CHALLENGES UTILIZING PUSHFRAME CAMERA IMAGES. J.A. Anderson[1] and M.S. Robinson[2], [1]U.S. Geological Survey, (2255 N Gemini Dr, Flagstaff, Arizona 86001, janderson@usgs.gov), [2]Arizona State University (School of Earth and Space Exploration, Box 871404, Tempe, Arizona 85287-1404, robinson@ser.asu.edu).

Introduction: Recent NASA missions have utilized pushframe cameras, a relatively new type of multispectral imaging system (Mars Odyssey Thermal Emission Imaging System (THEMIS) Visible (VIS) camera, Mars Reconnaissance Orbiter Mars Color Imager (MARCI) [1,2]), to successfully collect digital images of Mars. This class of camera is a hybrid between a frame camera and a pushbroom camera. A frame camera collects an entire image at a single instant in time, while a pushbroom camera collects one line of image data over a time range and builds a two-dimensional image through spacecraft motion. Pushframe cameras collect a small multiple of lines in one integration, called a framelet, and build up a complete image through downtrack motion of the spacecraft, similar to a pushbroom camera. Figure 1 is an example of a THEMIS VIS image with three framelets.

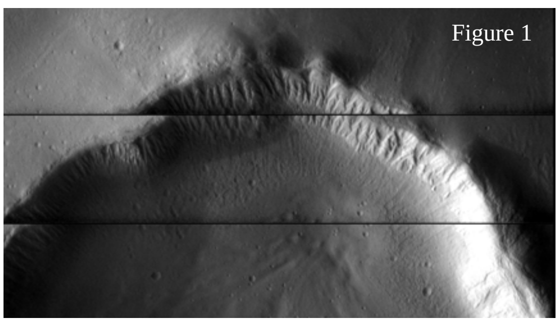


Figure 1

The THEMIS VIS and MARCI cameras collect 192 and 16 line framelets, respectively, and then record hundreds to thousands of framelets to build a large two-dimensional image. Additionally, the pushframe cameras collect multispectral data; not with a filter wheel, but using a novel solution by configuring multiple filters on top of a standard frame camera CCD and recording only the lines covered by the filters. MARCI and THEMIS VIS have seven and five bands, respectively. In this abstract, we discuss some of the challenges using data sets collected by pushframe cameras.

Image Visualization: Archival images are provided by mission teams to the Planetary Data System (PDS) and can be visualized using common software tools such as the Integrated Software for Imagers and Spectrometers (ISIS) [3,4] or NASAView. Figure 2 is a subset of a raw PDS Engineering Data Record (EDR) for the MARCI camera. The subset image contains thirty framelets and five bands. There is difficulty visualizing the image content (Mars South Pole) for several reasons: 1) the five multi-spectral framelets are all stored in a single band which causes the horizontal stripes and 2) features break apart between framelets in ascending orbits.

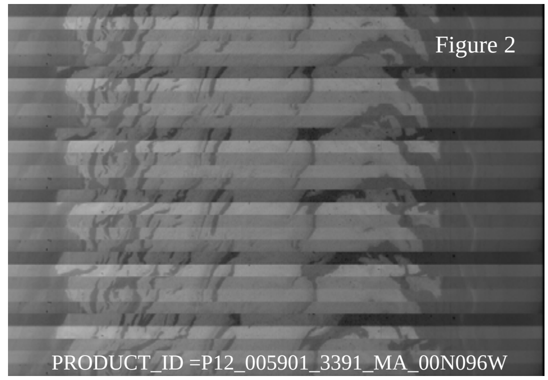


Figure 2

The latter is illustrated in Figure 3, which has been reduced to show only a single band (IR filter). The crater on the left appears in four separate framelets seemingly out-of-order. Because framelets are stored time sequentially in the image file and were collected in an ascending orbit, the southern portion of the crater is imaged and stored first, and as the spacecraft moves north, more of the crater is observed in each successive framelet.

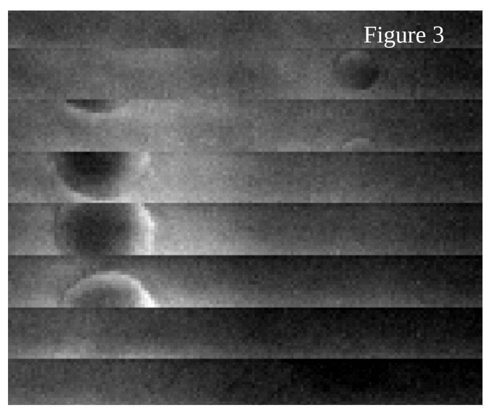


Figure 3

Neither frame nor pushbroom cameras exhibit the visualization difficulties of raw products like pushframe cameras. Special software must be written to break apart and reorganize the framelets just for simple viewing. Commercial-off-the-shelf (COTS) and open source software typically lack support for this esoteric file structure. Therefore, basic image selection and analysis becomes a much more difficult task when compared against traditional cameras.

Cartographic Processing: Rendering cartographic products from one or more EDR images is necessary for scientific analysis and mission planning [5]. The process of creating a map projected image from a raw EDR is a geometric transformation of pixels akin to an image rotation (albeit slightly more complicated). The geometric transform is facilitated at each output pixel (samp,line) in the map projected image as follows:

(latitude,longitude) = *MapProjection*(samp,line)
(tsamp,tline) = *SensorModel*(latitude,longitude)
OutputPixel(samp,line) = InputPixel(tsamp,tline)

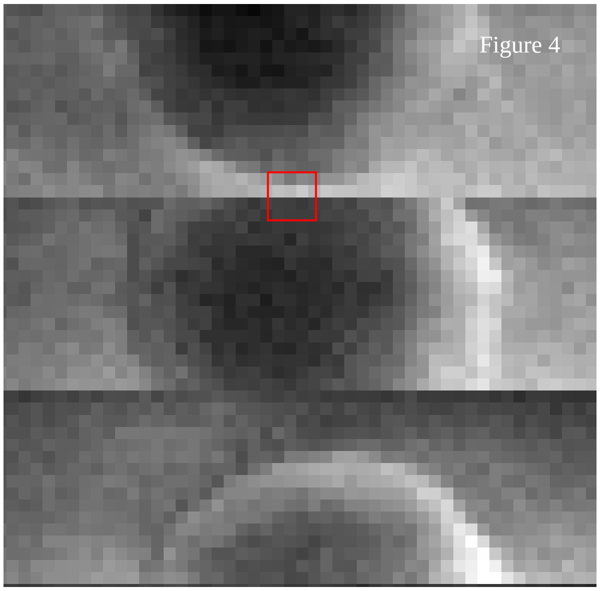
MapProjection is the equation for a specifica map projection such as sinusoidal or polar stereographic and yields a unique ground coordinate for each pixel position in the output image.

SensorModel is the standard photogrammetric collinearity equations for the sensor model which maps a ground coordinate into the focal plane of the camera and hence to (tsamp,tline) in the EDR.

Pushframe sensor models pose significant problems when compared to frame and pushbroom cameras, both of which have a one-to-one relationship between pixel location and ground position. That is, for each (sample,line) coordinate in the image there is exactly one (latitude,longitude) coordinate. Revisiting figure 1, it is apparent that for pushframe images, ground coordinates map to two pixel coordinates in many instances because of overlap at the bottom of framelet 1 and top of framelet 2.

Therefore, in the geometric transformation, there are two locations for InputPixel(tsamp,tline). It could be argued it doesn't matter, just choose one of the two pixels, but this strategy results in nearest neighbor interpolation leaving a noticeable artifact. Many geometric transformation programs provide other interpolators such as bilinear and cubic convolution (weighted averages). Both algorithms require a small window of pixels about the nearest neighbor pixel, 2x2 and 4x4, respectively. Figure 4 shows the problems encountered by a cubic convolution interpolator used on a MARCI push frame EDR at the framelet boundary. The red box delineates a 4x4 window of pixels used for the interpolation. The discontinuity at framelet boundaries will cause erroneous results in the interpolation.

Therefore, cartographic processing of pushframe images requires that each framelet be extracted into a separate file (or memory buffer), which can be individually map projected, and then joined using mosaicking software tools. This strategy was implemented in ISIS2 for the THEMIS VIS camera in 2002. The number of framelets (and consequently files) typically numbered less than hundred for an EDR. However, MARCI images can have in excess of ten thousand framelets and thus produce an equivalent number of files resulting in increased processing times and file management headaches.



These file management problems can be reduced by creating two files. One file would contain data only in the odd framelets and the even framelets would be replaced with null padding. The second file would similarly contain even framelets with null padding. The bilinear and cubic convolution interpolators used in ISIS geometric processing understand how to interpret the null pixels such that the images can be effectively map projected and then mosaicked without significant changes to the existing software

Conclusion: There is always a trade-off for building and launching inexpensive, light weight cameras. Over the past forty years, hundreds of man-years have gone into the development of COTS and open source photogrammetric and cartographic software for processing pushbroom and frame camera planetary images. This work must be revisited to address unique problems encountered with pushframe cameras. In the mean time, these valuable data sets are under utilized by the scientific community because of data processing complexities and the lack of available software. The U.S. Geological Survey and Arizona State University are actively pursing these issues, using MARCI images as a test bed, in preparation for data collected by the Lunar Reconnaissance Orbiter Wide Angle Camera.

References: [1] Christensen P. R. et al. (2003) *Science, 2003,* 2056-2061. [2] Malin, M.C. et al. (2001) *JGR, 106,* p. 17651. [3] Torson and Becker, (1997) *LPS XXVIII,* pp. 1443-1444 [4] Gaddis, et al., (1997) *LPS XXVIII,* pp. 387-388. [5] Eliason, E., Production of Digital Image Maps with ISIS (1997) *LPS XXVIII,* pp. 387-388.