td>eventtime</td>
</tr>
<tr>
<td>TARGET_NAME</td>
<td>target</td>
</tr>
<tr>
<td>INSTRUMENT_ID</td>
<td>camera</td>
</tr>
<tr>
<td>INSTRUMENT_MODE_ID</td>
<td>mode</td>
</tr>
<tr>
<td>EXPOSURE_DURATION</td>
<td>expos</td>
</tr>
<tr>
<td>FILTER1_NAME</td>
<td>filter1</td>
</tr>
<tr>
<td>FILTER2_NAME</td>
<td>filter2</td>
</tr>
<tr>
<td>GAIN_MODE_ID</td>
<td>gain</td>
</tr>
<tr>
<td>LIGHT_FLOOD_STATE_FLAG</td>
<td>lightflood</td>
</tr>
<tr>
<td>ANTIBLOOMING_STATE_FLAG</td>
<td>antibleoming</td>
</tr>
<tr>
<td>CALIB_LAMP_STATE_FLAG</td>
<td>callamsn</td>
</tr>
<tr>
<td>CONVERSION_TYPE</td>
<td>conversion</td>
</tr>
<tr>
<td>ENCODING_TYPE</td>
<td>encoding</td>
</tr>
<tr>
<td>GROUP_BLOCKS</td>
<td>blocks</td>
</tr>
<tr>
<td>ALGORITHM</td>
<td>algorithm</td>
</tr>
<tr>
<td>BLOCK_TYPE</td>
<td>btype</td>
</tr>
<tr>
<td>QUANTIZATION_FACTOR_INDEX</td>
<td>qfactor</td>
</tr>
<tr>
<td>COMPRESSION_RATIO</td>
<td>compratio</td>
</tr>
<tr>
<td>ILLUMINANT</td>
<td>illuminant</td>
</tr>
<tr>
<td>RADIANCE</td>
<td>radiance</td>
</tr>
<tr>
<td>OPTICS_TEMPERATURE</td>
<td>opticstemp</td>
</tr>
<tr>
<td>FILTER_TEMPERATURE</td>
<td>filtertemp</td>
</tr>
<tr>
<td>DETECTOR_TEMPERATURE</td>
<td>ccdtemp</td>
</tr>
<tr>
<td>MISSING_LINES</td>
<td>missinglines</td>
</tr>
<tr>
<td>COMMENT</td>
<td>comment</td>
</tr>
<tr>
<td></td>
<td>quality</td>
</tr>
<tr>
<td></td>
<td>diskid</td>
</tr>
<tr>
<td></td>
<td>directory</td>
</tr>
<tr>
<td></td>
<td>file</td>
</tr>
<tr>
<td></td>
<td>extension</td>
</tr>
<tr>
<td></td>
<td>archfile</td>
</tr>
<tr>
<td></td>
<td>created</td>
</tr>
</tbody>
</table>

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. This 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-offset, dark-current and radiometric response.

7.1.2 Catalog Usage
Storing, modifying and extracting information from the catalog tables involved only a small set of software modules. All these modules were new and involved sending commands and queries to the remote. UNIX database server and receiving responses.

The program CATALOGER was written to store information on images newly acquired from the EGSB. 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, SAVISS and SAVICAL stored items in the ‘products’ and ‘calfiles’ tables.

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

Users could extract information from the catalog tables by using DBVIEW or through the new program CALRPT. CALRPT generated reports containing user-selected fields from user-selected sets of records. The records could be selected by ‘observation_id’ or by ‘slck’ (i.e., ‘image_number’) range.

The program GIMMI 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 offline archive locations of images. GIMMI’s key 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 key programs with the addition of tabular output files. 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 the software can give you” to “what kind of plot would you like?”. The format of these products being ASCII tab-delimited text made them essentially generic platform-independent products that could be easily distributed to other team members.

Commercial off-the-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 SAVISS. This relieved the users of much of the 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 the 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 the same System Gain job which made GAIN.TBL14 would also have the linkid ’14’ in their filename. Thus, 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 the information necessary to associate image filenames with the various tests ('observations') run by the JGSI. Therefore, the arduous and error-prone task of typing filenames into procedures could be eliminated by using an interface between the catalog and the procedures. Instead of installing a catalog interface in each program which needed files, the new program GIMMI 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 each test.

For those programs needing multiple frames simultaneously, GIMMI'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 the 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 the output products. The 'linkid' value is incremented and stored in the file. Next, GIMMI is run to extract from the catalog the filenames for the observation being processed. These files are then processed as a group by a program which makes an output table.

The output product (usually one or more tab-delimited ASCII table), GIMMI's list of files and perhaps other files produced by the procedure are all given filenames using the 'linkid' to associate them. The new program SAVEISS then catalogs these files for future use 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).

```plaintext
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

GIMMI 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 one or more tables of results

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

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 the parameters and the current value of the 'linkid', followed by GIMMI extracting from the catalog the filenames for the observation being processed. These files are processed sequentially via a loop. Each loop processes the next file in GIMMI's list and creates 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.

```plaintext
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

GIMMI: queries catalog for certain Observation; outputs list of filenames to process
```

```plaintext
1.001
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
SAVEIS catalog's table's location
SAVEIS catalogs list output by GIMMI:
```

**Example of resulting table:**

<table>
<thead>
<tr>
<th>IMAGE_NUMBER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>0.1</td>
</tr>
<tr>
<td>123457</td>
<td>0.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>123460</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**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.

```plaintext
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

GIMMI: queries catalog for certain Observation; outputs list of filenames to process
```

```plaintext
1.001
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
SAVEIS catalog's table's location
SAVEIS catalogs list output by GIMMI:
```

1.001
Example of resulting table:

<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>IMG123456</th>
<th>IMG123457</th>
<th>...</th>
<th>IMG123460</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1000.</td>
<td>1000.</td>
<td>...</td>
<td>1000.</td>
</tr>
<tr>
<td>0.1</td>
<td>200.</td>
<td>150.</td>
<td>...</td>
<td>160.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.5</td>
<td>10.</td>
<td>9.</td>
<td>...</td>
<td>12.</td>
</tr>
</tbody>
</table>

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 of the raw and processed files enabled process automation and flexibility to make the effort responsive to the instrument development team.

9. ACKNOWLEDGEMENTS

The authors wish to thank Tom Livermore, Cindy Kahn, and Bill Harris for guiding the ISS 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 Haff put the system described herein through its paces while calibrating both the Engineering and the Flight Model versions of the ISS cameras.

10. REFERENCES


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