PURPOSE OF CODE

HDFDUMP is a utility program to read in any gridded MISR data (such as L1B2 Ellipsoid and Terrain Data, Geometric Parameters, Level2 Aerosol/Surface and Top-of-Atmosphere/Cloud) that is written in the HDF-EOS grid format. It reads in the data from the MISR file and translates them into raw binary data which is then written out to disk.

A single file contains an entire orbit's worth of MISR data, subsetted into "blocks" which are 140.8 km long and 563.2 km wide. The daytime length of the orbit is approximately 143 blocks.

The data is stored in terms of grids (akin to sub-directories in a filesystem) and fields (like the actual files in those subdirectories). The list of grids and fields in a file will be displayed when you run HDFDUMP and you will be asked to choose the grid and field number corresponding to the data that you want to access.

RUNNING THE PROGRAM

Once the program has been compiled, run it by typing in "hdfdump" (or whatever name was given to the executable) and then enter the filename that you want to open at the prompt.

A list of grids and fields will then be displayed and you will be prompted to enter the desired numbers. All the MISR gridded data has the dimensions of "XDim" (along-track), "YDim" (across-track) and "SOMBlockDim" (block number), and some fields have extra dimensions such as band or camera number.

If these extra dimensions exist, you have to enter a single value corresponding to the slice of the data that you want to view. For example, if the data is additionally dimensioned by band and camera (such as the BRF's as registered to the Reflecting Level Reference Altitude such as produced by the Level 2 Top-of-Atmosphere/Cloud subsystem), and you enter band=3 and camera=5, you will view the BRF's for the red-band and nadir camera for the specified block range. Since hdfdump is geared to produce 2-dimensional images of the data, you cannot specify dimension ranges for any additional dimensions.

After specifying the index numbers for any existing additional dimensions, you have to enter the block range that you want to dump the data for. Be aware that at the highest resolution (275m) MISR gathers vast quantities of data and trying to dump an entire swath's worth of data at that resolution will overload most computers. However, dumping an orbit's worth of data at 1.1km resolution only requires 160 Meg for 32-bit data. The program will list the blocks for which valid data exists and ask you to enter numbers within that range. HDFDUMP uses absolute block numbers so block 1 will always correspond to the first block in the orbit.

HDFDUMP will then dump the data for the specified slice and block range and will inform you of the number-type of the data and the sizes of the output files. You will be asked to confirm your

choice of data and the resolution of the data. The data resolution calculated by the program will be correct unless the size of a data block changes.

SAMPLE SESSION

```
catherin@khamsin:/scir1/catherine/MISR/path113  [8]>hdfdump
 enter name of HDF-EOS GRID file to open
MISR_AM1_TC_ALBEDO_P113_O002326_F02_41.hdf

 PLEASE CHOOSE GRID/FIELD TO DUMP
 GRID  1  AlbedoregParams
   FIELDS:
         1 NumLocAlbedoRetrieved
         2 NumLocAlbedoRetrievedNoFill
         3 AverageLocAlbedo
         4 AverageLocAlbedoNoFill
         5 StdDevLocAlbedo
         6 StdDevLocAlbedoNoFill
         7 NumDetermSucceeded
         8 NumDetermFailed
         9 NumStochSucceeded
        10 NumStochFailed
        11 RegionalMu0
        12 RestrAlbedoTop
        13 RestrAlbedoTotal
        14 NumSubRestrAlbedoTop
        15 RestrAlbedoSideCameras
        16 NumSubRestrAlbedoSide
        17 RestrAlbedoSidePerCam
        18 ExpansiveAlbedo
        19 NumSubExpansiveConsidered
        20 NumSubExpansiveUsed
        21 NumSubExpansiveMissBrfTop
        22 NumSubExpansiveBadObscTop
        23 NumSubExpansiveMissBrfSide
        24 NumSubExpansiveBadObscSide
        25 NumSubExpansiveOblique
        26 FractionExpansiveClearHC
        27 FractionExpansiveClearLC

 GRID  2  RegParams
   FIELDS:
         1 SolZenAngle
         2 ViewZenAngle
         3 RelativeAzimuthAngle
```

```
  GRID  3  RLRAregParams
    FIELDS:
        1 TopBRF
        2 TopBRFFillFlags
        3 SideBRF
        4 SideBRFFillFlags
        5 NumUnobscuredTop
        6 NumUnobscuredSide
        7 TopTextureIndex1
        8 TopTextureIndex2
        9 TopTextureIndex3
       10 LocalAlbedo
       11 CloudyClearDesig
       12 CloudyClearSource
       13 CloudLiquidPercent
       14 CloudPhaseOverrideFlag
       15 CloudTopTemp
       16 SurfaceType
       17 CsscSurfaceType
       18 HighCloudIndicator
       19 HomogeneityRefCam
       20 Texture
       21 MethodandSAWReason
       22 ClearSkyR0
       23 ClearSkyKappa
       24 ClearSkyBeta
       25 ClearSkyCameras
       26 ClearSkyChiSquare
       27 SolZenAngIndex
       28 CamZenAngIndex
       29 RelAzimBinInterp
       30 BlueAltBinIndex
       31 GreenAltBinIndex
       32 BrightnessIndex
       33 IglooIndex

ENTER GRID NUMBER:  1 -  3
3
ENTER FIELD NUMBER:  1 - 33
1
 YOU HAVE SELECTED GRID/FIELD = RLRAregParams : TopBRF

 NAMES AND SIZES OF DIMENSIONS
NCamDim                         9
NBandDim                        4
```

```
YDim                          256
XDim                           64
SOMBlockDim                   180

 select slice number for dimension = NCamDim
5
 select slice number for dimension = NBandDim
3

 ENTER FIRST AND LAST BLOCKS TO READ IN
 USE ABSOLUTE BLOCK NUMBERS
 VALID DATA EXISTS FOR BLOCKS:  93  -  101
93 101

 DETERMINING DATA RESOLUTION
 LIKELY RESOLUTION = (LTRACK,XTRACK) =   2200.0  2200.0
 IS THIS CORRECT? (Y/N)
Y

 IS DATA SELECTION OK? (Y/N)
Y

 DATA TYPE IS 32-BIT FLOATING POINT
 WRITING OUT CONTENTS OF DATA FILES
 NLINE =  576
 NSMP =  256
 NSMP+IPAD =  328
```

OUTPUT OF PROGRAM

HDFDUMP will produce two binary files corresponding to two different ways of dumping the data. The MISR blocks are not directly aligned with each other, but are rather offset from one another in units of 17.6km to account for the curve of the swath. One output file will properly align all the blocks to produce a continuous picture and the other simply stacks the blocks one of top of the other. This latter file is mostly useful for debugging purposes if you want to exactly pinpoint the line and sample number with respect to the start of the block for a single pixel.

Both files are written out as unformatted flat binary with no header bytes. These can easily be read into almost any image analysis package or into your own program. The datatype of these files is as indicated in the output of the program, 32-bit floating point for the example session shown above.

The "aligned" file is given the name of '*img.align.bin*' and the stacked file is called '*img.stack.bin*'. The *img.align.bin* has NLINE lines and NSMP+IPAD columns, and the dimensions of the *img.stack.bin* file are NLINES by NSMP.

The following diagram illustrates how the MISR blocks are aligned with respect to each other in a typical portion of the swath.
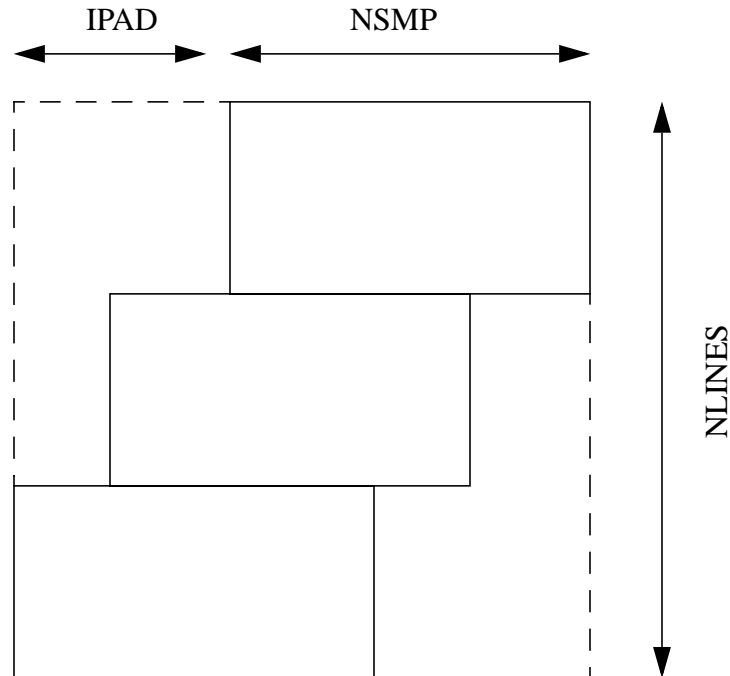
COMPILING CODE

The source code is provided for you free of charge to serve as a learning tool for those people who eventually want to write their own HDF-EOS access code. To compile the code you need a Fortran90 compiler (that has the option of extending the source code lines to 80 characters), and the HDF and HDF-EOS libraries. A copy of the executable compiled and run on an SGI IRIX system (with the -n32 flag) has also been provided.

Set the following environmental variables:
        HDFINC = location of HDF include files
        HDFLIB = location of HDF libraries
        HDFEOS_LIB = location of HDF-EOS libraries

```
prompt>  f90 -col80 -o hdfdump hdfdump_public.f \
         -I$HDFINC -L$HDFLIB -lmfhdf -ldf -ljpeg -lz \
         -L$HDFEOS_LIB -lhdfeos -lGctp
```

The HDF and HDFEOS libraries can be found at the following locations:
      HDF: http://hdf.ncsa.uiuc.edu
      HDFEOS:http://newsroom.gsfc.nasa.gov/sdptoolkit/toolkit.html

Note that when you compile the code for the first time it will probably fail, giving error messages about modules that it can't find. This is an expected error; recompile once or twice more and everything should work fine. The first time you compile, it compiles the mainline first of all and sees "use statements" for several modules - so it then looks for the corresponding .mod files which it can't find because it hasn't yet compiled the individual modules. Once the code has been compiled once, all the necessary .mod files now exist so the next compilation should succeed.

CONTACT INFORMATION

If you have any questions about HDFDUMP or problems getting it to run, please contact the author at the following address:

Catherine Moroney
Jet Propulsion Laboratory
Mail Stop 169-315
4800 Oak Grove Drive
Pasadena, CA 91109-8099

(818) 393-3392
Catherine.M.Moroney@jpl.nasa.gov

COPYRIGHT INFORMATION

Last Updated: November 9, 2000