JPEG 2000 as a PDS Archive Format: Supplementary Material

Working Group: Sue LaVoie, Elizabeth Rye, Brad Castalia, Patty Garcia, Steve Hughes, Chris Isbell, Ron Joyner, Myche McAuley, and Betty Sword

Sept. 28, 2005

What is JPEG 2000?

- a compression scheme, and

- an image format

- A raw image, when compressed using the JPEG 2000 compression algorithm, becomes a JPEG 2000 codestream (mime type: J2C)

- The JPEG 2000 codestream, when encapsulated in the JPEG 2000 binary wrapper, becomes a JP2 formatted image (mime type: JP2)

What makes JPEG 2000 different from other compression algorithms?

- The structure of the codestream is highly flexible. This enables selective decompression of:
  - different resolution "layers"
  - images of varying precision
  - portions (or "tiles") of the image
  - targeted regions of interest (ROIs)

Why should we care about the JP2 format?

- The J2C codestream is self-contained and contains sufficient information for a conforming software program to fully decompress the image data. However...

- Without the JP2 header, a software program can't take advantage of the additional capabilities of the JPEG 2000 format.

What impact does this have on the use of JPEG 2000 in PDS archives?

- The working group is proposing that we permit data providers to provide data files in either J2C or JP2 format.

What precedents exist in the PDS for handling compressed data and data in non-PDS formats?

- The precedents for compressed data can be broken down into two main categories:

  – data files containing distinct objects, some or all of which consist of compressed data

  – data files where the entire file is a single, compressed entity

- PDS archived, compressed data sets which consist of data files containing distinct objects are:
  - the Clementine image EDRs
  - the MGS MOC SDPs (essentially EDRs)
  - the Viking Orbiter image EDRs
  - the Voyager image EDRs

- The data products in each of these data sets have attached PDS labels which contain information describing both the compressed and the decompressed files within a single, implicit FILE object.

- In each case, the decompression software included in the archive produces a decompressed image file with an attached PDS label modified from the original label to describe only the decompressed file.

- An attached PDS Viking label for a compressed product:

```
CCSD3ZF0000100000001NJPL3IF0PDS200000001 = SFDU_LABEL
/*  FILE FORMAT AND LENGTH */
RECORD_TYPE                         = VARIABLE_LENGTH
RECORD_BYTES                        = 1204
FILE_RECORDS                        = 2176
LABEL_RECORDS                       = 60
/*  POINTERS TO START RECORDS OF MAJOR OBJECTS IN FILE */
^IMAGE_HISTOGRAM                    = 61
^ENCODING_HISTOGRAM                 = 62
^ENGINEERING_TABLE                  = 64
^LINE_HEADER_TABLE                  = 65
^IMAGE                              = 1121
/*  IMAGE DESCRIPTION */
...
OBJECT                              = IMAGE
 ENCODING_TYPE                      = HUFFMAN_FIRST_DIFFERENCE
 LINES                              = 1056
 LINE_SAMPLES                       = 1204
 SAMPLE_TYPE                        = UNSIGNED_INTEGER
 SAMPLE_BITS                        = 8
 SAMPLE_BIT_MASK                    = 2#11111110#
 CHECKSUM                           = 18081344
END_OBJECT
END
```

- An attached PDS Viking label for a decompressed product:

```
CCSD3ZF0000100000001NJPL3IF0PDS200000001 = SFDU_LABEL
/*  FILE FORMAT AND LENGTH */
RECORD_TYPE                              = FIXED_LENGTH
RECORD_BYTES                             = 1204
FILE_RECORDS                             = 1115
LABEL_RECORDS                            = 2
/*  POINTERS TO START RECORDS OF MAJOR OBJECTS IN FILE */
^IMAGE_HISTOGRAM                         = 3
^ENGINEERING_TABLE                       = 4
^LINE_HEADER_TABLE                       = 5
^IMAGE                                   = 60
/*  IMAGE DESCRIPTION */
...
OBJECT                                   = IMAGE
 LINES                                   = 1056
 LINE_SAMPLES                            = 1204
 SAMPLE_TYPE                             = UNSIGNED_INTEGER
 SAMPLE_BITS                             = 8
 SAMPLE_BIT_MASK                         = 2#11111110#
END_OBJECT
END
```

- PDS archived, compressed data sets which consist of data files treated as a single, compressed entity have, to date, exclusively used Zip compression. They are:
  - the MPF Rover Engineering data
  - the Cassini Radar LBDR data
  - the MER mobility reports
  - the HST Saturn Ring Plane Crossing supplementary data files

- An combined detached Cassini LBDR label:

```
PDS_VERSION_ID                 = PDS3

/*        PRODUCT DESCRIPTION */

DATA_SET_ID                    = "CO-V/E/J/S-RADAR-3-LBDR-V1.0"
DATA_SET_NAME                  = "CASSINI RADAR LONG BURST DATA RECORD"
...
OBJECT                         = COMPRESSED_FILE
  FILE_NAME                    = "LBDR_02_003_V01.ZIP"
  RECORD_TYPE                  = UNDEFINED
  ENCODING_TYPE                = ZIP
  INTERCHANGE_FORMAT           = BINARY
  UNCOMPRESSED_FILE_NAME       = "LBDR_02_003_V01.TAB"
  REQUIRED_STORAGE_BYTES       = 80994528
  ^DESCRIPTION                 = "SOFTWARE/SOFTINFO.TXT"
END_OBJECT                     = COMPRESSED_FILE

OBJECT                         = UNCOMPRESSED_FILE
  FILE_NAME                    = "LBDR_02_003_V01.TAB"
  RECORD_TYPE                  = FIXED_LENGTH
  RECORD_BYTES                 = 132344
  FILE_RECORDS                 = 612
  LABEL_RECORDS                = 1

/*        POINTERS TO START RECORDS OF OBJECTS IN FILE */

 ^LBDR_TABLE                   = ("LBDR_02_003_V01.TAB", 2)

...
```

- Non-PDS formats for data files which have been included in PDS archives are:
  - FITS images
  - ISIS images
  - ISIS qubes
  - VICAR images

- Data sets which include these formats have provided a combination of attached and detached labels to provide both the PDS and native format labeling information. Typically, the PDS label has been the detached label.

What would a JPEG 2000 product label look like?

- Because JP2 formatted files can contain intermingled header and image data, they are best considered as a single, compressed entity, rather than as a set of independently compressed objects. Thus, the combined detached labeling approach most accurately describes the file.

```
PDS_VERSION_ID              = PDS3

(identification and descriptive data elements)

OBJECT                      = COMPRESSED_FILE
  FILE_NAME                 = "filename.jp2"
  RECORD_TYPE               = UNDEFINED
  ENCODING_TYPE             = "JP2"
  ENCODING_TYPE_VERSION_ID  = "ISO/IEC15444-1:2004"
  INTERCHANGE_FORMAT        = BINARY
  UNCOMPRESSED_FILE_NAME    = "filename.img"
  REQUIRED_STORAGE_BYTES    = nnnn
  ^DESCRIPTION              = "jp2_description.txt"
END_OBJECT                  = COMPRESSED_FILE

OBJECT                      = UNCOMPRESSED_FILE
  FILE_NAME                 = "filename.img"
  RECORD_TYPE               = FIXED_LENGTH
  RECORD_BYTES              = nnn
  FILE_RECORDS              = nnn

/* POINTER TO DATA OBJECT */

  ^IMAGE                    = "filename.img"

/* DATA OBJECT DEFINITION */

  OBJECT                    = IMAGE
    LINES                   = nnn
    LINE_SAMPLES            = nnn
    (etc.)
  END_OBJECT                = IMAGE
END_OBJECT                  = UNCOMPRESSED_FILE
END
```

What other issues are there surrounding the JPEG 2000 compression format?

- The JPEG 2000 specification permits both lossless and lossy compression.

- We propose that the PDS should limit the use of this compression algorithm to lossless compression.

- Data conforming to part 1 of the JPEG 2000 specification are available on a "royalty and license fee free" basis.

- Extensions to the format providing additional capabilities are described in other parts of the specification. These may require the payment of royalties or licensing fees.

- We propose that JPEG 2000 formatted data in the PDS be limited to the syntax and features defined in part 1 of the specification.

What impact would acceptance of JPEG 2000 have on PDS tools?

- Reference decompression software (including source code) is freely available in C and Java

- The Object Access Library would need to be updated to include these decompression algorithms.