Imaging Services Flight Software on the Mars Exploration Rovers

Todd E. Litwin
Jet Propulsion Laboratory
4800 Oak Grove Blvd.
Pasadena, CA 91109
Mail Stop 107
Todd.E.Litwin@jpl.nasa.gov

Justin N. Maki
Jet Propulsion Laboratory
4800 Oak Grove Blvd.
Pasadena, CA 91109
Mail Stop 264-621
Justin.N.Maki@jpl.nasa.gov

Abstract – *The Imaging Services module of the Mars Exploration Rovers' on-board flight software is responsible for providing image data to the rest of the system. It acquires images from a suite of cameras, performs on-board image processing, labels the results with metadata, and delivers the final products to a diverse set of consumers, both on board and on the ground. The demands for flexibility and speed lead to a design involving multiple tasks and a large set of parameters controlling the acquisition of the images, the on-board processing, and the method of product delivery.*

Keywords: Robotics, Image Processing, Mars Rovers, Camera Models.

1 Introduction

NASA's Jet Propulsion Laboratory sent two spacecraft to Mars in 2003 as part of the Mars Exploration Rover project. The two robotic vehicles they carried reached the Red Planet in January 2004. Designed to spend approximately three months exploring the geology of the planet, they have continued to operate for over a year at the time of this writing.

Images sent back to Earth from the rovers are vital to the mission. They were used by the on-board software to help guide the final stages of entry, descent, and landing. They are used by engineers to determine the rovers' state of health and whether or not critical operations[1] were performed correctly. They are used by scientists to select targets worthy of investigation, and by operators and planners to evaluate the safety of proposed pathways. They are used by on-board software for obstacle detection[2] while driving, and to recover the motion of the vehicle[3] and its final orientation[4]. They form an important part of the science data. And they appear prominently in newspapers, television reports, and web pages.

The on-board Imaging Services software is responsible for the acquisition and delivery of all of these images. Its form was driven by a combination of hardware constraints, performance requirements, and end-user needs. The resulting software, written in C, took over two years to produce and comprised over 25,000 lines. In this paper we will review what drove the design, the end product itself, and other activities that were critical for success.

2 Camera Hardware and Electronics

The cameras and interfaces were designed and built at JPL [1]. Each system has 10 cameras, 1 on the lander and 9 on the rover:

- DIMES[5] imager, mounted on the lander, used during Entry, Descent, and Landing (EDL)
- Pancam pair, narrow field of view, mounted on the Pancam Mast Assembly[6] (PMA), used mainly for science
- Navcam pair, medium field of view, also mounted on the PMA, used for navigation and engineering
- Microscopic Imager (MI), mounted on Instrument Deployment Device (IDD, a robotic arm), used for science
- Front Hazcam pair, wide field of view, mounted on the front of the rover body, used for IDD placement and hazard detection while driving
- Rear Hazcam pair, a second pair of hazard cameras on the back

[1] Especially the results of landing, cable cutting, mast deployment, airbag retraction, standup, and egress operations performed during the first days on the planetary surface

[2] Using stereo vision

[3] Using visual servoing

[4] By finding the direction to the sun

[5] Descent Image Motion Estimation System

[6] A one-meter mast, holding four cameras and the miniTES, a thermal-emission spectrometer

While they have different lenses, the cameras are otherwise identical.[7] Full images contain 1024x1024 monochrome pixels of 12 bits each. The interface to the flight computer supports not only acquiring these full-sized images with programmable exposure times, but options for acquiring less than full images. There are controls to alter the analog and digital characteristics of the camera's performance, and to generate test images.

The cameras may be identical, but their uses are not. The Hazcams are the simplest, mounted at fixed locations on the rover body. The Pancams and Navcams are mounted on top of the PMA, and are pointed by controlling the mast's pan and tilt. The Pancams sit behind actuated filter wheels. The Microscopic Imager is on a turret at the end of the IDD, and is positioned through arm motion. Its lens has a translucent dust cover that can be opened and closed. The DIMES imager, mounted on the lander, was usable only before landing, and was later disconnected and left behind.

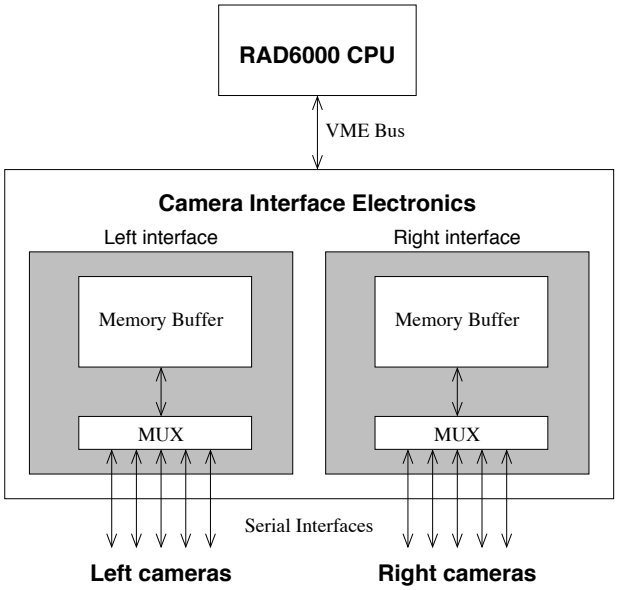

Figure 1: Camera Interfaces

The relevant camera interfaces are shown in Fig. 1. Half of the cameras are connected through a left interface, and half through a right one. At most, only a single left camera and a single right camera can be selected for use at a time. The interface supports taking either a single image from a selected camera or a synchronized left/right pair. While the interface is busy with one camera or pair, no other cameras may be used, or even powered on.

When a camera acquires an image, all of the image data is sent over a serial interface where it is stored in one of the memory buffers shown on the camera-interface electronics. The CPU, a 20MHz RAD6000 running the VxWorks real-time operating system, is then responsible for reading the image data across a VME bus into its own memory.

3 Software Architecture

3.1 Requirements

The fundamental requirement for the software was to acquire usable images through the interfaces to the camera hardware. Additionally the user community wanted a large number of exposure and processing features, which will be detailed in Section 4. There were other desires as well:

- Offer a variety processing options and data products both to reduce the demands for telemetry volume and to provide useful image data for on-board clients.[8]
- Avoid delays. Since many images would be taken, especially for panoramas, and since the interface electronics represents a bottleneck, make sure to keep the requests flowing through the hardware.
- As a corollary to the previous item, promptly start the next image acquisition even in the face of compute-intensive post-processing operations (such as image compression) for prior images
- Support requests from multiple clients at all times.
- Schedule the more time-critical requests before those that are less time-critical.
- As a special case of the above, be capable of delivering descent images rapidly enough to support a safe landing.
- Command the requested movements of the PMA and Pancam filter wheels.
- Coordinate with other subsystems for the PMA, which is a shared resource.
- Coordinate ground and on-board control of imaging.

3.2 Design

The resulting design contained three tasks, as shown in Fig. 2. The Imaging Services task receives all imaging requests, and is responsible for the overall scheduling of image acquisition and coordination with other subsystems. The Camera Control and Post Processing tasks deal with the actual image production, and will be discussed in detail in Section 4.

[7] The sole exception is a detail in the electronics of the DIMES imager affecting exposure gain; this is invisible to the software.

[8] Throughout this paper, the term "client" will be used to refer to other on-board software modules that request images.

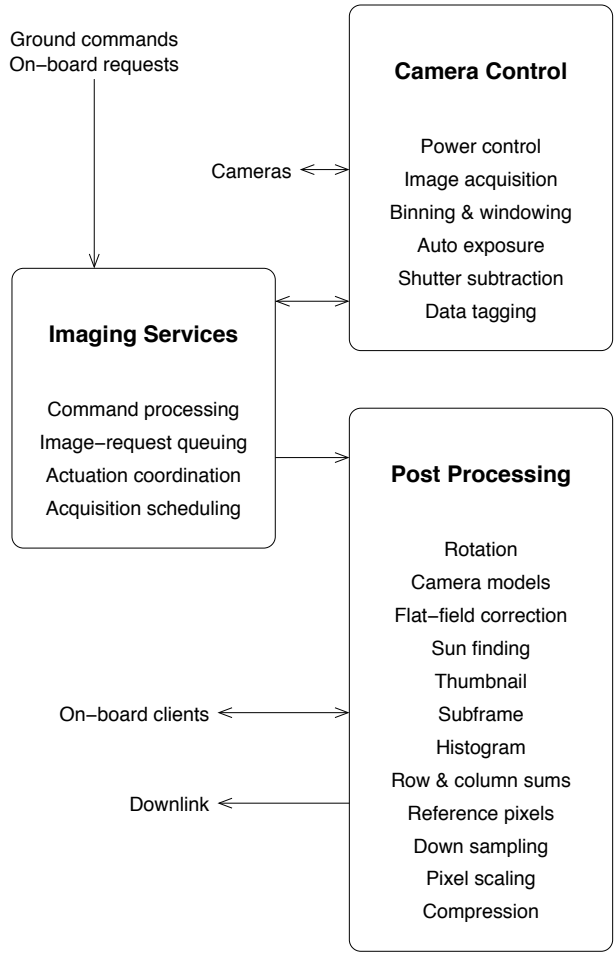

Figure 2: Software Tasks

Image requests are held in the Imaging Services task until ready for processing. They then pass to the Camera Control task for the acquisition of the image data. After that they pass back through the Imaging Services task to the Post Processing task for all time-consuming image manipulations, product production, and dispatching into telemetry.

3.3 Resources

The system handles multiple image requests in parallel, allowing each to be in a different stage of processing. All clients share a pool of image buffers in the CPU's memory. These buffers are used to hold the data required for individual requests. Request parameters, image data, and other information are all stored there together. In effect, a buffer *is* an image request, or rather, the current state of that request. As image requests are serviced, buffers as passed through the system. Buffers are allocated when needed. In special cases they are allocated earlier and held to guarantee availability.

It is important to be aware of the existence of these image buffers since they are a limited resource. If imaging commands from the ground arrive when no more buffers are available, those commands wait in a queue for earlier requests to complete. All ground imaging commands pass through this queue, maintained by the Imaging Services task, before they are turned into normal requests as described below. Ground planners must be aware of this issue when planning operations.

3.4 Making a request

When a client prepares to make a request, it starts by setting request parameters to default values and placing them into the image buffer. There are different default parameter sets for different applications. These values are controllable by ground command. The client has the option to override these values autonomously if needed before forwarding the buffer to the Imaging Services task.

3.5 Scheduling and Coordination

Each request is accompanied by a priority related to the time criticality of fulfilling it. When the Imaging Services task receives a request, it is placed into a queue appropriate to that priority. Higher-priority requests are serviced before lower-priority ones.

There are two queues for each priority level: one for PMA-mounted cameras, and one for non-PMA cameras. Using PMA-mounted cameras requires extra time. The right to use the PMA must be negotiated with other subsystems. The PMA may need to be repointed. The Pancam filter wheel may need to be rotated.

If a non-PMA request arrives before these extra duties are complete, then the new request will be serviced first. As long as the cameras are ready for a new request and some request is ready to go, that request will be sent to the cameras. This means that more images can be acquired in less time than would be possible if requests were serviced in the order of arrival. But it also means that it is not possible to predict with certainty the order in which images for different cameras will be taken.

Acquiring images, discussed more in the next section, can take many seconds. Once that is complete, the request is sent to the Post Processing task for all lengthy follow-on operations, which frees up the Camera Control task to take more images. Imaging Services' job is to keep the Camera Control task as busy as possible.

4 Image Processing

4.1 Camera Control

A number of activities are performed in the process of acquiring images.

Power. Cameras are usually powered off. Before taking an image, the required cameras are powered on. In anticipation of repeated use of the same cameras, the power is left on for a while. It is automatically turned off to conserve energy if left unused for a programmable

period of time, or immediately if other cameras are selected for use.

Vehicle State. In parallel with acquisition, certain information is collected and recorded in the image buffer. This information includes time, rover position and orientation, temperatures, and the states of the vehicle suspension, Pancam filter wheels, MI cover, PMA, IDD, and High-Gain Antenna (HGA). This collection has proven so useful and unique in the telemetry that non-imaging ground users often refer to it.

Manual Exposure. Exposure time is commanded in 5.1ms units. The duration can be specified either as an exact number, or as a scaled value relative to the most recently used and saved exposure time for any camera.

Auto Exposure. Commanding automatic exposure results in multiple images being acquired, ending when a desired profile of intensities is obtained or a repeat limit is reached. If subframing is being used (see next section), then the intensity analysis is restricted to the subframe. The resulting exposure time is remembered for later use by the scaled manual exposure described in the previous paragraph.

Hardware Binning. The camera can be commanded to combine the pixel data from sets of 4 rows to create lower-resolution images that contain one-fourth the usual number of rows. Corresponding pixels in each of the four rows are summed together. To prevent saturation, the exposure time needs to be reduced to one-fourth of normal, an adjustment that the software makes automatically.

Hardware Windowing. When not using hardware binning, the camera can be commanded to send only a subset of the 1024 image rows. During testing it was found that using this feature caused image corruption inside the camera. So hardware windowing is never used.

Shutter Subtraction. After the desired image is acquired, a step can be commanded to reduce the effects of dark current.[9] An additional, zero-time exposure is made, the results of which are subtracted pixel-by-pixel from the image.

Pixel Scaling. While the native pixel data is represented as a 12-bit integer after digital sampling, the interface electronics can be commanded to read in only the high-order 8 bits. A more sophisticated version of this pixel-scaling operation is available in the post-processing steps described in the next section.

4.2 Post Processing

The balance of the processing consists of several steps:

- Early processing
- Early delivery to client
- Late processing & telemetry generation
- Late delivery to client

4.2.1 Early processing

Rotation. Many of the cameras are mounted on their sides or upside down. To make the view of all images consistent for all clients and reduce confusion to humans, the image is rotated as needed to make it appear upright before being made available for use by any client.

Camera Model Labelling. The image is then tagged with a geometric model; see Section 5.

Bad-Pixel Correction. Sometimes, especially in radiation-rich environments, CCDs fail such that single pixels or entire columns are corrupted. Bad pixels usually have values held artificially high or low. Once bad, these values tend to stay bad and can be detected on the ground. Correction is performed by replacing each bad pixel with an average of some its neighbors. A variety of neighbor patterns is available.

Flat-Field Correction. A function-based intensity correction can be applied to counteract lens vignetting effects.

Sun Finding. An algorithm [2] can be applied to the image to locate the sun's 2D image location. The camera model is used to compute the 3D vector to the sun from the 2D location. This vector is used by the on-board attitude-control software when determining the orientation of the rover on the planet's surface, and contributes to accurate pointing of the HGA for communications with Earth.

4.2.2 Early delivery to client

At this point it is possible to exit from post processing and return the image data to an on-board client. This is the earliest point that an image is available to any external user. If this option is taken, it is up to the client to decide if and when the rest of the post processing should occur. This feature was used during EDL when rapid image acquisition and DIMES processing was essential; the bulk of post processing was performed later, after first impact while the rover was still bouncing on the planet's surface.

4.2.3 Late processing

A number of separate data products are produced: thumbnails, row sums, column sums, histograms, normal images. Any subset of these may be selected for creation and transmission. Choosing the thumbnail, sum, or histogram products rather than full images reduces telemetry volume.

Thumbnail generation. Before any of the remaining image filtering is performed, a reduced-resolution version of the full image can be created. Such thumbnail images take up less telemetry volume and are given higher transmission priorities than normal image data.

[9] The accumulation of charge in the CCDs not caused by incident light, but by electrons in the valence band acquiring enough thermal energy to jump into the conduction band. The effect increases with temperature.

Thumbnails provide operators on the ground with early views of on-board images and their header data.

Subframing. All of the pixels outside a chosen rectangular subframe can be discarded, leaving a smaller image. While normally selected manually, a subframe can also be constructed automatically around where the sun is detected.

Histogramming. A histogram can be created containing the counts of all possible pixel values found in the image.

Row & Column Summing. The summations of the pixel values for all of the rows or all of the columns can be computed, which is useful for assessing the spatial distribution of intensities without the need to transmit a full image.

Reference Pixels. When an image is acquired, part of the data from the camera comes from outside the 1024x1024 illuminated region. These reference pixels, 31 per row, provide a measure of the noise and digitizing behavior of the camera, and can be packaged into a separate data product for transmission.

Spatial Downsampling. In addition to any hardware binning, the software may be commanded to downsample the image's resolution. Available options for such downsampling include mean, mean after rejecting one outlier, and median.

Pixel Scaling. The image's 12-bit pixels can be converted to 8-bit. There are options to select any specified contiguous 8 bits out of the 12, and one to select the 8 bits automatically based on the maximum value present. There are also on-board lookup tables programmable by ground command. Scaling can be applied to regular images and to thumbnails. When the early-return option described above is exercised, it is possible to request pixel scaling at that time rather than here.

Compression. Images can be compressed with a sophisticated algorithm [3] that is either lossy or lossless, or with a faster lossless algorithm [4]. Compression is used for virtually every image [5]. This includes the main image as well as the thumbnails and reference pixels. Compression is time-consuming. If scheduling is a problem, it may be delayed to a later time, at which point explicit ground commands will be necessary to invoke it.

4.2.4 Telemetry Generation

As they are produced, the generated products are queued separately for transmission within the telemetry stream at requested priorities. Command parameters and the state data sampled earlier are packaged along with each product.

4.2.5 Late delivery to client

The image in its final form is made available to the on-board client. This is the end of processing for an image request.

5 Geometric Camera Models and Calibration

All images are labelled with geometric camera models shortly after acquisition. They describe the mapping from pixel space to the rover's body-fixed coordinate system shared by all subsystems. The models are transformed as needed to account for any applicable platform motion by the PMA or IDD. They are also adjusted to reflect any subframing or spatial downsampling.

Two model types are used. Most of the cameras use a linear perspective-projection model that includes radial distortion terms [6]. The Hazcams, which have fisheye lenses, use a more general model [7]. The software also supports a much simpler model with linear perspective-projection but no radial distortion [8], [9], used only to label idealized models generated from the on-board stereo-vision software.

The advantage of consistent labeling and transforming for pose is that the users do not need to check elsewhere for geometrical image context. All information needed to understand the image's viewpoint in rover coordinates is contained within the image. Since the image header also includes the rover's position and orientation on the planet's surface, it is even possible to understand the image in the larger context.

The models were calibrated by taking images of visual targets that were surveyed in rover coordinates. The model parameters were determined with a least-squares fit described in the references. Over the course of testing and preparation for flight, over 80 such models were created for the flight and test cameras.

6 Testing

6.1 Unit Testing

One byproduct of the architecture described above is the independence of the client from the producer. The Imaging Services module neither knows nor cares about the identity of the client. Any client can exercise any service. This makes it easy to design special on-board unit-test software to allow the developer to exercise the system. Two such pieces of software were written. They were used only on the ground and were not launched with the spacecraft.

The first was informal software for experimenting with the cameras early in development. Since networking with the flight computer was available in the testbeds, an on-board HTTP server was written. A browser addressing the system would see a web page containing text boxes and radio buttons that could be used to define a request. When the "submit" button was pressed, the server would form a request, send it to Imaging Services, wait for the resulting image, and send it back to the web browser for display. This tool was a great help in getting the details of interface control right, as well as in providing early feedback on the

performance of the cameras. A spartan version was in use within days after development began, long before a working ground system would enable viewing image data in telemetry. This web interface continued to be of service for a long time thereafter, evolving as the implementation progressed.

More comprehensive unit-test software was written later using the same basic technique. This software, as an on-board client, exercised one functionality after another, testing timing and validating results, all in a fully automatic test suite that took about an hour to run. When features were added, the unit test was upgraded to check them. When defects were discovered, it was upgraded to duplicate them, and later used to validate that they had been corrected. When the flight software was changed, it was run as a regression suite to check for the introduction of new defects. While the unit test did not cover all possible errors, it helped keep the number of problems discovered in system testing and operations to a minimum.

6.2 System Testing

Imaging was formally tested as part of the larger system tests designed to validate the performance of the entire integrated vehicle system. Detailed commands were designed to exercise all of the system's degrees of freedom, including those of Imaging Services. Since it was not possible to test all combinations of those degrees of freedom, several multi-day tests were also run to rehearse likely operations scenarios.

7 Conclusions

The Imaging Services system is large and complex. It requires over 70 parameters to specify an image acquisition, making it tricky to use. But the users demanded the power that implies such complexity. Every degree of freedom has been used in operations, most of them heavily.

The Imaging Services system has produced a large volume of data. By early 2005, over 70,000 images had been returned to Earth, examples of which are shown in Fig. 3 and Fig. 4. Those images have been used for everything from navigating to making scientific discoveries to communicating with the public.

Possibly the most surprising use has been for engineering. On most prior missions, sensor readings alone were used to verify operations. On MER, even where such data were present, the engineering staff was not satisfied with reaching conclusions before images were examined. Many such engineering images are taken.

None of this would have been possible without the current architecture, supporting high throughput and fine-grained control. Imaging on the Mars Exploration Rover project will set a standard for future missions.

Figure 3: Spirit looking out to Columbia Hills

Figure 4: Opportunity's first view after landing

Acknowledgments

The authors would like to recognize the following for their significant contributions:

- The camera team, especially Dave Brown, Arsham Dingizian, Tom Elliott, Ed Hagerott, Paul Karlmann, Larry Scherr, David Thiessen, Mark Wadsworth, Len Wayne, and Mary White
- Alan Lee and John Waters for the camera-interface electronics
- Aaron Kiely and Matt Klimesh for the design and implementation of the compression algorithms, and Bruce Bon for their integration into the flight software
- Carl Christian Liebe for the implementation of the sun-finding algorithm
- Reg Willson for the especially painstaking calibration of the DIMES imagers
- Ed Motts, Eugene Poyorena, and Mark Thompson for the surveying measurements used in camera calibration
- Earlier work with the Mars Pathfinder cameras [10], experience from which flowed into the current effort in many ways

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

References

[1] J. N. Maki, J. F. Bell, K. E. Herkenhoff, S. W. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, M. Maimone, E. Baumgartner, A. Collins, M. Wadsworth, S. T. Elliot, A. Dingizian, D. Brown, E. C. Hagerott, L. Scherr, R. Deen, D. Alexander, J. Lorre, "The Mars Exploration Rover Engineering Cameras," Journal of Geophysical Research — Planets, Special Issue: The Mars Exploration Rover Mission, 2003.

[2] Allan R. Eisenman, Carl Christian Liebe, Ramiro Perez, "Sun Sensing on the Mars Exploration Rovers," Proceedings of the 2002 IEEE Aerospace Conference, Vol 5, pp. 5-2249 – 5-2262, March 9-16, 2002.

[3] A. Kiely, M. Klimesh, "The ICER Progressive Wavelet Image Compressor," IPN Progress Report 42-155, pp. 1-46, July-September, 2003.

[4] M. Klimesh, V. Stanton, and D. Watola, "Hardware Implementation of a Lossless Image Compression Algorithm Using a Field Programmable Gate Array," TMO Progress Report 42-144, October-December 2000, pp. 1-11, February 15, 2001.

[5] A. Kiely, M. Klimesh, "Preliminary Image Compression Results from the Mars Exploration Rovers," IPN Progress Report, vol. 42-156, February 15, 2004, pp. 1-8.

[6] D. Gennery, "Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points," in *Calibration and Orientation of Cameras in Computer Vision*, A. Grun and T. Huang, editors, Springer-Verlag, ISBN 3–540–65283–3, 2001.

[7] D. B. Gennery, "Generalized Camera Calibration Including Fish-Eye Lenses," JPL document clearance number 03–0869. Submitted to the International Journal of Computer Vision, 2005.

[8] Y. Yakimovsky and R. T. Cunningham, "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras," Computer Graphics and Image Processing 7, pp. 195–210, 1978.

[9] D. B. Gennery, T. E. Litwin, B. H. Wilcox, and B. B. Bon, "Sensing and Perception Research at JPL," Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, pp. 311–317, March 31 – April 3, 1987.

[10] J. Crisp, T. E. Litwin, J. J. Lorre, D. A. Alexander, A. J. Runkle, T. J. Parker, and H. J. Moore, "Pathfinder rover cameras: resolution, stereometry, and geology," Abstract in Eos Trans. AGU, 78(46), Suppl: F403, Fall 1997.