

Data Introduction

Deep Impact: Medium-Resolution Instrument (MRI) Data

Instrument Operations

The MRI is a bilateral frame transfer CCD with a 1024^2 -pixel image area at the focus of a 12-cm aperture Cassegrain telescope, and behind a nine-position filter wheel. The instrument took scientific data throughout Deep Impact's operational lifetime. The image is rapidly transferred to two storage regions on opposite sides and is then read out in four separate quadrants via four separate readout amplifiers. Thus quadrant boundaries are noticeable in the raw data and are not always completely eliminated in the calibrated data. In addition, the center lines of the image (511 and 512 for 0-based indexing of a full image) have $5/6$ the nominal pixel size, due to the fact that both sides of the frame transfer share one of the three clock cycles used to transfer the image clock. The data are calibrated to units of surface brightness, thus when computing total incident flux or when measuring distances across the center line, the smaller size of the center pixels should be considered. The instrument can be read out in sub-frame modes down to 64^2 , always centered on the chip. The data may be compressed on the spacecraft from 14-bit to 8-bit values through a look-up table, and decompressed by the calibration pipeline. Most of the science data from the Deep Impact prime mission were compressed. In the larger format readouts, overclock pixels replace columns and rows at the edges of the image. In addition, the first 128 bytes in one quadrant are overwritten by header information.

Reading the Data

The data are all stored as FITS files with detached PDS labels, one label and one FITS file for each exposure. Filenames are chronologically ordered. The first part of the filename is "mv" (for MRI Vis) followed by the truncated UTC date (YYMMDDHH). The second part of the filename is the sequence number (aka exposure ID) of the image as used in our documentation and in spreadsheets that list the files. The third numerical group is the image number of the given sequence. For calibrated data, this is followed by a 1- or 2-letter code to indicate type of product – "r" for radiance or "rr" for reversible radiance (I/F images are not provided in this version but we provide the multiplicative constant to convert from radiance to I/F). Note that the overclock and image header pixels may include raw or unphysical values that could affect the calculated minimum and maximum data values. Take care when displaying images with programs that automatically scale to the data minimum and maximum.

Each raw image consists of two data objects. The first data object is the image as an array of 1- or 2-byte integers, while the second (aka FITS extension 1) is a quality factor image (1-byte integers) flagging bad pixels. Each calibrated image product consists of four data objects. The primary image itself (floating point) is in units of radiance, and is followed by an array of quality factors (1-byte integers), an array of signal-to-noise ratio (floating point), and a small array (2 columns) of the applied stripe removal (mitigating an electrical interference pattern). The pixels in the quality factor and signal-to-noise ratio arrays have a one-to-one correspondence to every pixel in the primary image (the

columns in the stripe removal array are each applied to every column in half of the image).

If your favorite analysis environment is IDL, you can use the package `readpds.pro`, which is available at PDS-SBN (<http://pdssbn.astro.umd.edu/tools/>). If you type `data = readpds(<filename.lbl>)`, it will read the data, including all extensions, into an IDL structure containing all the parts of the data product. To see the various pieces, type `help, /struct, data` and it will list the pieces of the highest level of the structure. Some of those will themselves be structures and you can type, `help, /struct, data.piece1` to find out what is in the sub-structure `piece1`. If your favorite environment is ISIS, there is a routine `pds2isis`, although we have not exercised this routine.

PDS does not explicitly support FITS, but if your favorite analysis environment is based exclusively on FITS, you can read the FITS file directly (with extension `.fit`), ignoring the PDS label (the file with `.lbl` extension), but you need to be aware that PDS does not validate or officially support the FITS standard. We strongly encourage the use of the freely available routine `fv` (<http://heasarc.gsfc.nasa.gov/fv/>) to read the entire file and determine which extensions are of interest.