MESSENGER MDIS
Data Users' Workshop 2013

Kris Becker

Tammy Becker

Trent Hare

USGS Astrogeology Science Center

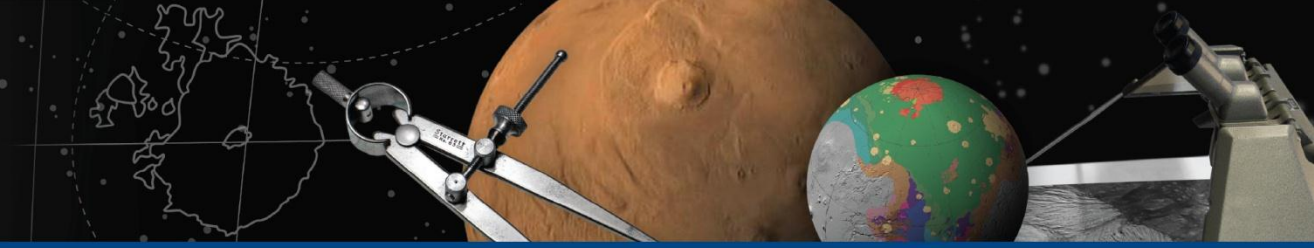
44th LPSC

March 17, 2013

ISIS3 Tutorial and MESSENGER MDIS Data Users' Workshop

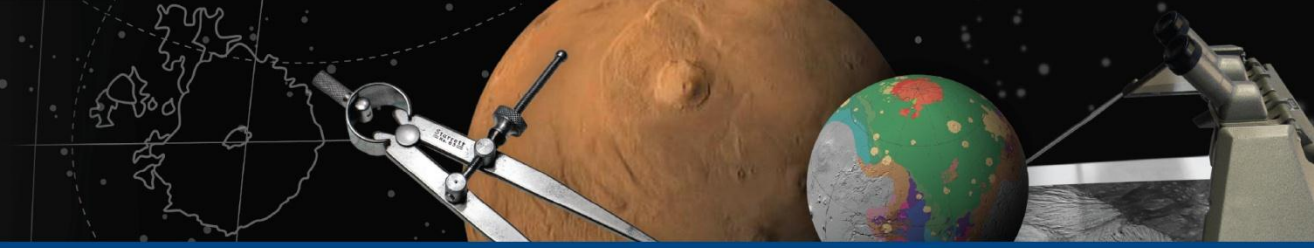
- Objective: Provide an introduction to ISIS3 and demonstrate its use in processing of MESSENGER MDIS data

1. ISIS3 Fundamentals
2. Standard Processing Concepts and Tools
3. Cartographic Map Projections
4. Creating Monochrome Map Mosaics (EDRs)
5. Creating Color Map Mosaics (EDRs)
6. Working with PDS Map Projected Products (BDRs/MDRs)
7. Export and Application Support for ISIS3 Products



ISIS - Integrated Software for Imagers and Spectrometers

- Over 300 image processing applications
- Strong emphasis on geometric functionality
  - Photogrammetry / Camera models
  - Cartography / Map projections
  - Photometry
  - Improving instrument position and orientation
    - Image control networks
    - Bundle adjustment (**jigsaw**)
  - Digital map mosaics
- In use for over 30 years (PICS, ISIS2, ISIS3)
- Support for over 50 NASA/ESA instruments
- Support for MESSENGER MDIS Narrow Angle (NAC) and Wide Angle (WAC) Cameras



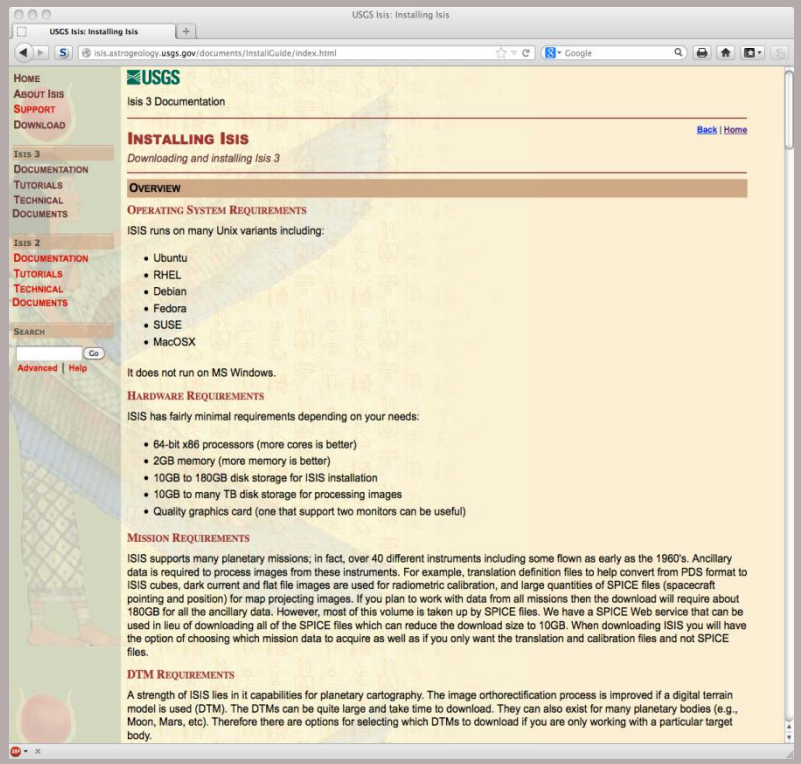
Mission Instruments Supported by ISIS3

- Lunar Orbiter III, IV, &, V (Medium and HiRes)
- Clementine UVVIS, NIR, HIRES, & LWIR
- Apollo Metric 15/16/17
- Apollo Panoramic 15/16/17
- Lunar Reconnaissance Orbiter NACL, NACR, WAC (VIS & UV), MiniRF
- Chandrayaan-1 MiniRF
- Mariner 10 (A & B)
- MESSENGER MDIS (NAC & WAC)
- Kaguya MI (VIS & NIR)
- Dawn FC (1 & 2), VIR
- Mars Global Surveyor MOC (NAC & WAC)
- Mars Odyssey THEMIS (VIS & IR)
- Mars Express HRSC
- Mars Reconnaissance Orbiter HiRISE, CTX, MARCI, CRISM
- Viking Orbiter 1 & 2 (A & B)
- Voyager 1 & 2 (NAC & WAC)
- Galileo SSI
- Cassini ISS (NAC & WAC), VIMS, RADAR
- Ideal Camera (Special ISIS Perfect Virtual Instrument – Distortion-Free!)

Fifty-five instruments in all!!

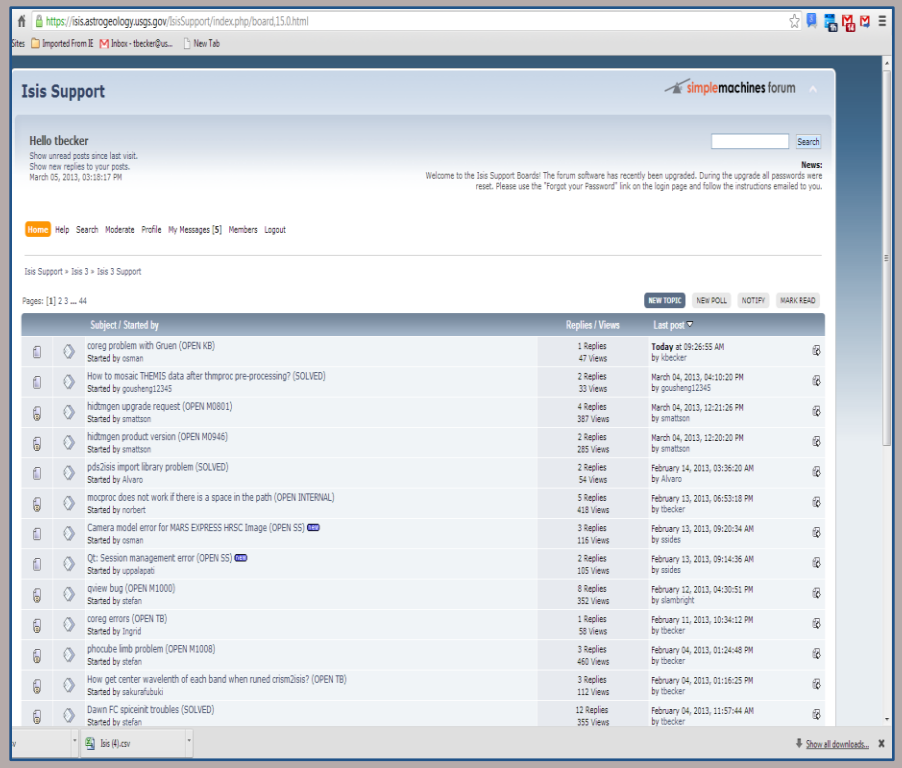
Current ISIS Status

- ISIS 3.4.2 (Released Dec 2012)
- ISIS 3.4.3 (Scheduled Release: End of March 2013)
- UNIX-based Supported Platform OSes
  - Mac OSX 10.6 and higher (32 & 64 bit Intel)
  - Debian 6.0.2 (64 bit)
  - Debian 7 (64 bit)
  - Fedora 16 (64 bit)
  - Fedora 18 (64 bit)
  - Redhat Enterprise 6.3 (64 bit) (via SL 6.3)
  - Scientific Linux (SL) 6.3 (64 bit)
  - SUSE Enterprise Server 11 (64 bit)
  - Ubuntu 12.04 LTS (64 bit)
- Download via Internet
  - Full distribution >200GB
  - Selective download using **rsync** utility
  - Java client installer



http://isis.astrogeology.usgs.gov/documents/InstallGuide

ISIS3 Documentation, Support and User Guides



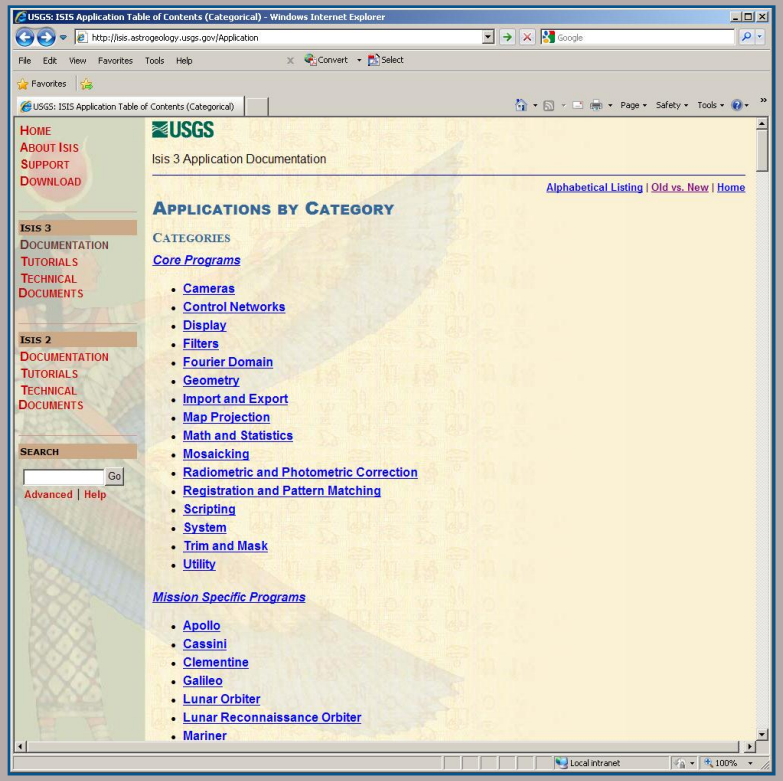
- General Information
  http://isis.astrogeology.usgs.gov
- Installation Guide
  http://isis.astrogeology.usgs.gov/documents/InstallGuide
- Table of ISIS Applications
  http://isis.astrogeology.usgs.gov/Application
- User Support Forums
  http://isis.astrogeology.usgs.gov/IsisSupport
- Online Workshops
  http://isis.astrogeology.usgs.gov/IsisWorkshop

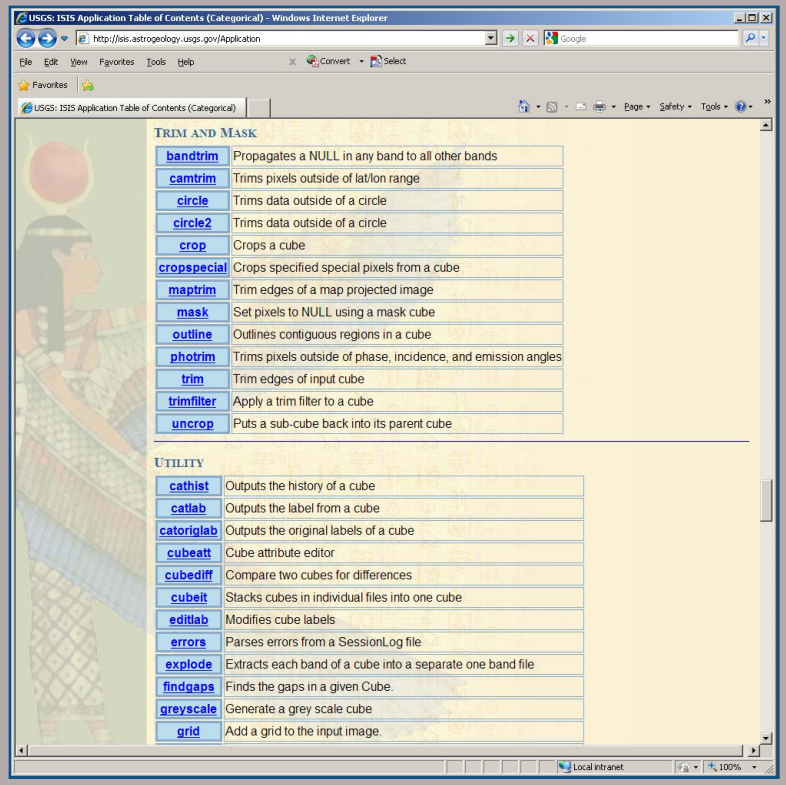
How To Download and Install ISIS 3

- Start at the ISIS Website
  - http://isis.astrogeology.usgs.gov
  - See the 'Install Guide' for info on installing ISIS for your OS and platform
    - http://isis.astrogeology.usgs.gov/documents/InstallGuide
  - Use the Java Client Installer (preferred)
    - **rsync –azv isisdist.wr.usgs.gov::installer .**
    - **java –jar install.jar**
  - Or manual installation
    - Create ISIS3 directory (no spaces!) and then download the software and two directories there
    - Base data and MESSENGER mission-specific data are both required
    - Use **rsync** to download the latest version of ISIS3 applications and data files
      - Example for MAC OSX 10.6 and higher (applications):
        - **rsync -azv --delete isisdist.wr.usgs.gov::x86_darwin_OSX/isis .**
        - Be sure to include the '.' at the end!!!
      - Minimum requirements for MESSENGER (data), two directories:
        - **rsync -azv --delete isisdist.wr.usgs.gov::isis3data/data/base data/**
        - **rsync -azv --delete isisdist.wr.usgs.gov::isis3data/data/messenger data/**
- Set up environment variable and run startup script (example for C/T shells)
  - setenv ISISROOT /work1/isis3/isis
  - source $ISISROOT/scripts/isis3Startup.csh
- Frequent updates to ISIS3 and ancillary data is recommended
  - Maintains ISIS system with software patches
  - Keeps SPICE kernels up to date for active missions
  - Just rerun **rsync** installation commands above

ISIS3 Application Documentation

ISIS Application Documentation is organized by Functional Categories and Mission Specific Programs





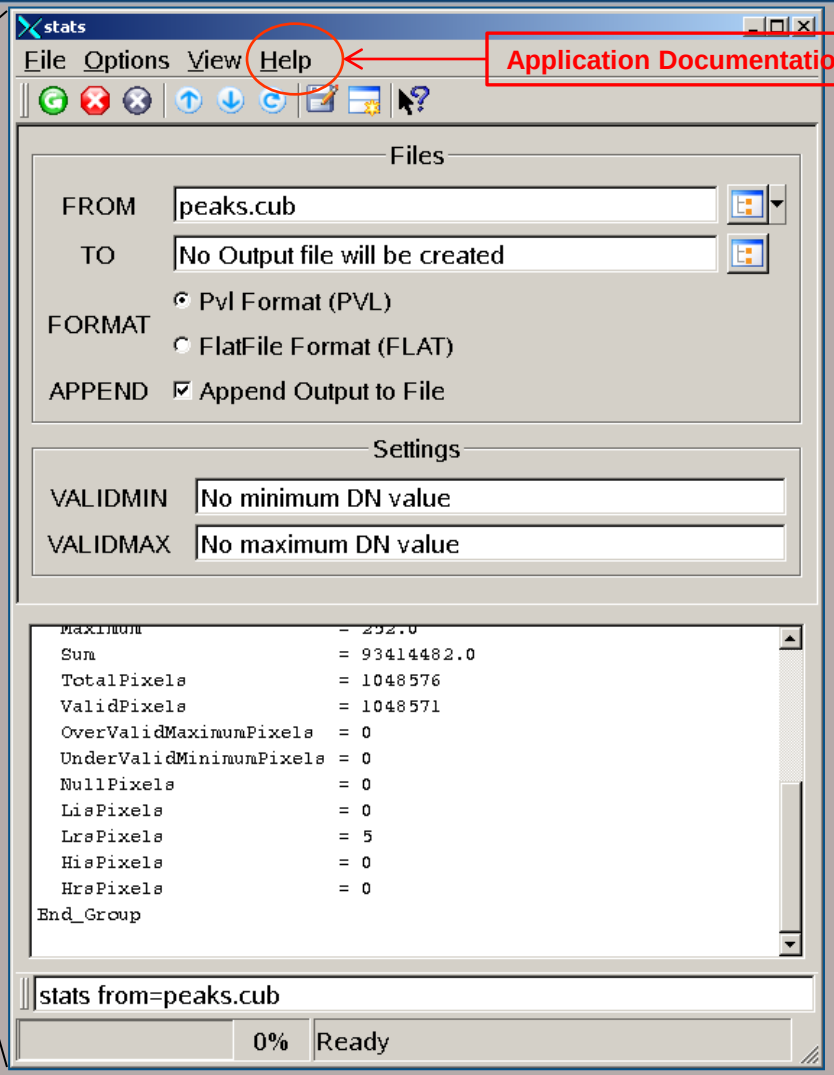
Executing ISIS3 Applications

Application GUI Interface

Or…

Applications can be executed at the command line

Command line enables key feature – batch execution



See http://isis.astrogeology.usgs.gov/documents/CommandLine/CommandLine.html for details on this feature.

Selecting a '**bolded/underlined**' term will display the **Glossary**

USGS

Isis 3 Application Documentation

Standard View | TOC | Home

CAMPT

Computes geometric and photometric information at a given pixel location

Description
Categories
Groups
History

Description

Campt computes geometric and photometric information at a given pixel location in the input image **cube**. The program computes spacecraft and instrument related information, and other types of coordinates as described later in this document. The user will have a choice of coordinates in which to output the information as well as a choice in the output format of the information acquired.

Note the input image cube has preliminary requirements:

- The input image requires **SPICE** information (see ***Spiceinit***).
- The input image cube must be a **Level0** or a **Level1** ISIS cube.
- To use a **Level2** file as the input image (see ***Mappt***).

The point of interest in the image can be entered as **Latitude**/**Longitude** coordinates or **Line**/**Sample** coordinates. Keep in mind that the input **Latitude** and **Longitude** values entered will be interpreted as **Universal Coordinates** (ISIS default) regardless of the target body. In the output, all positions are in **Body-Fixed** Coordinates.

The following is a partial list of coordinates computed in the campt application:

Cube

A cube is a 3-dimensional image with axis: samples, lines, and bands. The physical dimensions of a cube are called the number of samples (NS), number of lines (NL), and number of bands (NB).

Typically, the sample and line dimensions are used to represent spatial information while the band dimension represents spectral information.

Declination

Declination (Dec) is one of two angles of the north pole of a target body as a function of time.

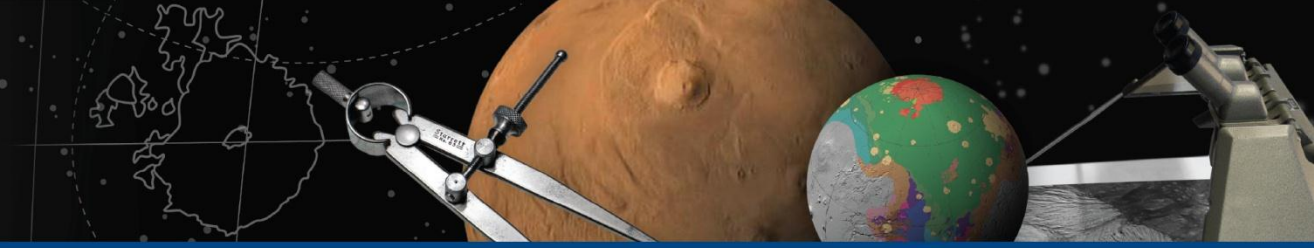
For more information, refer to Euler Angles, Right Ascension and Prime Meridian

Detector Resolution

The size of the detector on the focal plane for each pixel.

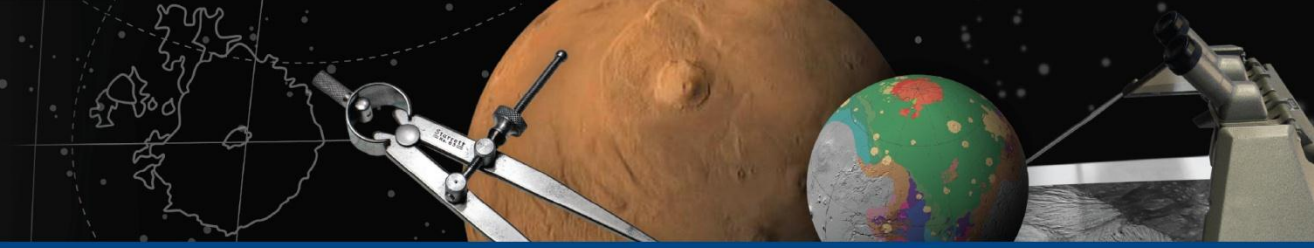
Digital Number

The numeric value of a single pixel in an image. The value may represent almost any unit. For example: reflectance (I/F), radiance, elevation, or radius. Digital Numbers (DNs) can be discrete integers or floating point values.



ISIS3 Terminology

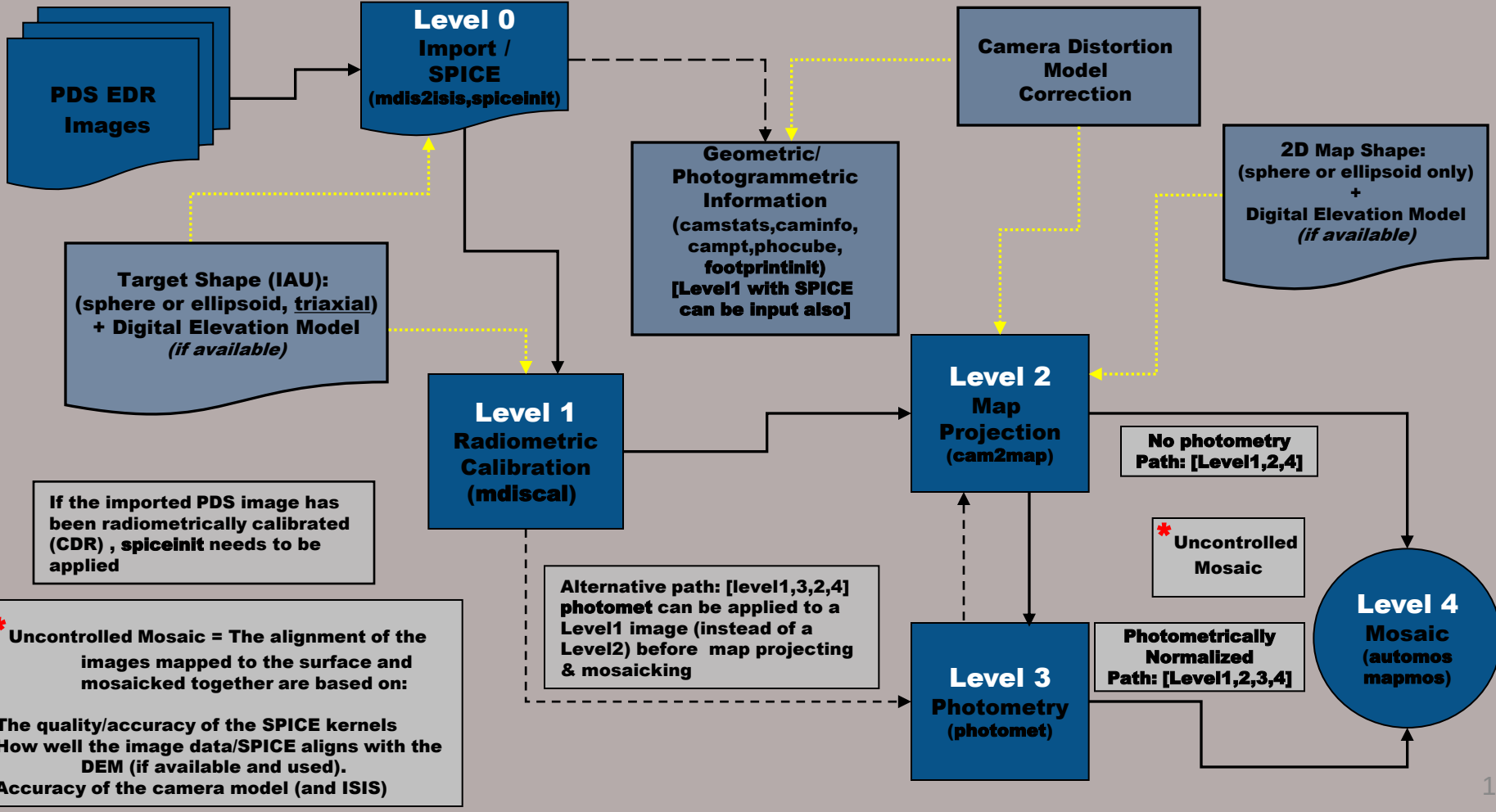
- **Level 0**
  - Decompressed spacecraft data
  - Import PDS EDR into ISIS3
- **SPICE (required)**
  - **S**pacecraft & **P**lanetary ephemerides,
  - **I**nstrument **C**-matrix and **E**vent kernels
  - ISIS3 uses the NAIF ToolKit for SPICE
- **Level 1**
  - Radiometric calibration
  - Noise Removal (optional)
- **Level2**
  - Project image to map coordinates
  - Camera distortion correction applied
- **Level 3**
  - Photometric normalization (optional)
- **Level 4**
  - Mosaicking (optional)
- **Glossary**
  - The user documentation for many ISIS3 applications link to a Glossary of definitions



ISIS Support for MESSENGER

- MESSENGER Development and Support Activities
  - Camera Models for Narrow Angle (NAC) and Wide Angle (WAC) cameras
  - Distribution of SPICE kernels
  - Radiometric calibration
  - Camera distortion correction
  - Photometric correction
    - Parameter setting are not released with ISIS, but will be supplied by command line in this tutorial for NAC and WAC filters
  - Generating global monochrome and color maps
  - Processing of PDS MDIS EDR, CDR, BDR and MDR data
  - Participate in development of special products (uncontrolled/controlled maps, updated kernels, stereo products and DEMs)
- NASA, MESSENGER Project, Johns Hopkins University Applied Physics Laboratory and Arizona State University have provided funding and/or support to the USGS for the development of ISIS3 software and MDIS data products

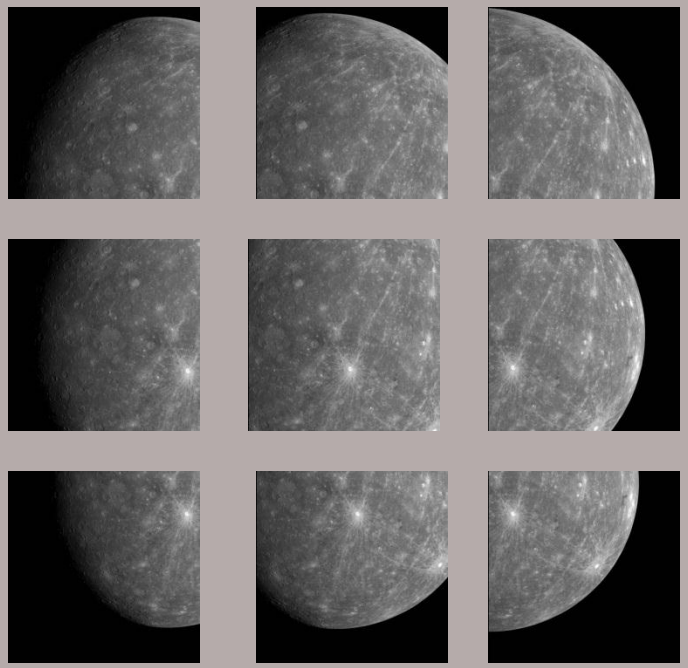
Standard MESSENGER Processing Flow within ISIS3

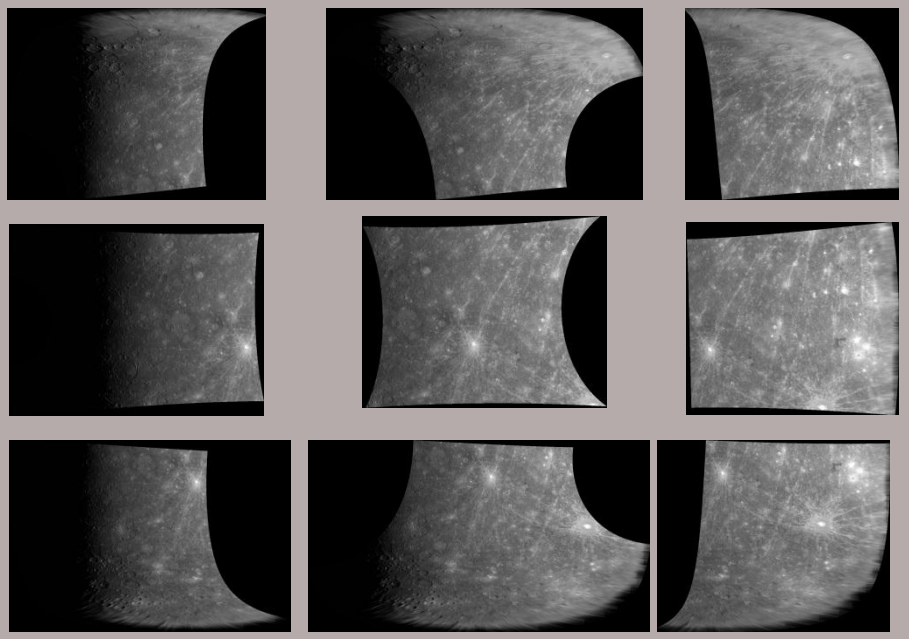


Level 1 Calibrated with SPICE

Level 2 Equirectangular Map Projection*

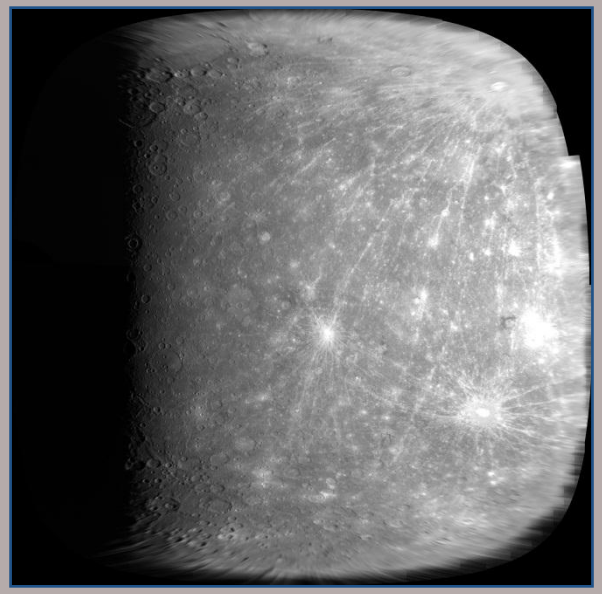
*The same map resolution and center longitude was defined for all images as required to mosaic



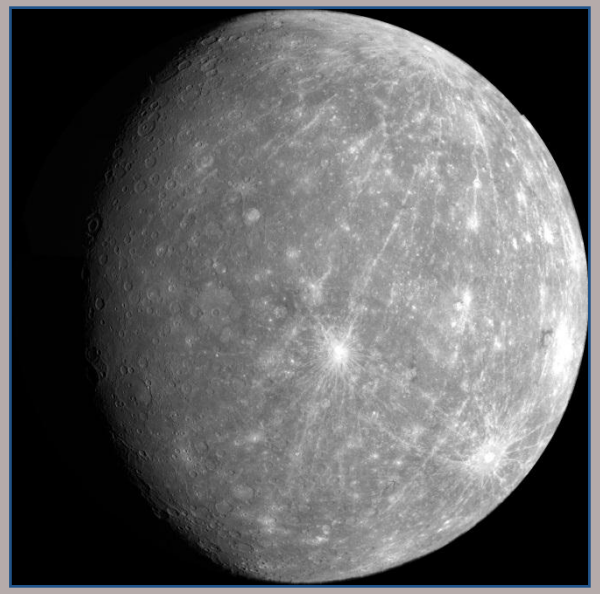


M2 Depart Color 3x3 WAC-G Filter

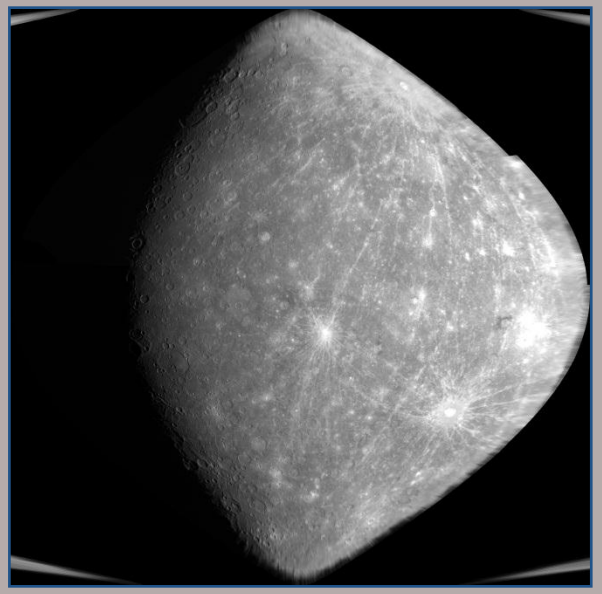
Level 4 MOSAICS
Different Map Projections



Equirectangular



Orthographic



Sinusoidal

M2 Depart Color 3x3 WAC-G Filter

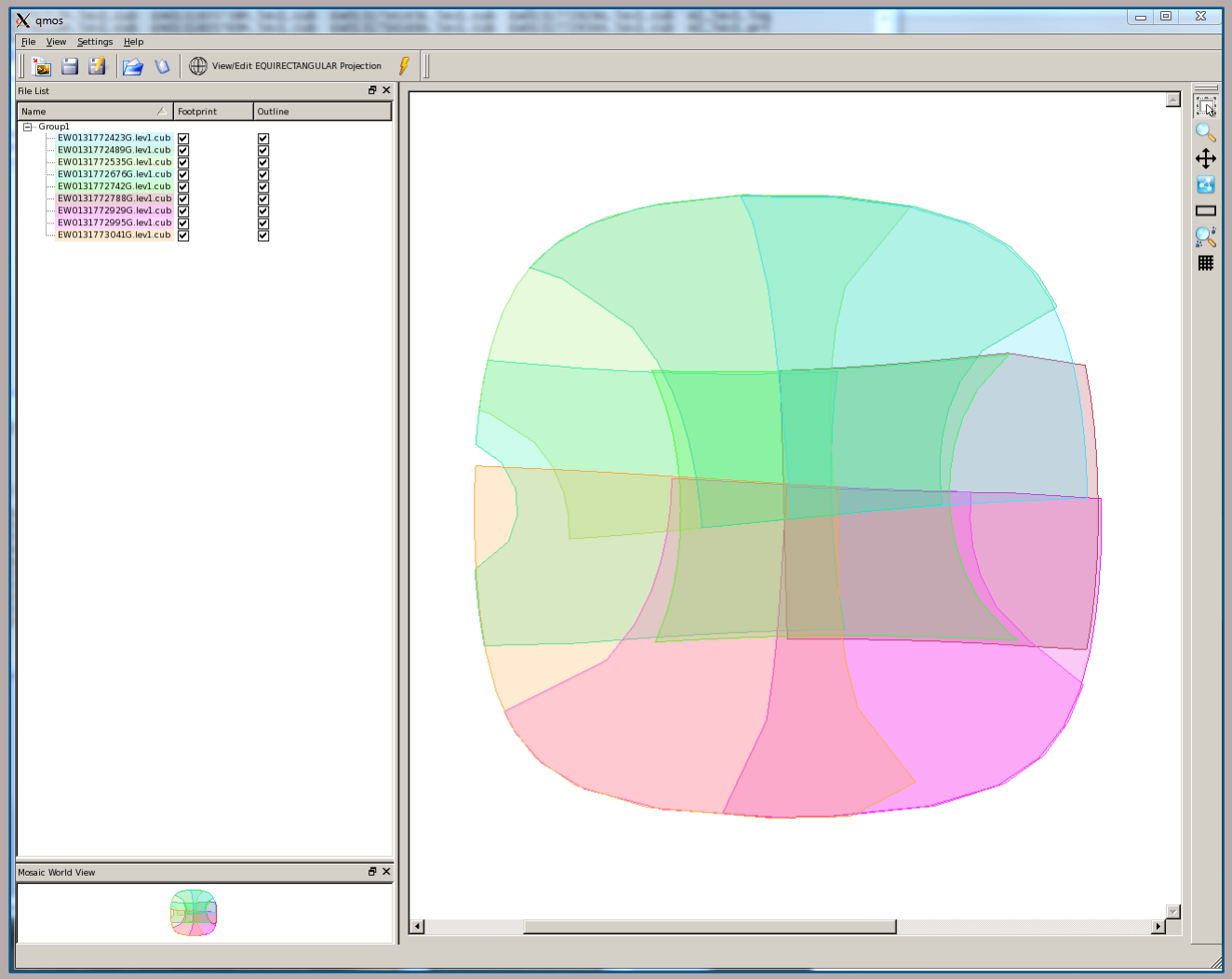
qmos

Interactive Footprint Display

Displays Level1 or Level2 image footprints in any ISIS map projected format

Requirements:

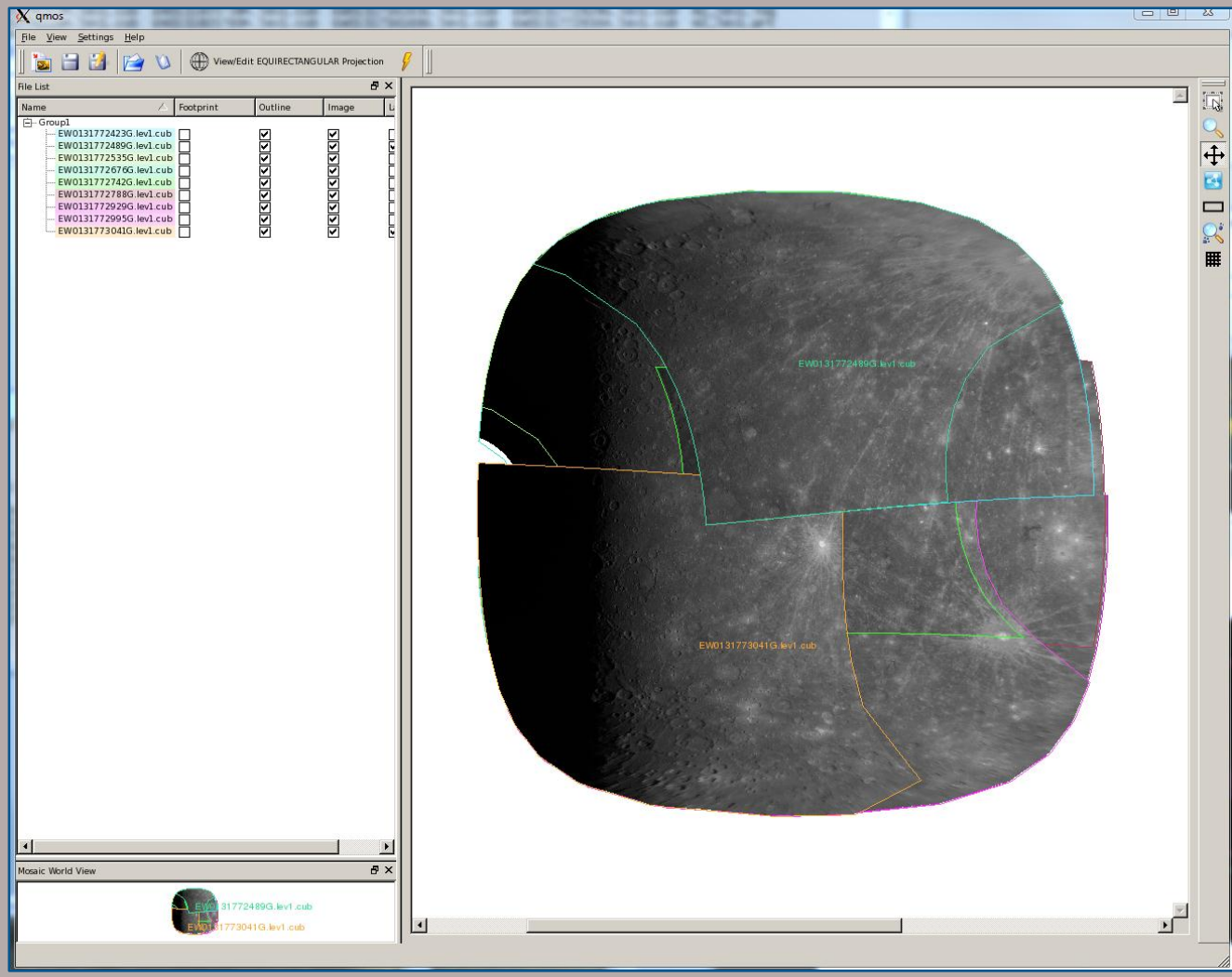
- -SPICE (**spiceinit**)
- -Polygon (**footprintinit**)
- -Geometry/Photometric Info (**camstats**)



qmos

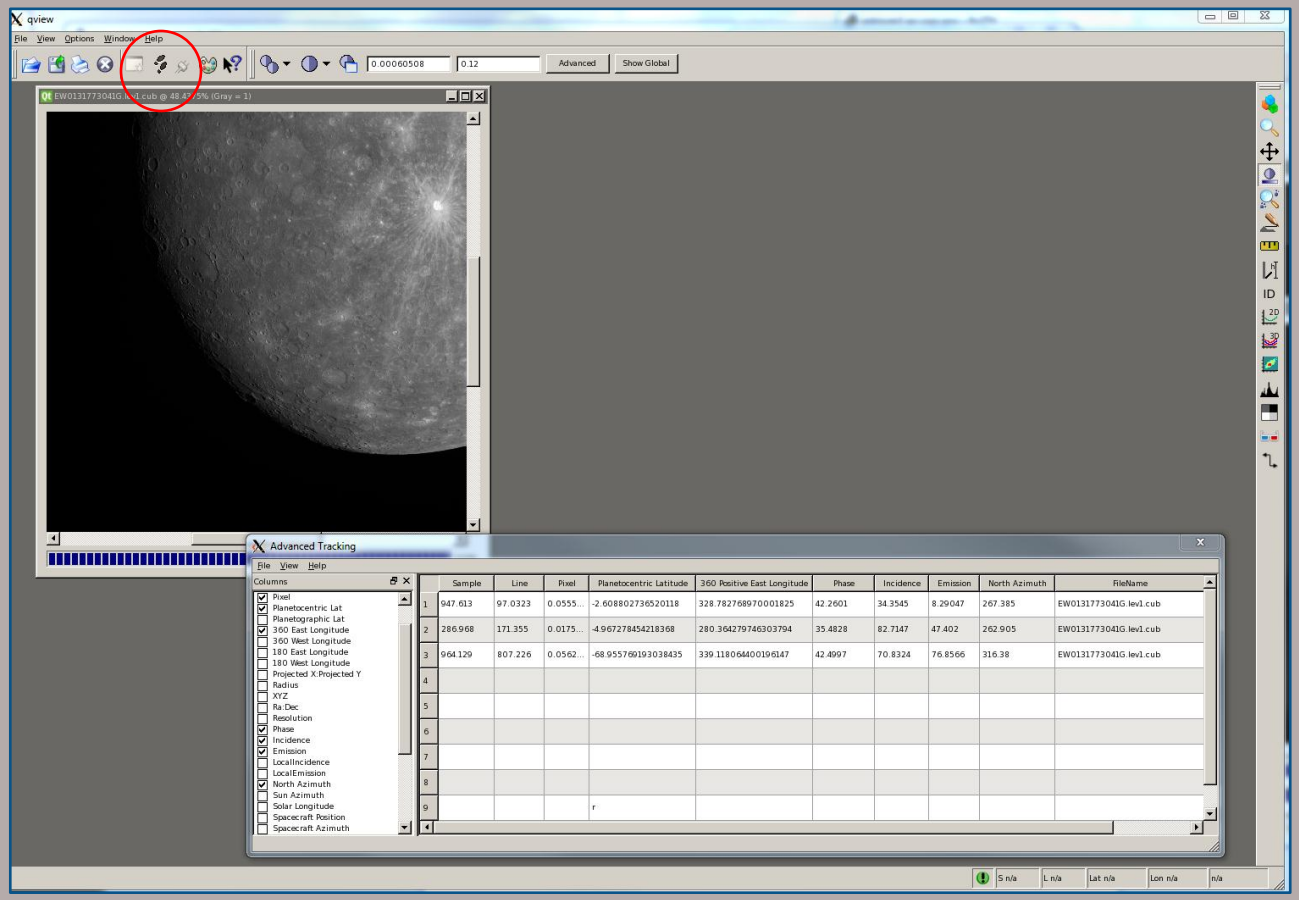
Interactive Footprint Display

qmos can display imagery! Use with caution! The rendering of images slows the response time significantly!



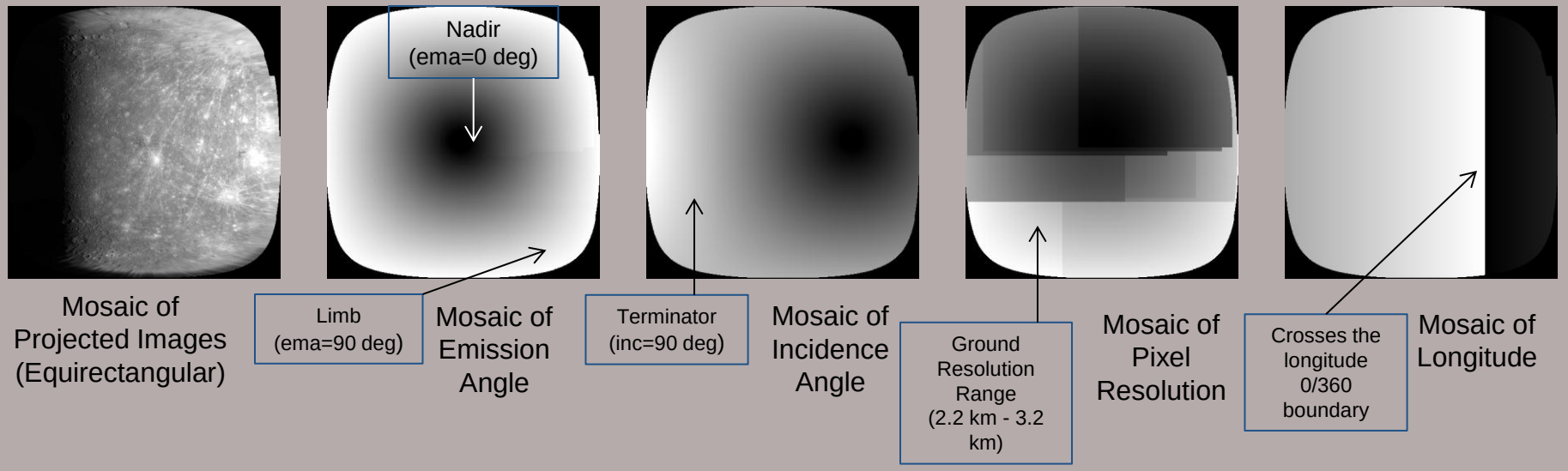
GEOMETRIC and PHOTOMETRIC TOOLS

qview – Interactively reports information at every pixel



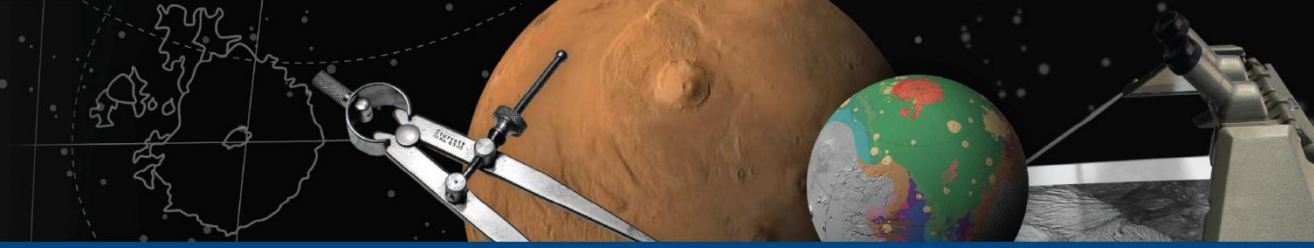
GEOMETRIC and PHOTOMETRIC TOOLS

phocube – Application generates data for every pixel and creates an output cube



There are currently 20 data options available in **phocube**

Additional applications that report observation, geometric and photometric information
campt, camstats, caminfo, camrange

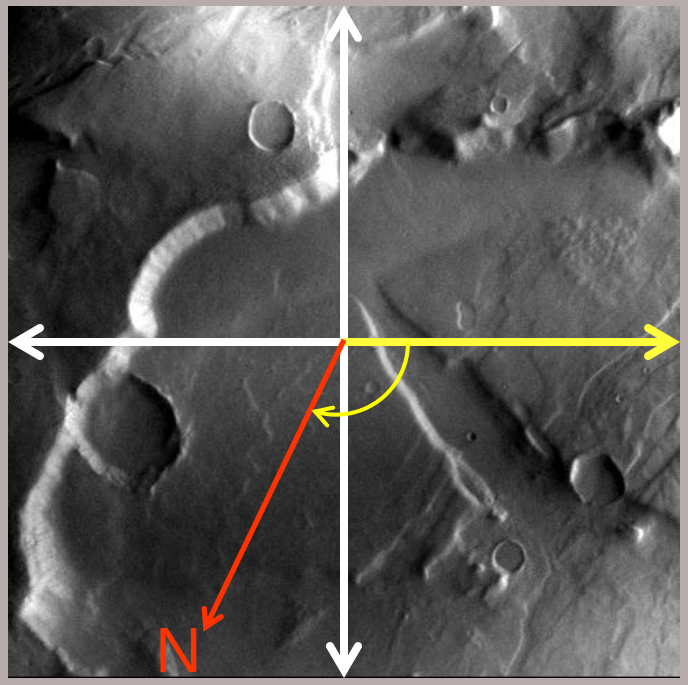


Output of campt

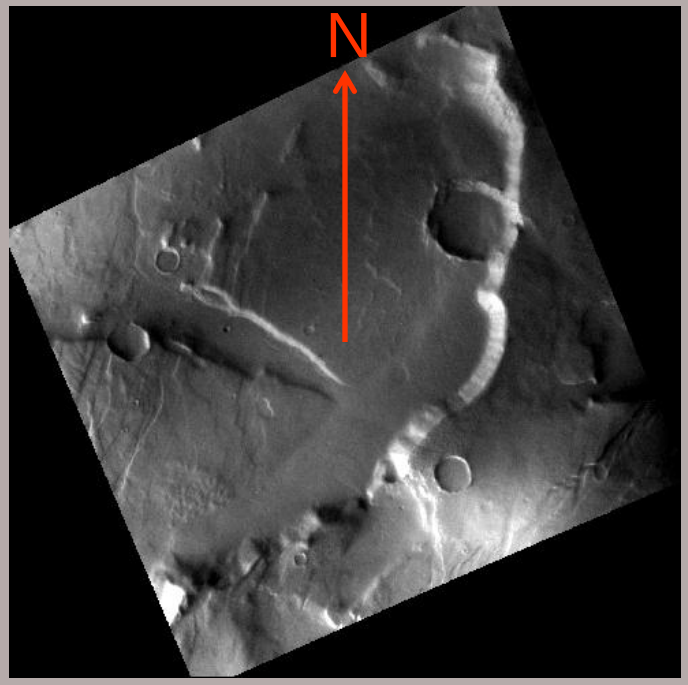
```
Group = GroundPoint
  Sample                  = 512.0
  Line                    = 512.0
  PixelValue              = 0.00260657
  RightAscension          = 19.722259374476
  Declination             = -59.58327428227
  PlanetocentricLatitude  = 48.354237551489
  PlanetographicLatitude  = 48.689922325464
  PositiveEast360Longitude = 277.86849622301
  PositiveEast180Longitude = -82.131503776986
  PositiveWest360Longitude = 82.131503776986
  PositiveWest180Longitude = 82.131503776986

 # Sun Information
  SunPosition             = (-177533702.4284,
                             -103045154.79627,
                             -47419881.109118)<km>
  SubSolarAzimuth         = 359.48348806842
  SolarDistance           = 1.4082928265777 <AU>
  SubSolarLatitude        = -13.007723818623
  SubSolarLongitude       = 210.13198846511
  SubSolarGroundAzimuth   = 218.08431231392

  # Illumination and Other
  Phase                   = 65.955039121282
  Incidence               = 85.578537079895
  Emission                = 20.534413411423
  NorthAzimuth            = 111.81291300246

  # Time
  EphemerisTime           = 288188598.81919 <seconds>
  UTC                     = 2009-02-18T00:22:12.634
  LocalSolarTime          = 16.51576718386 <hour>
  SolarLongitude          = 211.92252854089
EndGroup
```

Azimuths: North, Sun, and Spacecraft

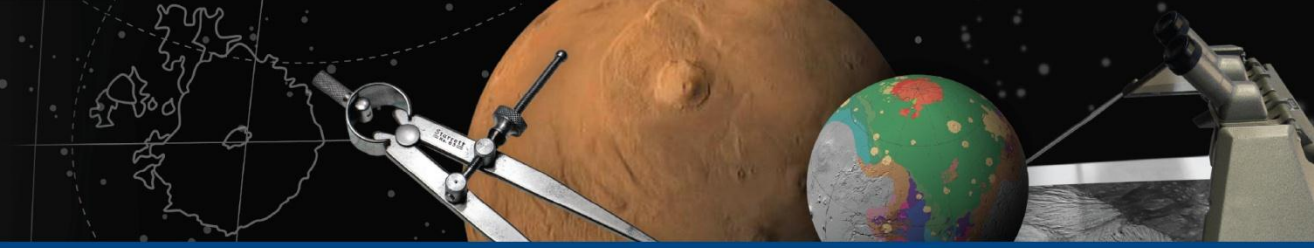
Image Azimuth Reference Line at Center Pixel



Ground Azimuth Reference Line



North azimuth ~112 degrees clockwise from reference line drawn from center point to right of side of image (3 o'clock)

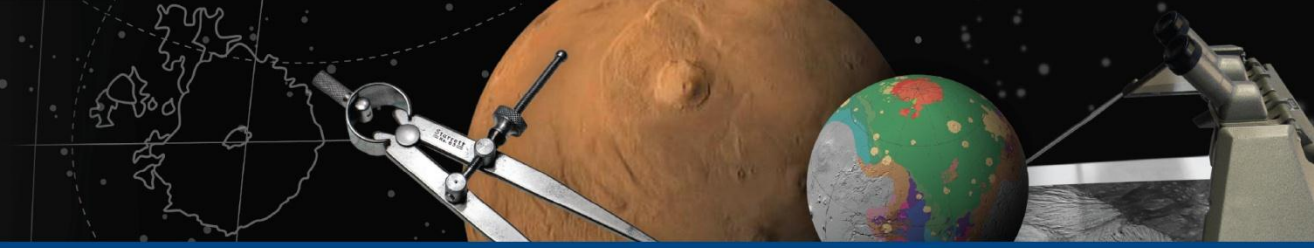


Supported Map Projections in ISIS3

- Equirectangular
- Lambert Conformal
- Mercator
- Oblique Cylindrical
- Orthographic

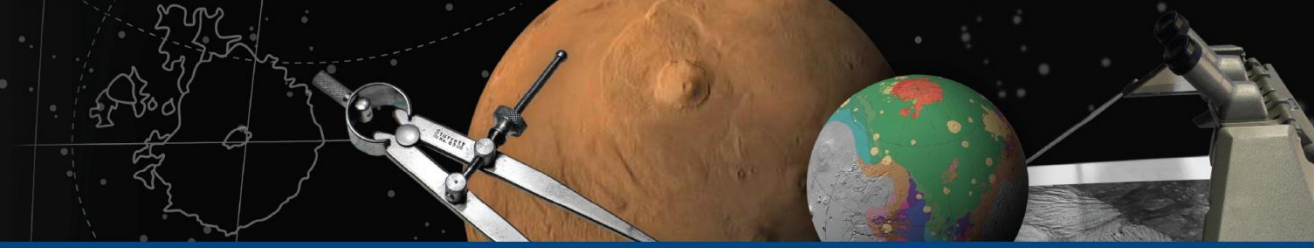
- Point Perspective
- Polar Stereographic
- Simple Cylindrical
- Sinusoidal Equal Area
- Transverse Mercator

Interactive on-line map projection tutorial:
http://isis.astrogeology.usgs.gov/IsisWorkshop/index.php/Learning_About_Map_Projections



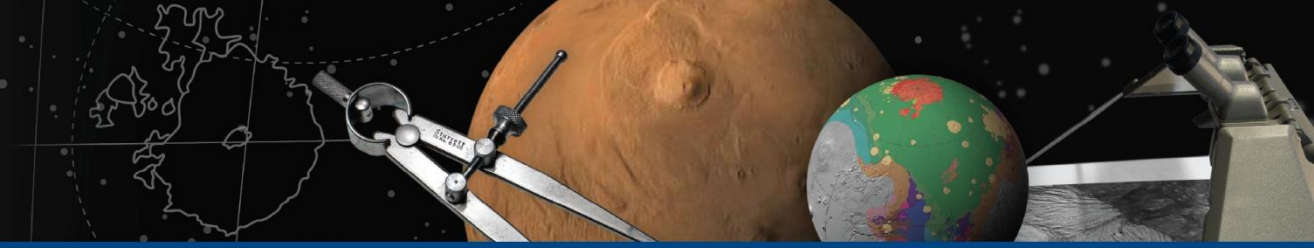
Default Map Projection

- ISIS3 map projection tools default to:
  - Sinusoidal map projection ***($ISIS3DATA/base/templates/maps/sinusoidal.map)***
  - IAU target body shape (ellipsoid or sphere)
  - Planetocentric latitude system
  - Positive longitude direction is East
  - Longitude Domain 360 [longitude range is defined as 0 to 360 degrees]
  - Remaining required map definitions are computed automatically
    - Pixel resolution (meters/pixel)
    - Ground range (latitude and longitude extents)
    - Center latitude/longitude



Define your own Map Projection

- Parameter files (map templates) are used to define the output map projection
  - Any map parameter can be set to over-ride the defaults
- A map template is available for every supported projection
  - $ISIS3DATA/base/templates/maps/
- Applications with a GUI interface to the map parameters
  - **maptemplate**
  - **mosrange**
- Other ISIS3 map projected image cubes
  - Simply provide an ISIS image cube in the *map* parameter to **cam2map** or **map2map**

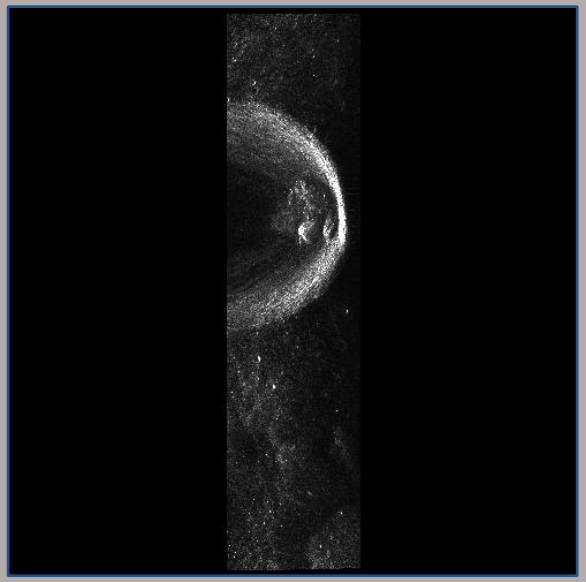


Target Body Properties

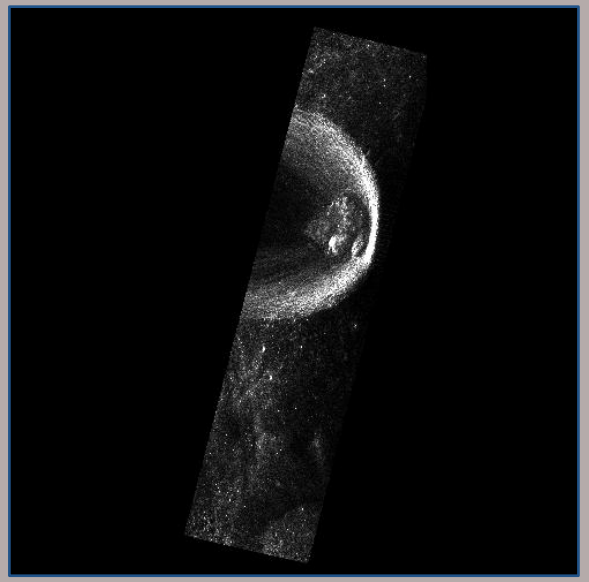
- ISIS3 uses the IAU target radius values from NAIF SPICE kernels for the output map projection
  - The IAU radius for Mercury is: 2439.7 km
  - The MESSENGER Team uses: 2440.0 km
    - Supported within ISIS3 MESSENGER kernel subsystem
  - Be aware of this when merging or analyzing Mercury map products (i.e., Mariner 10) that are mapped to the IAU radius
    - Mariner 10, supported in ISIS3, uses the IAU radius value of 2439.7 km whereas MESSENGER uses 2440.0 km for Mercury's radius!
  - An ellipsoid or mean radius will be used for a triaxial body (e.g., Io, Vesta)
- Orthorectification
  - A Digital Elevation Model (DEM) can be used when projecting images
  - ISIS3 defaults to LOLA for the Moon and MOLA for Mars
  - Mercury shape model coming
    - March 2014

Orthorectification

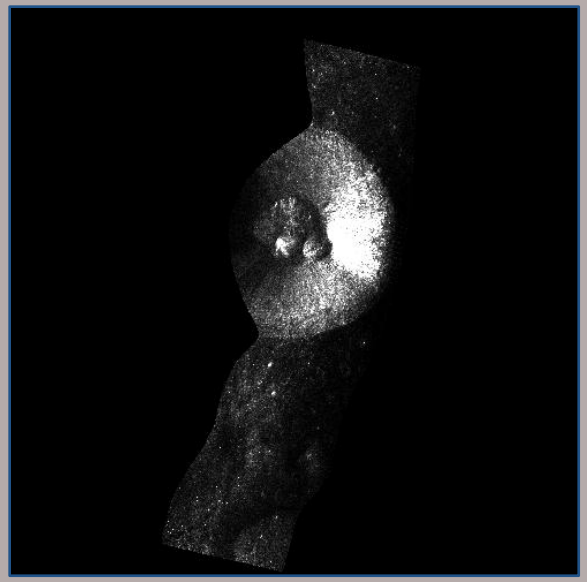
- Removing effects of topography in the map projection
  - The DEM must be provided in the **spiceinit** application
  - Shackleton crater example below [LRO-MiniRF LSZ_02261_1CD_XKU_89S140_V1]



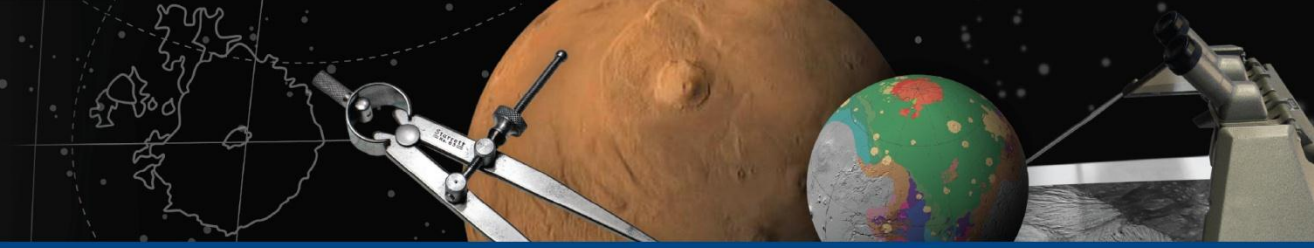
LRO MiniRF Level 1



Projected onto sphere



Projected onto DEM



Using ISIS3 to Process MESSENGER MDIS Data

Creating Regional MDIS Monochrome Mosaics

- Objective: Use ISIS3 to create an MDIS monochrome mosaic of a specific region of interest (ROI)
  - Create from PDS EDRs
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_1001/
- Target ROI: Raditladi Basin
- General Processing Steps
  - Determine MDIS images to include in mosaic (PDS, PILOT, etc…)
    - http://pilot.wr.usgs.gov/
  - Download images from source
  - Process using ISIS3
    - Import, apply SPICE, radiometric calibration (CDR), map project, photometric correction and assemble mosaic
- See the RaditladiBasin demo

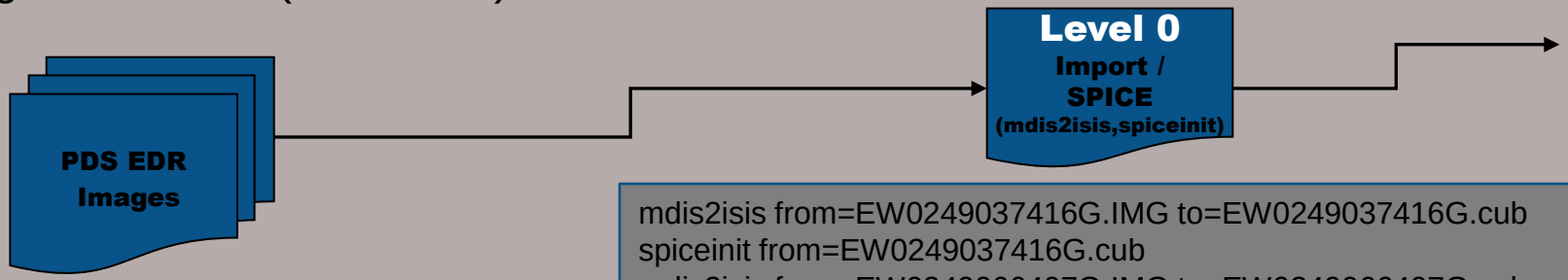
Import and Applying SPICE

Region of interest: **Raditladi Basin**

(http://www.nasa.gov/mission_pages/messenger/multimedia/messenger_orbit_image20130114_1.html)

Latitude Range=(17N to 36.5N) Longitude Range=(111E to 128E)

Wavelength Filter = G (748.7 NM) InstrumentId = MDIS-WAC



Selected PDS EDR Images:

2012_176/EW0249037416G.IMG
2012_176/EW0249066407G.IMG
2012_176/EW0249037545G.IMG
2012_176/EW0249037688G.IMG
2012_176/EW0249066345G.IMG
2012_176/EW0249037671G.IMG
2012_176/EW0249037562G.IMG

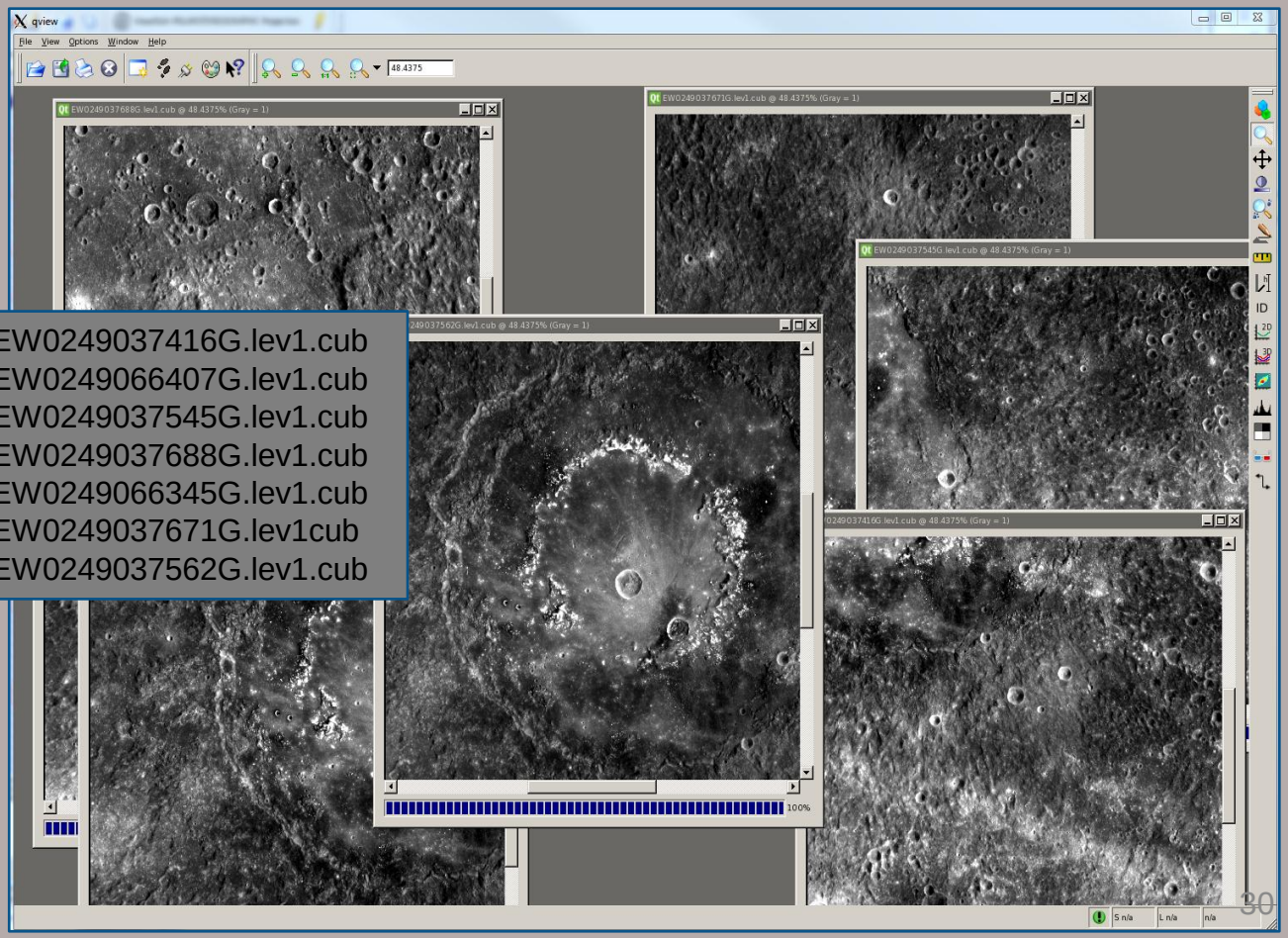
```
mdis2isis from=EW0249037416G.IMG to=EW0249037416G.cub
spiceinit from=EW0249037416G.cub
mdis2isis from=EW0249066407G.IMG to=EW0249066407G.cub
spiceinit from=EW0249066407G.cub
mdis2isis from=EW0249037545G.IMG to=EW0249037545G.cub
spiceinit from=EW0249037545G.cub
mdis2isis from=EW0249037688G.IMG to=EW0249037688G.cub
spiceinit from=EW0249037688G.cub
mdis2isis from=EW0249066345G.IMG to=EW0249066345G.cub
spiceinit from=EW0249066345G.cub
mdis2isis from=EW0249037671G.IMG to=EW0249037671G.cub
spiceinit from=EW0249037671G.cub
mdis2isis from=EW0249037562G.IMG to=EW0249037562G.cub
spiceinit from=EW0249037562G.cub
```

http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_1001/DATA/2012_176/

Radiometric Calibration

Level 1
Radiometric Calibration (mdiscal)

```
mdiscal from=EW0249037416G.cub to=EW0249037416G.lev1.cub
mdiscal from=EW0249066407G.cub to=EW0249066407G.lev1.cub
mdiscal from=EW0249037545G.cub to=EW0249037545G.lev1.cub
mdiscal from=EW0249037688G.cub to=EW0249037688G.lev1.cub
mdiscal from=EW0249066345G.cub to=EW0249066345G.lev1.cub
mdiscal from=EW0249037671G.cub to=EW0249037671G.lev1cub
mdiscal from=EW0249037562G.cub to=EW0249037562G.lev1.cub
```

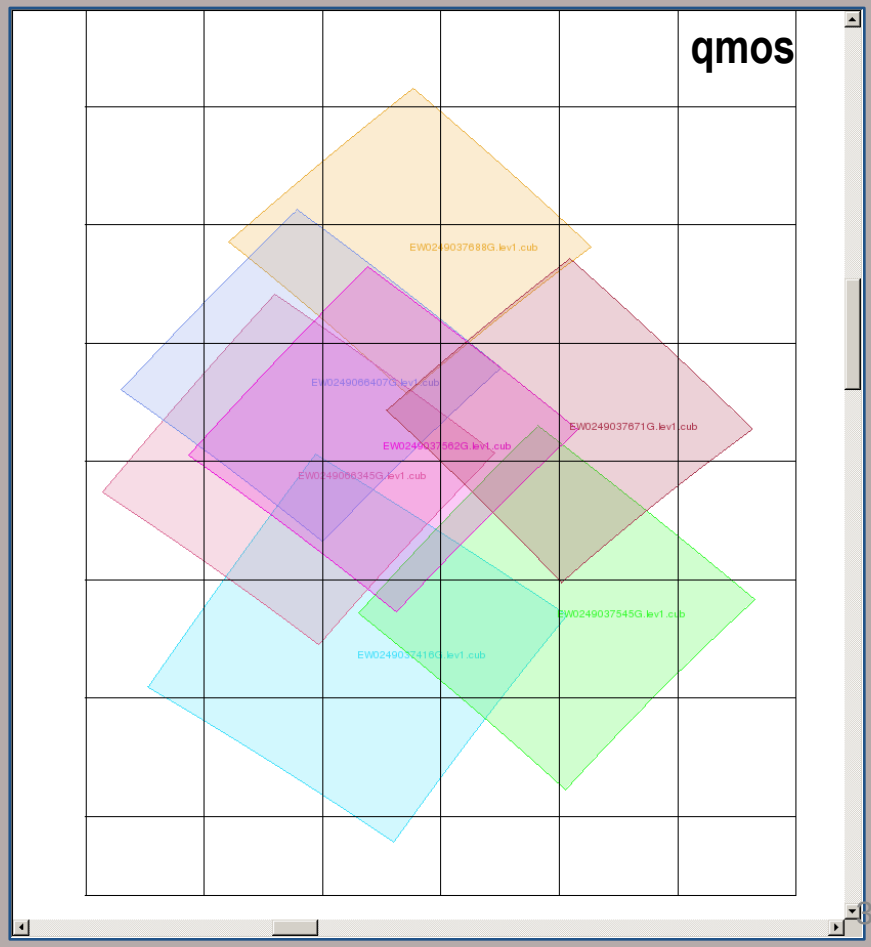


Specialized Geometric Tools

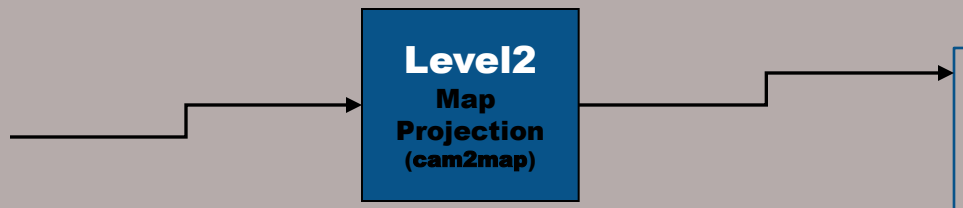
Geometric/ Photogrammetric Information (camstats,caminfo, campt,phocube, footprintinit)

```
camstats from=EW0249037416G.lev1.cub attach=true
footprintinit from=EW0249037416G.lev1.cub
camstats from=EW0249066407G.lev1.cub attach=true
footprintinit from=EW0249066407G.lev1.cub
camstats from=EW0249037545G.lev1.cub attach=true
footprintinit from=EW0249037545G.lev1.cub
camstats from=EW0249037688G.lev1.cub attach=true
footprintinit from=EW0249037688G.lev1.cub
camstats from=EW0249066345G.lev1.cub attach=true
footprintinit from=EW0249066345G.lev1.cub
camstats from=EW0249037671G.lev1.cub attach=true
footprintinit from=EW0249037671G.lev1.cub
camstats from=EW0249037562G.lev1.cub attach=true
footprintinit from=EW0249037562G.lev1.cub
```

Required applications for qmos



Generating Map Projection Specifications



Contents of *equi.map*

```
Group = Mapping
  ProjectionName          = EQUIRECTANGULAR
  TargetName              = Mercury
  EquatorialRadius        = 2440000.0 <meters>
  PolarRadius             = 2440000.0 <meters>
  LatitudeType            = Planetocentric
  LongitudeDirection      = PositiveEast
  LongitudeDomain         = 360
  PixelResolution         = 262.0 <meters/pixel>
  Scale                   = 163.0 <pixels/degree>
  MinPixelResolution      = 226.92996378504 <meters>
  MaxPixelResolution      = 296.53943218364 <meters>
  CenterLongitude         = 120.0
  CenterLatitude          = 27.0
  MinimumLatitude         = 17.0
  MaximumLatitude         = 37.0
  MinimumLongitude        = 111.0
  MaximumLongitude        = 128.0
End_Group
```

Create a PVL file to define map

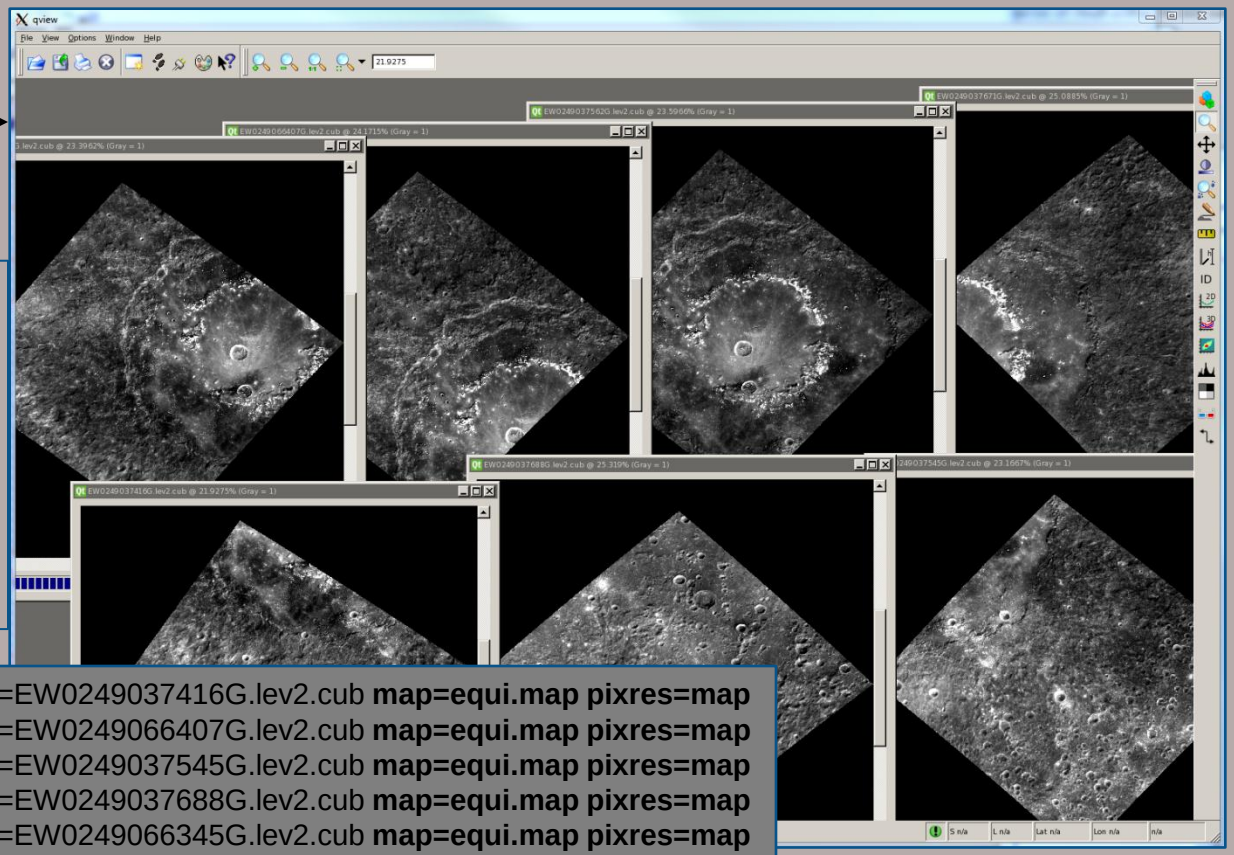
*For a mosaic, it is required that all input files are map projected to the exact same pixel resolution and center latitude and longitude

- **maptemplate**
- **mosrange**
- **Existing map projected ISIS image/mosaic cube**
- **Manual edit**

```
ls *.lev1.cub > lev1.lis
mosrange fromlist=lev1.lis to=equi.map projection=EQUIRECTANGULAR precision=0
```

Map Projection

Level2
Map Projection
(cam2map)



- Pixel resolution is the same for all images (*pixres=map*)
- Latitude/longitude range varies image-to-image (*defaultrange=minimize)*
- CenterLatitude and CenterLongitude must exist in map file
- Result of **mosrange** defines the map projection specifications (*map=equi.map*)

cam2map from=EW0249037416G.lev1.cub to=EW0249037416G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249066407G.lev1.cub to=EW0249066407G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249037545G.lev1.cub to=EW0249037545G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249037688G.lev1.cub to=EW0249037688G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249066345G.lev1.cub to=EW0249066345G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249037671G.lev1.cub to=EW0249037671G.lev2.cub **map=equi.map pixres=map**
cam2map from=EW0249037562G.lev1.cub to=EW0249037562G.lev2.cub **map=equi.map pixres=map**

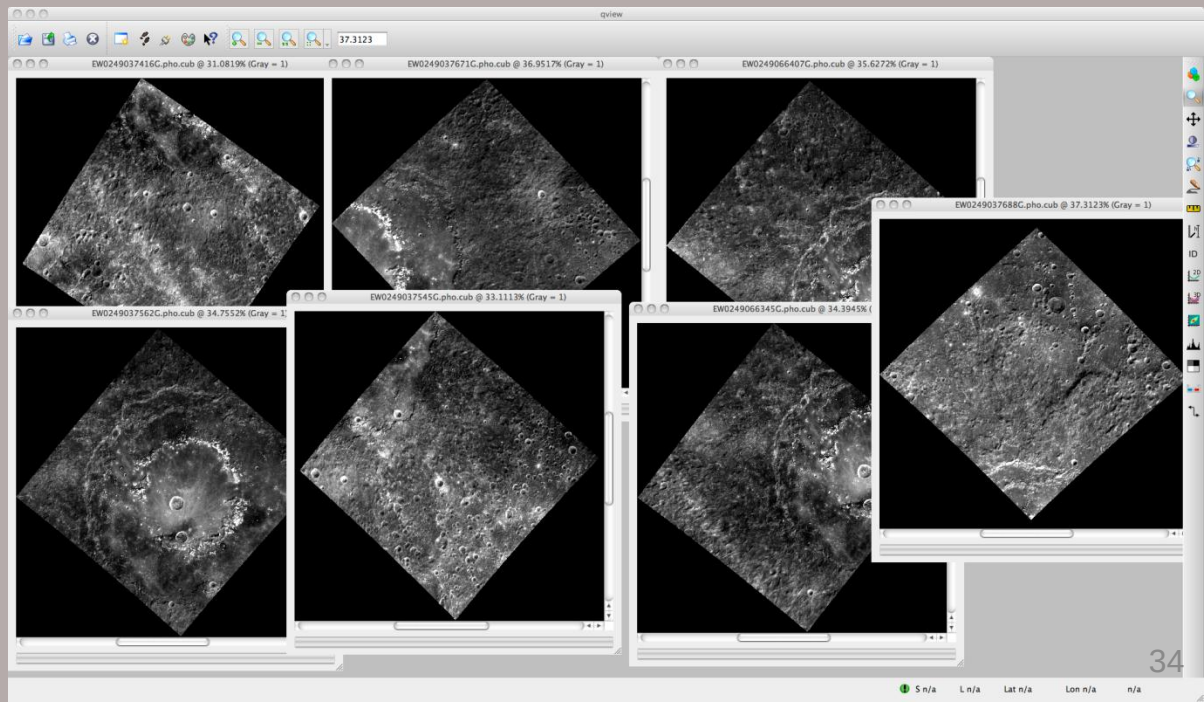
Photometric Correction

Level 3
Photometry
(photomet)

General Form (batch processing mode):
photomet –batchlist=basename.lis from=\$1.lev2.cub to=\$1.pho.cub
phtname=hapkehen theta=17.76662946 wh=0.278080114
hg1=0.227774899 hg2=0.714203968 hh=0.075 b0=2.3
zerob0standard=false **normname=albedo**
incref=30.0 incmat=0.0 thresh=10e30 albedo=1.0

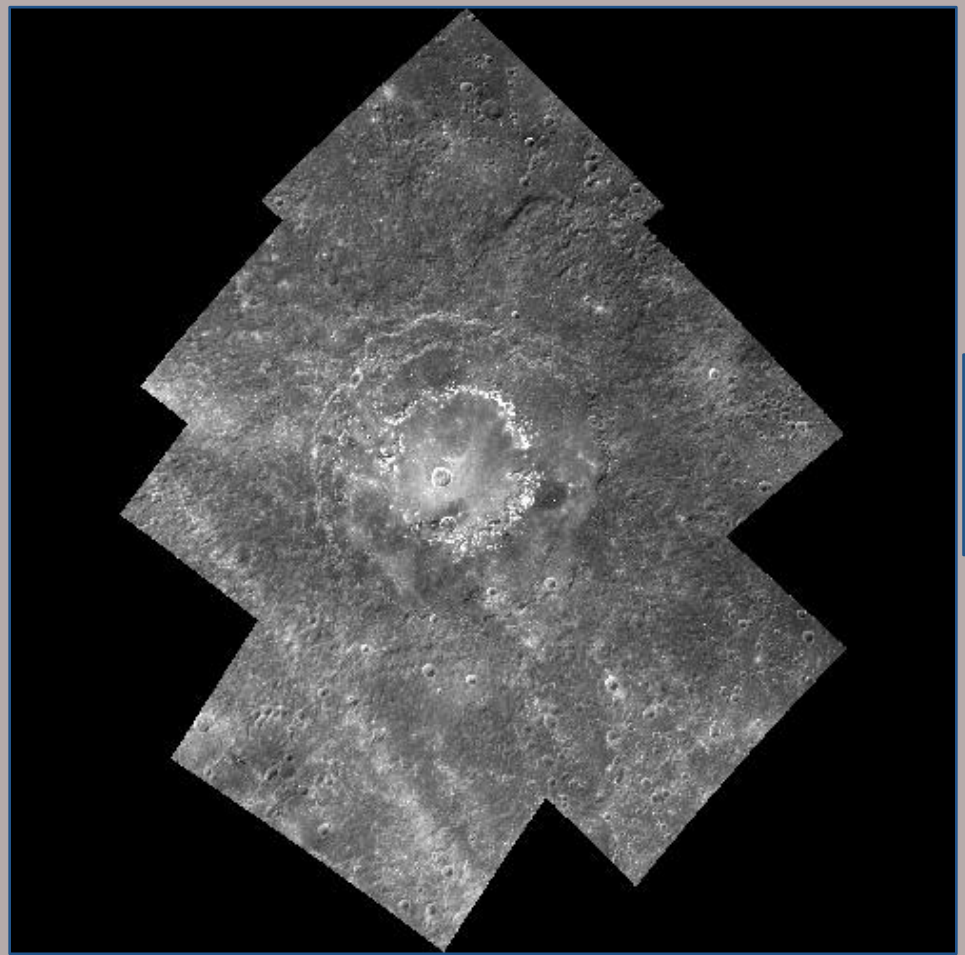
Apply a photometric correction on all projected ".lev2.cub's" using values established for the "G" filter

NOTE: MDIS NAC and WAC-G use the same parameters as they are very close in wavelength



Assemble Monochrome Mosaic

Raditladi Basin

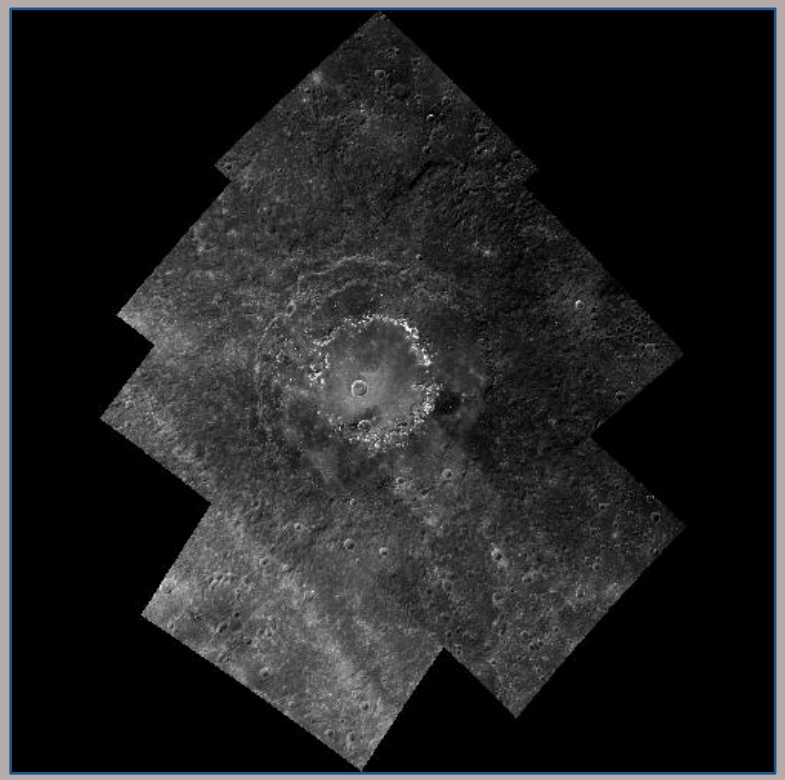


Level 4
Mosaic
(automos mapmos)

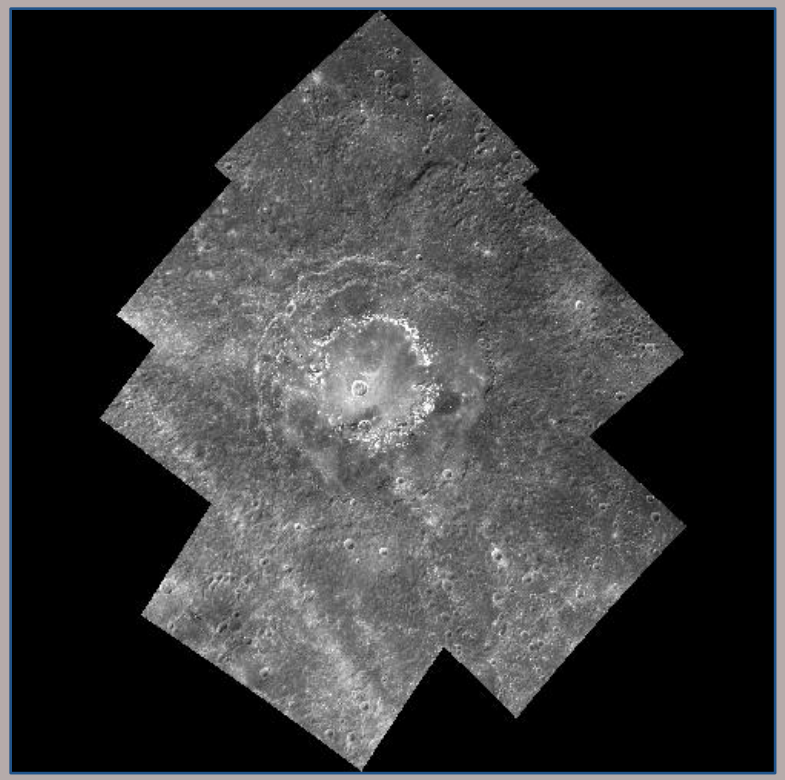
```
ls *.pho.cub > pho.lis

automos  fromlist=pho.lis
   mosaic=RaditladiBasin.cub
```

Importance of Photometric Correction

No Photometric Correction



After Photometric Correction



Creating Regional Color Mosaics

- Objective: Use ISIS3 to create a 3-color MDIS mosaic of Raditladi Basin
  - Uses some of the WAC-G images from monochrome mosaic
  - Why? These images happen to be part of the "Three Color" image campaign
    - Determined by OBSERVATION_TYPE keyword in PDS labels
    - Means there is a WAC-F and WAC-I accompanying image for each WAC-G
- Processing steps
  - Use same monochrome processing steps up through Photometric Correction
  - Additional processing handles color registration within color sets
    - Coregistration of images to one another within each color set
    - Stack coregistered color set images into wavelength ordered single cube
    - Trim excess around edges within sets for better seamless presentation
    - Assemble mosaic
- See the RaditladiBasin_3Color demo

Identify Color Sets

Selected PDS-EDR Images [2012_176]

Color Set1:	EW0249037562**G**.IMG	EW0249037566**F**.IMG	EW0249037570**I**.IMG
Color Set2:	EW0249066345G.IMG	EW0249066349F.IMG	EW0249066353I.IMG
Color Set3:	EW0249066407G.IMG	EW0249066411F.IMG	EW0249066415I.IMG
Color Set4:	EW0249037688G.IMG	EW0249037692F.IMG	EW0249037696I.IMG
Color Set5:	EW0249037617G.IMG	EW0249037621F.IMG	EW0249037625I.IMG

http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_1001/DATA/2012_176/

Stacking Order for Color Mosaic (B,G,R)

EW0211111676F.IMG	Center = 433.2 <NM>
EW0211111682G.IMG	Center = 748.7 <NM>
EW0211111674I.IMG	Center = 996.2 <NM>

Common Image Processing

```
ls *.IMG | sed 's/\.IMG//' > basename.lis

mdis2isis -batchlist=basename.lis from=\$1.IMG to=\$1.cub

spiceinit -batchlist=basename.lis from=\$1.cub

mdiscal -batchlist=basename.lis from=\$1.cub to=\$1.lev1.cub

camstats -batchlist=basename.lis from=\$1.lev1.cub attach=true linc=10 sinc=10

footprintinit -batchlist=basename.lis from=\$1.lev1.cub

ls *G.lev1.cub > G_lev1.lis
mosrange fromlist=G_lev1.lis to=equi.map projection=equirectangular precision=0

ls *G.lev1.cub | sed 's/\.lev1\.cub//' > Gonly_basename.lis

cam2map -batchlist=Gonly_basename.lis from=\$1.lev1.cub to=\$1.lev2.cub map=equi.map pixres=map

cam2map -batchlist=colorsets.lis from=\$2.lev1.cub to=\$2.lev2.cub map=\$1.lev2.cub matchmap=true

cam2map -batchlist=colorsets.lis from=\$3.lev1.cub to=\$3.lev2.cub map=\$1.lev2.cub matchmap=true
```

Level 0
Import / SPICE
(mdis2isis,spiceinit)

Level 1
Radiometric Calibration
(mdiscal)

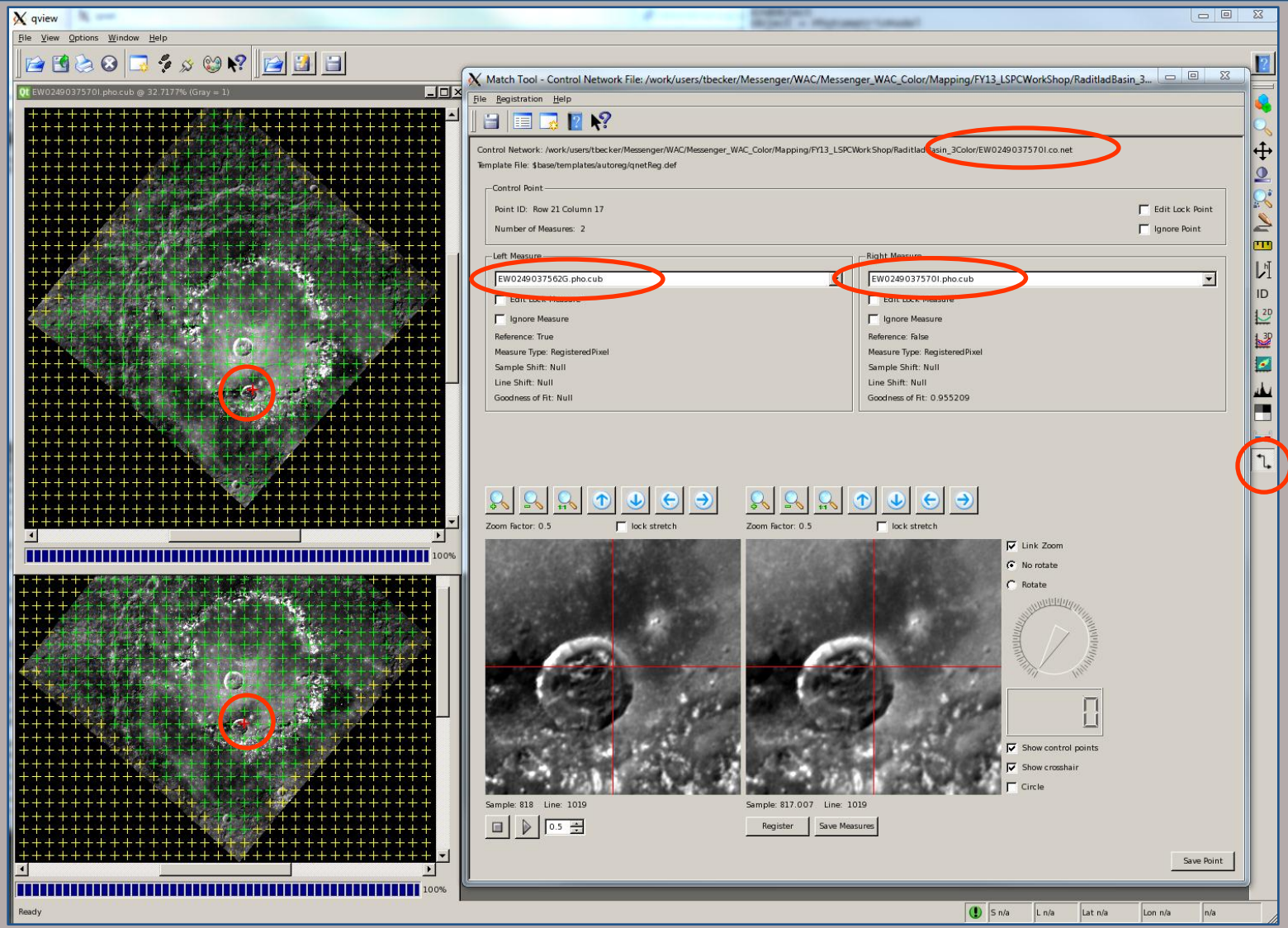
Geometric/ Photogrammetric Information
(camstats,caminfo, campt,phocube, footprintinit)

Level2
Map Projection
(mosrange, cam2map)

Color Photometry Correction

```
# Photometry correction with coefficient values for "G"
photomet -batchlist=colorsets.lis from=\$1.lev2.cub to=\$1.pho.cub phtname=hapkehen \
        theta=17.76662946 wh=0.278080114 hg1=0.227774899 hg2=0.714203968 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 incmat=0.0 thresh=10e30 albedo=1.0

# Photometry correction with coefficient values for "F"
photomet -batchlist=colorsets.lis from=\$2.lev2.cub to=\$2.pho.cub phtname=hapkehen \
        theta=12.07775431 wh=0.153713769 hg1=0.221313433 hg2=0.887633784 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 incmat=0.0 thresh=10e30 albedo=1.0

# Photometry correction with coefficient values for "I"
photomet -batchlist=colorsets.lis from=\$3.lev2.cub to=\$3.pho.cub phtname=hapkehen \
        theta=18.41686847 wh=0.35324478 hg1=0.276538744 hg2=0.613700193 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 incmat=0.0 thresh=10e30 albedo=1.0

#-----------------------------------------------------------------------------------------------------------------------------------------------------------
# Pattern Match Control Points and Rubber Sheet the F-filter and I-filter to the G Filter for each color set
# See important discussion on pattern matching at http://isis.astrogeology.usgs.gov/documents/PatternMatch/PatternMatch.html
coreg -batchlist=colorsets.lis from=\$2.pho.cub match=\$1.pho.cub to=\$2.co.cub deffile=coreg.def onet=\$2.co.net \
        transform=warp degree=2 interp=bilinear rows=32 columns=32

coreg -batchlist=colorsets.lis from=\$3.pho.cub match=\$1.pho.cub to=\$3.co.cub deffile=coreg.def onet=\$3.co.net \
        transform=warp degree=2 interp=bilinear rows=32 columns=32
```

Level 3
Photometry
(photomet)



qview
MatchTool

Displays of a **coreg** network of registered control points between an "I" and "G" images

Multiband Processing

```
# Stack sets into a multi-band cube
cubeit from=set1.lis to=set1.cub
bandtrim from=set1.cub to=set1_tr.cub

cubeit from=set2.lis to=set2.cub
bandtrim from=set2.cub to=set2_tr.cub

cubeit from=set3.lis to=set3.cub
bandtrim from=set3.cub to=set3_tr.cub

cubeit from=set4.lis to=set4.cub
bandtrim from=set4.cub to=set4_tr.cub

cubeit from=set5.lis to=set5.cub
bandtrim from=set5.cub to=set5_tr.cub

ls *set*tr.cub > color_mos.lis
automos fromlist=color_mos.lis mosaic=RaditladiBasin_rgb.cub
```

EW0249037566F.co.cub
EW0249037562G.pho.cub
EW0249037570I.co.cub

EW0249066349F.co.cub
EW0249066345G.pho.cub
EW0249066353I.co.cub

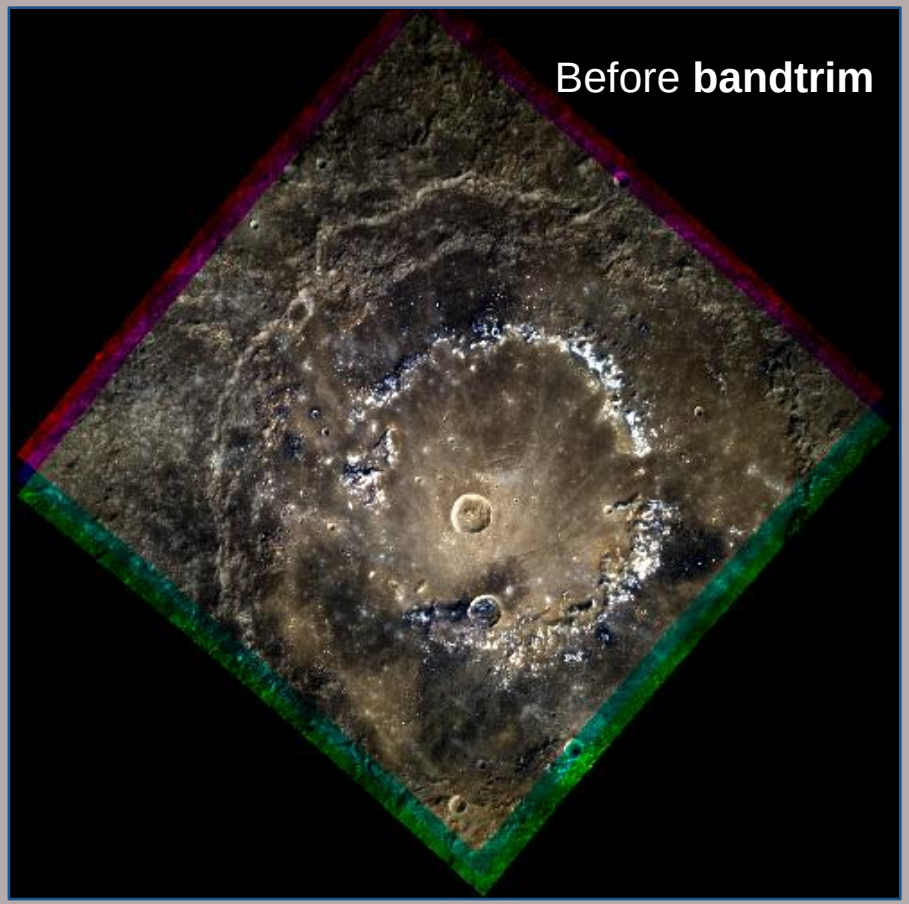
EW0249066411F.co.cub
EW0249066407G.pho.cub
EW0249066415I.co.cub

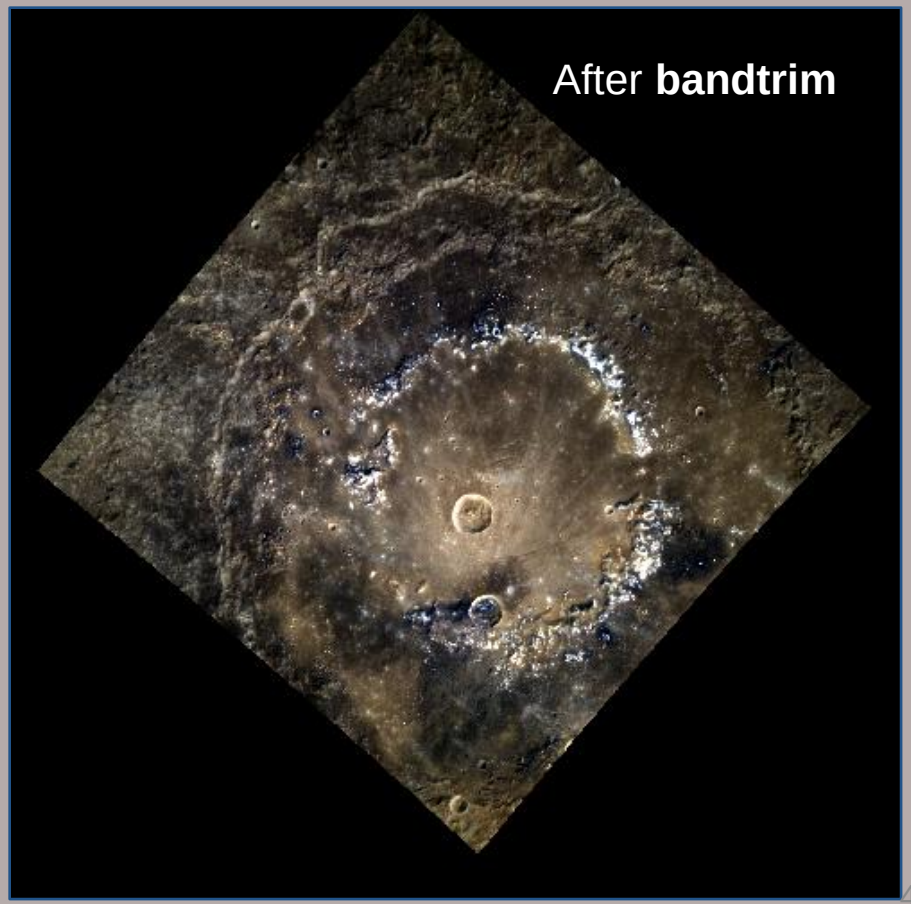
EW0249037692F.co.cub
EW0249037688G.pho.cub
EW0249037696I.co.cub

EW0249037621F.co.cub
EW0249037617G.pho.cub
EW0249037625I.co.cub

Level 4
Mosaic
(automos mapmos)

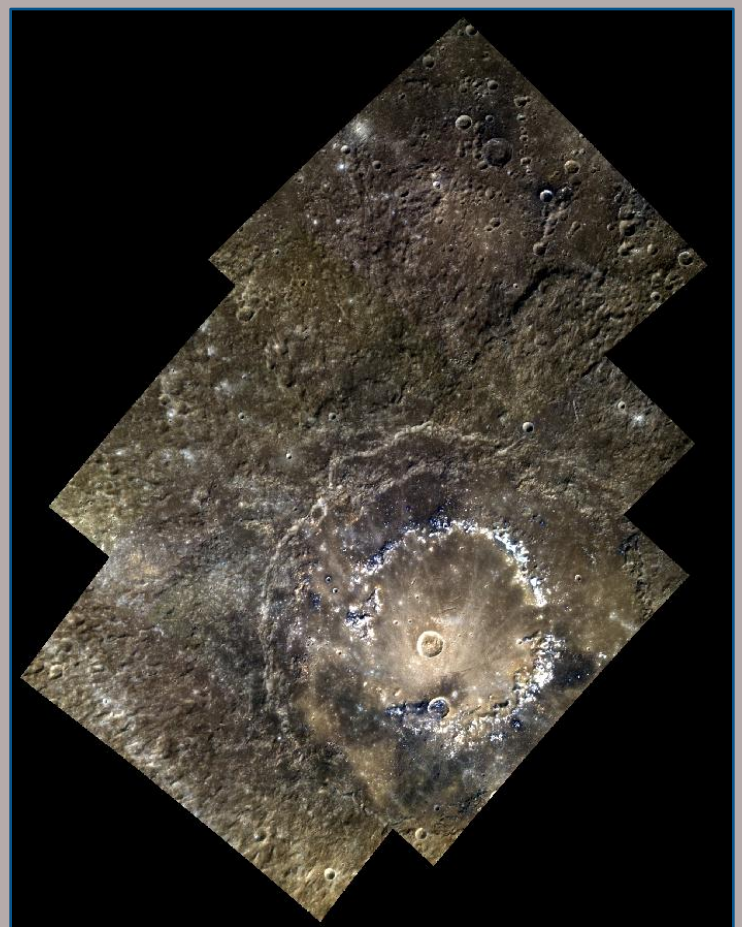
Rationale for Trimming Image Boundaries





Assemble 3-Color Mosaic

Raditladi Basin In Color



Level 4
Mosaic
(automos mapmos)

Three-Color Observation Filters

I (Red)	Center = 996.2 <NM>
G (Grn)	Center = 748.7 <NM>
F (Blu)	Center = 433.2 <NM>

Creating 8-Color Sets

- Objective: Use ISIS3 to create an 8-color MDIS set of Praxiteles
  - Images are part of the “Color” image campaign
    - Determined by OBSERVATION_TYPE keyword in PDS labels
    - Sets are comprised of one each of WAC-F, WAC-C, WAC-D, WAC-E, WAC-G, WAC-L, WAC-J and WAC-I (in wavelength order) filters
- Processing steps
  - Use same processing steps in 3-Color up to stacking
  - Eight filters requires specialized order of images to retaining increasing wavelength order in color image cube
- See the Praxiteles_8Color demo

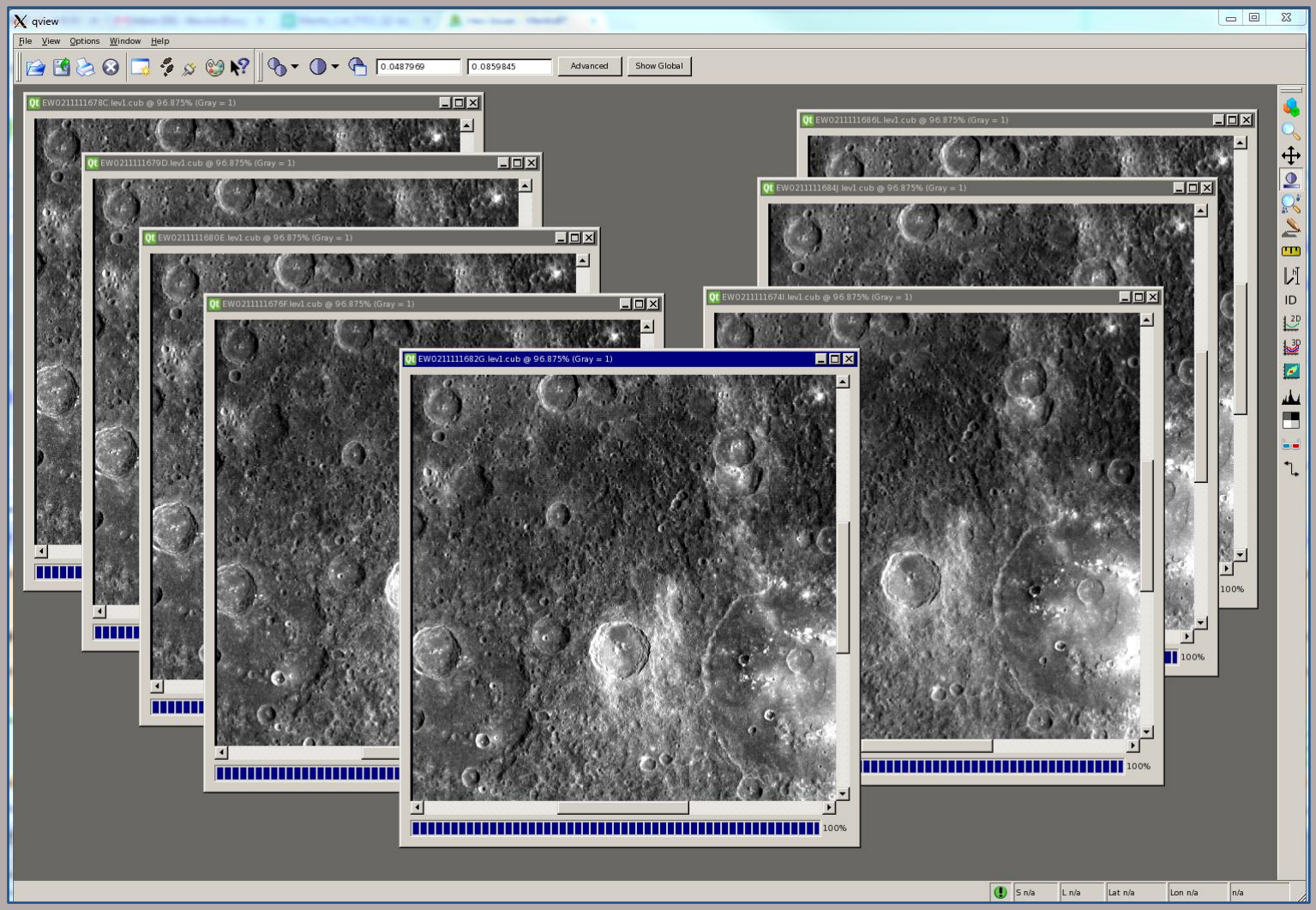
Create 8-Color MultiSpectral Map

Area of Interest: Praxiteles

MinimumLatitude	= 24.0
MaximumLatitude	= 39.0
MinimumLongitude	= 290.0
MaximumLongitude	= 307.0

Selected PDS-EDR Images [2011_102]

EW0211111676F.IMG	Center = 433.2 <NM>
EW0211111678C.IMG	Center = 479.9 <NM>
EW0211111679D.IMG	Center = 558.9 <NM>
EW0211111680E.IMG	Center = 628.8 <NM>
EW0211111682G.IMG	Center = 748.7 <NM>
EW0211111686L.IMG	Center = 828.4 <NM>
EW0211111684J.IMG	Center = 898.8 <NM>
EW0211111674I.IMG	Center = 996.2 <NM>


qview
File View Options Window Help
0.0487969
0.0859845
Advanced
Show Global
EW0211111678C.lev1.cub @ 96.875% (Gray = 1)
EW0211111679D.lev1.cub @ 96.875% (Gray = 1)
EW0211111680E.lev1.cub @ 96.875% (Gray = 1)
EW0211111676F.lev1.cub @ 96.875% (Gray = 1)
EW0211111682G.lev1.cub @ 96.875% (Gray = 1)
EW0211111686L.lev1.cub @ 96.875% (Gray = 1)
EW0211111684J.lev1.cub @ 96.875% (Gray = 1)
EW0211111674I.lev1.cub @ 96.875% (Gray = 1)
100%
S n/a
L n/a
Lat n/a
Lon n/a
n/a

Standard Processing

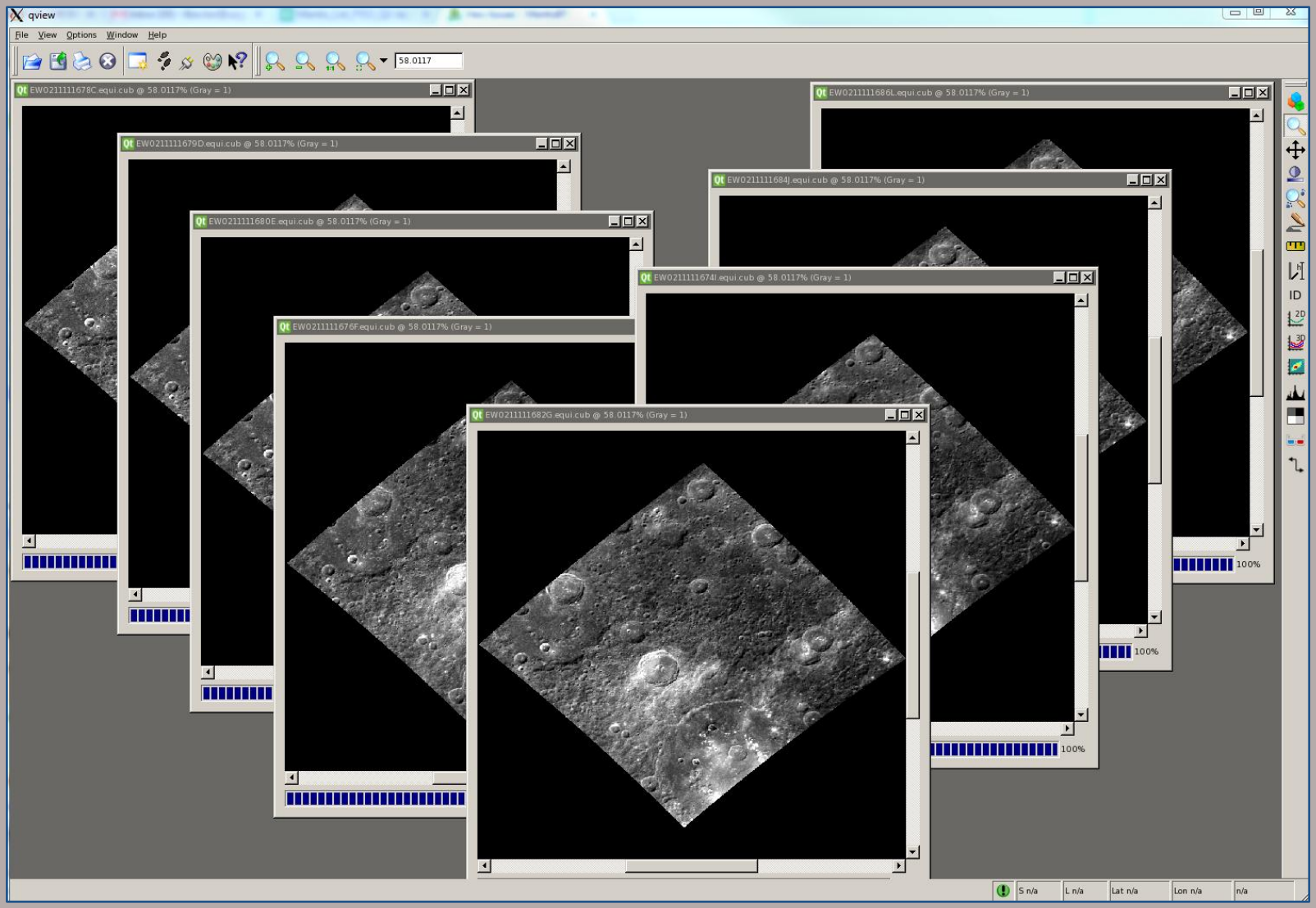
```
ls *IMG | sed 's/\.IMG//' > basename.lis

mdis2isis -batchlist=basename.lis from=\$1.IMG to=\$1.cub

spiceinit -batchlist=basename.lis from=\$1.cub

mdiscal -batchlist=basename.lis from=\$1.cub to=\$1.lev1.cub

ls *.lev1.cub > lev1.lis

mosrange fromlist=lev1.lis to=equi.map projection=equirectangular precision=0

cam2map -batchlist=basename.lis from=\$1.lev1.cub to=\$1.lev2.cub \
            map=equi.map pixres=map defaultrange=map

# All photometric parameters for 8 WAC filters
photomet  from=EW0211111678C.lev2.cub to=EW0211111678C.pho.cub phtname=hapkehen \
        theta=13.82780392 wh=0.182212955 hg1=0.212533357 hg2=0.856934992 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0  thresh=10e30 albedo=1.0

photomet  from=EW0211111679D.lev2.cub to=EW0211111679D.pho.cub phtname=hapkehen \
        theta=15.78892162 wh=0.215984749 hg1=0.206649235 hg2=0.811417942 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0
```

8-Color Photometry

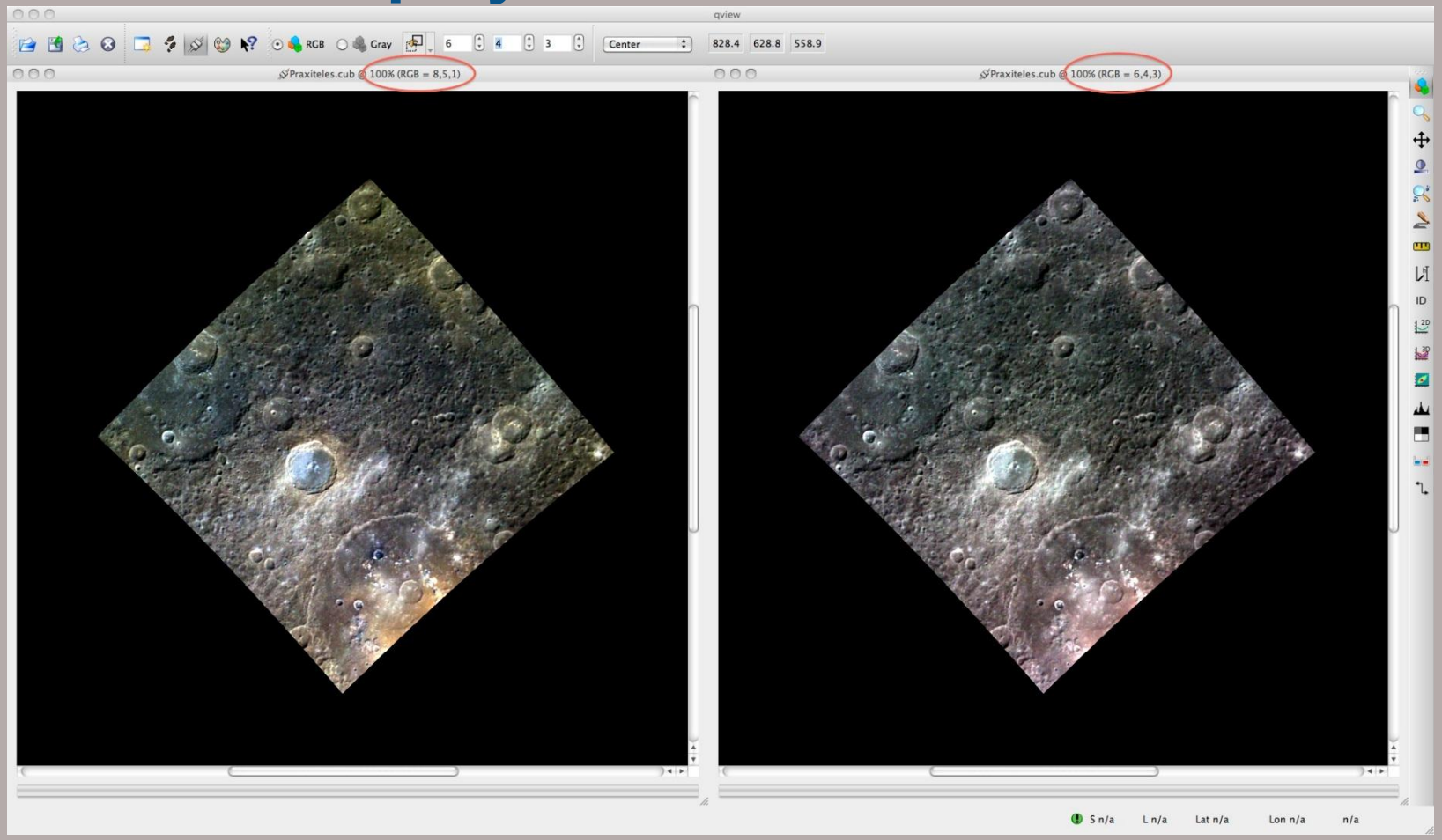
```
photomet  from=EW0211111680E.lev2.cub to=EW0211111680E.pho.cub phtname=hapkehen \
        theta=15.78892162 wh=0.215984749 hg1=0.206649235 hg2=0.811417942 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0

photomet  from=EW0211111676F.lev2.cub to=EW0211111676F.pho.cub phtname=hapkehen \
        theta=12.07775431 wh=0.153713769 hg1=0.221313433 hg2=0.887633784 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0

photomet  from=EW0211111682G.lev2.cub to=EW0211111682G.pho.cub phtname=hapkehen \
        theta=17.76662946 wh=0.278080114 hg1=0.227774899 hg2=0.714203968 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0

photomet  from=EW0211111674I.lev2.cub to=EW0211111674I.pho.cub phtname=hapkehen \
        theta=18.41686847 wh=0.35324478 hg1=0.276538744 hg2=0.613700193 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0
```

8-Color Set Registration

```
photomet  from=EW0211111684J.lev2.cub to=EW0211111684J.pho.cub phtname=hapkehen \
        theta=18.07191127 wh=0.32654443 hg1=0.261680383 hg2=0.650146228 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0

photomet  from=EW0211111686L.lev2.cub to=EW0211111686L.pho.cub phtname=hapkehen \
        theta=17.96224797 wh=0.304047732 hg1=0.245886415 hg2=0.678657724 \
        hh=0.075 b0=2.3 zerob0standard=false \
        normname=albedo incref=30.0 thresh=10e30 albedo=1.0

ls *.pho.cub | sed 's/\.pho\.cub//' | grep -v G > no_g.lis

coreg -batchlist=no_g.lis from=\$1.pho.cub deffile=coreg.def \
     to=\$1.co.cub onet=\$1.co.cub match=EW0211111682G.pho.cub \
     transform=warp degree=2 interp=bilinear rows=32 columns=32

cubeit fromlist=color_order.lis to=Praxiteles_stack.cub

bandtrim from=Praxiteles_stack.cub to=Praxiteles.cub
```


qview
File View Options Window Help
58.0117
EW0211111678C.equi.cub @ 58.0117% (Gray = 1)
EW0211111679D.equi.cub @ 58.0117% (Gray = 1)
EW0211111680E.equi.cub @ 58.0117% (Gray = 1)
EW0211111676F.equi.cub @ 58.0117% (Gray = 1)
EW0211111682G.equi.cub @ 58.0117% (Gray = 1)
EW0211111674I.equi.cub @ 58.0117% (Gray = 1)
EW0211111684J.equi.cub @ 58.0117% (Gray = 1)
EW0211111686L.equi.cub @ 58.0117% (Gray = 1)
100%
100%
100%
S n/a
L n/a
Lat n/a
Lon n/a
n/a

3-Color Display Combinations



Processing MESSENGER PDS BDR and MDR Data Products

- These are higher level, map-projected global data products
- Intended to free users from having to derive their own products from raw (EDR) data
- Two map projected MESSENGER PDS data products
  - Map Projected Basemap RDR (BDR)
    - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_4001/
  - Map Projected Multispectral RDR (MDR)
    - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_5001/
- ISIS3 provides a very basic PDS ingestion application called **pds2isis**
  - Imports data and some label keywords
  - Translates most all PDS map projection keywords and properties
  - After ingestion, ISIS3 can be used to produce maps of different projection types and resolutions (**map2map**)

Processing MESSENGER PDS BDR and MDR Data Products

- MESSENGER Monochrome Basemap Mosaic (BDR)
  - Product Highlights
    - Coverage is very nearly global
      - Comprised of MDIS NAC and WAC-G images
    - Map projected to Equirectangular projection
      - North and South poles are in Polar Stereographic
    - Fifty-four non-overlapping *tiles* in fifteen quadrangles
    - Resolution: ~166 <meters/pixel> (265 <pixels/degree>)
    - Contains five additional reference data *backplanes*
      - ObservationId, BDR stacking metric (mosaic ordering), Solar Incidence angle, Emission angle and Phase angle
- RDR SIS contains complete description of BDR products
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_4001/DOCUMENT/MDIS_CDR_RDRSIS.PDF
- Important reference *prior* to usage and analysis with this data!
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_4001/CATALOG/MDIS_BDR_DS.CAT

Processing MESSENGER PDS BDR and MDR Data Products

- MESSENGER Multispectral (Color) Mosaic (MDR)
  - Product Highlights
    - Coverage is very nearly global
      - Comprised of eight WAC filter images (8 bands of color data)
    - Map projected to Equirectangular projection
      - North and South poles are in Polar Stereographic
    - Fifty-four non-overlapping *tiles* in fifteen quadrangles
    - Resolution: ~665 <meters/pixel> (64 <pixels/degree>)
    - Contains five additional reference data *backplanes*
      - ObservationId, BDR stacking metric (mosaic ordering), Solar Incidence angle, Emission angle and Phase angle
- RDR SIS contains complete description of MDR products
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_5001/DOCUMENT/MDIS_CDR_RDRSIS.PDF
- Important reference *prior* to usage and analysis with this data!
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_5001/CATALOG/MDIS_MDR_DS.CAT

Processing MESSENGER PDS BDR and MDR Data Products

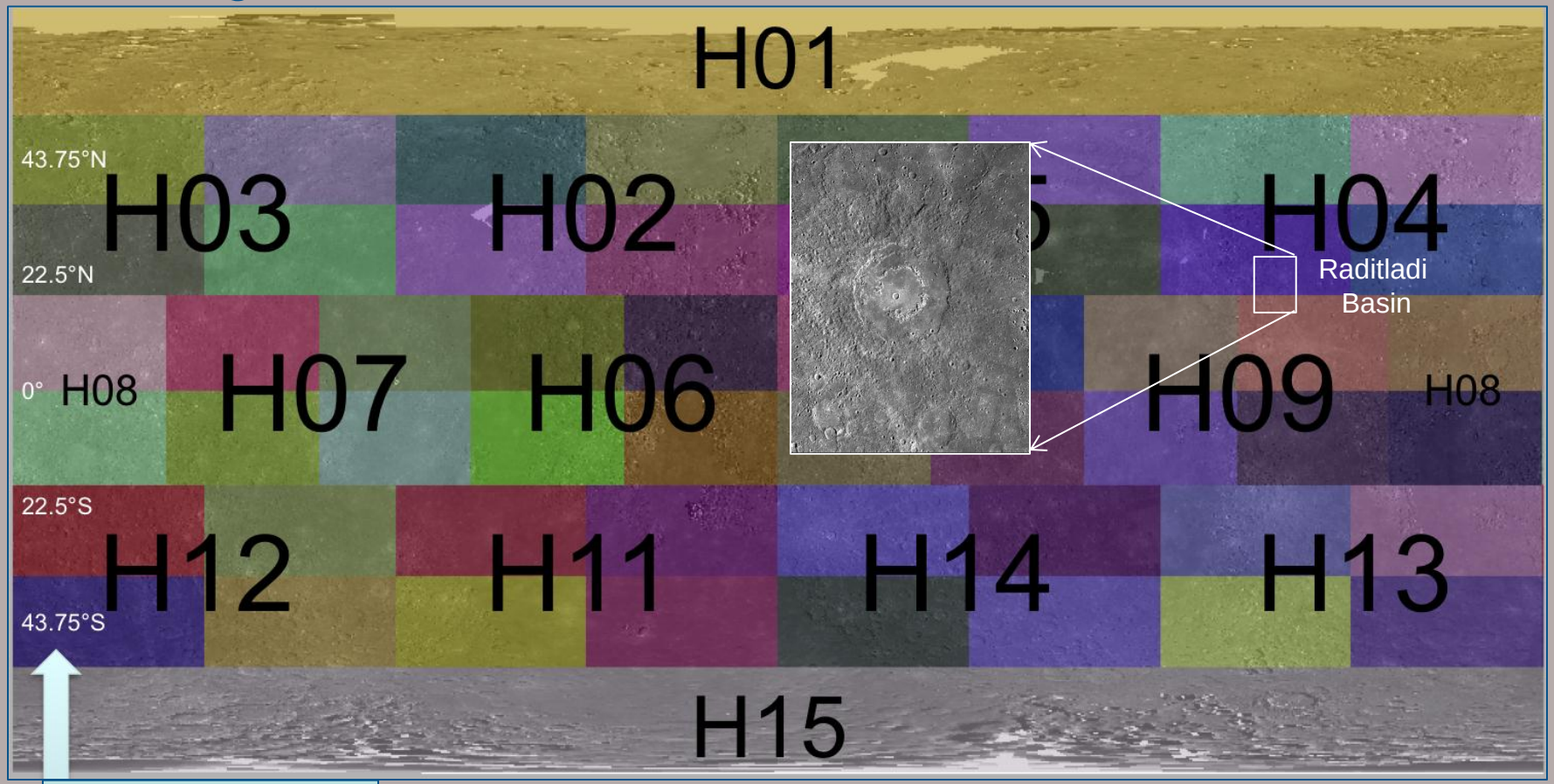
- Our objective is to extract the same region around Raditladi as our individual monochrome and color and compare the results
- Some things to consider:
  - Resolution Differences
    - EDR monochrome product has a different map pixel resolution (262 <meters/pixel>) than the BDR (166 <meters/pixel>
  - Map Projection central latitude and longitude differ
    - In fact, all quad tiles have a different center latitude and center longitude
  - BDR and MDR are both projected into the 180° longitude domain – our EDR products are in the 360° longitude domain
    - The Raditladi Basin regional mosaic is between 17° N and 36° N latitude, 111° and 128° positive east longitude range
- We will make a BDR monochrome and a MDR color map mosaic for direct comparison with our EDR derived monochrome and 3-color mosaics, respectively
- Must reproject all BDR/MDR tiles that contain common areas of coverage to the same map properties as our monochrome and color products

Processing MESSENGER PDS BDR and MDR Data Products

Quadrangle	Subdirectory name	Latitude (degrees)	Longitude (deg. east)
H-1 Borealis	H01	65 to 90	0 to 360
H-2 Victoria	H02	22.5 to 65	270 to 360
H-3 Shakespeare	H03	22.5 to 65	180 to 270
H-4 Liguria	H04	22.5 to 65	90 to 180
H-5 Apollonia	H05	22.5 to 65	0 to 90
H-6 Kuiper	H06	-22.5 to 22.5	288 to 360
H-7 Beethoven	H07	-22.5 to 22.5	216 to 288
H-8 Tolstoj	H08	-22.5 to 22.5	144 to 216
H-9 Solitudo Criophori	H09	-22.5 to 22.5	72 to 144
H-10 Pieria	H10	-22.5 to 22.5	0 to 72
H-11 Discovery	H11	-65 to -22.5	270 to 360
H-12 Michelangelo	H12	-65 to -22.5	180 to 270
H-13 Solitudo Persephones	H13	-65 to -22.5	90 to 180
H-14 Cyllene	H14	-65 to -22.5	0 to 90
H-15 Bach	H15	-90 to -65	0 to 360

Table shows tile boundaries for all Mercury Quadrangles
Longitude coordinates in positive east 360° domain – map products projected into 180° domain
Each tile contains four quadrants (NW, NE, SW and SE) excluding North and South poles (54 in all)

Processing MESSENGER PDS BDR and MDR Data Products



Latitude of true scale for equidistant cylindrical projection

MDIS BDR Monochrome Mosaic

- Objective: Use ISIS3 to create an MDIS monochrome mosaic from PDS BDRs of Raditladi Basin
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_4001/BDR/
  - Because each quadrant has a different CenterLatitude/CenterLongitude, you must reproject each to a common center coordinate to combine into map mosaic
- General Processing Steps
  - Determine quadrants in quads with ROI coverage (H04SW0, H09NE0)
  - Download images from PDS
  - Process using ISIS3
    - Import (**pds2isis**), map project (**map2map**) and mosaic (**automos**)
- See the MDIS_BDR demo

MDIS BDR Processing Sequence

- Provide detached label in *from* parameter in **pds2isis**
- Uses same map file created in RaditladiBasin demo
- *matchmap=true* ensures the same output dimensions
- A Perl script, **mdis_pds_proc**, provides a one line command to process a list of PDS BDR or MDR quadrant files or **pds2isis** converted ISIS cubes
- Use **--bdr** for monochrome option (extracts/projects first band only)

ls -1 MDIS_BDR_256PPD_*.LBL > Raditladi.lis
./mdis_pds_proc -v --bdr --list=Raditladi.lis –map=Raditladi.map --matchmap --mosaic=Raditladi_bdr.cub

```
#  Execution sequence generated by Perl script mdis_pds_proc
pds2isis from=MDIS_BDR_256PPD_H04SW0.LBL to=mdis_bdr_256ppd_h04sw0.pds.cub
map2map from=mdis_bdr_256ppd_h04sw0.pds.cub+1 to=mdis_bdr_256ppd_h04sw0.proj.cub map=Raditladi.map matchmap=true

pds2isis from=MDIS_BDR_256PPD_H09NE0.LBL to=mdis_bdr_256ppd_h09ne0.pds.cub
map2map from=mdis_bdr_256ppd_h09ne0.pds.cub+1 to=mdis_bdr_256ppd_h09ne0.proj.cub  map=Raditladi.map matchmap=true

ls -1 *.proj.cub > mosfiles.lis
automos fromlist=mosfiles.lis mosaic=Raditladi_bdr.cub matchbandbin=false priority=beneath
```

MDIS BDR Monochrome Mosaic

Raditladi Basin from PDS BDR

MDIS MDR 3-Color Mosaic

- Objective: Use ISIS3 to create an MDIS 3-Color (or 8-Color) mosaic from PDS MDRs of Raditladi Basin
  - http://pdsimage.wr.usgs.gov/Missions/MESSENGER/MSGRMDS_5001/MDR/
  - Like BDRs, each MDR quadrant has a different CenterLatitude/CenterLongitude
    - Reproject required to a common center coordinate to combine into map mosaic
- General Processing Steps
  - Same quadrants as BDR mosaic (H04SW0, H09NE0)
  - Download images from PDS
  - Process using ISIS3
    - Import (**pds2isis**), map project (**map2map**) and mosaic (**automos**)
- See the MDIS_MDR demo

MDIS MDR Processing Sequence

- Same process as BDR only choose color MDR options in **mdis_pds_proc**
- Use **--rgb** for 3-color, **--mdr** for 8-color mosaics

ls -1 MDIS_MDR_64PPD_*.LBL > Raditladi.lis
./mdis_pds_proc -v --rgb --list=Raditladi.lis –map=Raditladi.map --matchmap --mosaic=Raditladi_rgb.cub

```
#  Execution sequence generated by Perl script mdis_pds_proc
pds2isis from=MDIS_MDR_064PPD_H04SW0.LBL to=mdis_mdr_064ppd_h04sw0.pds.cub
map2map from=mdis_mdr_064ppd_h04sw0.pds.cub+1,5,8 to=mdis_mdr_064ppd_h04sw0.proj.cub \
        map=Raditladi.map matchmap=true

pdisis from=MDIS_MDR_064PPD_H09NE0.LBL to=mdis_mdr_064ppd_h09ne0.pds.cub
map2map from=mdis_mdr_064ppd_h09ne0.pds.cub+1,5,8 to=mdis_mdr_064ppd_h09ne0.proj.cub \
        map=Raditladi.map matchmap=true

ls -1 mdis_mdr_*.proj.cub > mosfiles.lis
automos fromlist=mosfiles.lis mosaic=Raditladi_rgb.cub matchbandbin=false priority=beneath
```

MDIS MDR 3-Color Mosaic

Raditladi Basin In 3-Color from PDS MDR

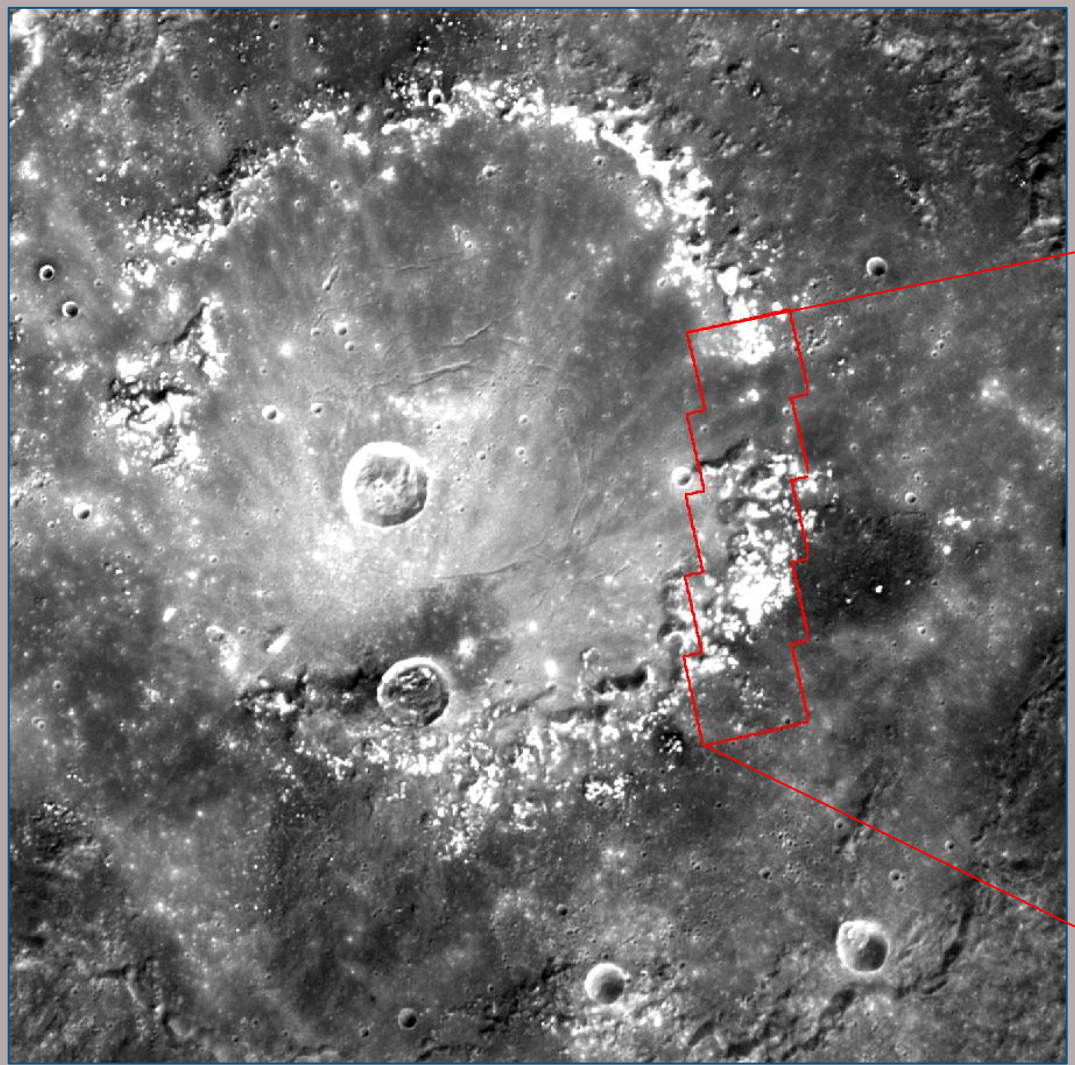
Three-Color Observation Filters

I (Red)	Center = 996.2 <NM>
G (Grn)	Center = 748.7 <NM>
F (Blu)	Center = 433.2 <NM>

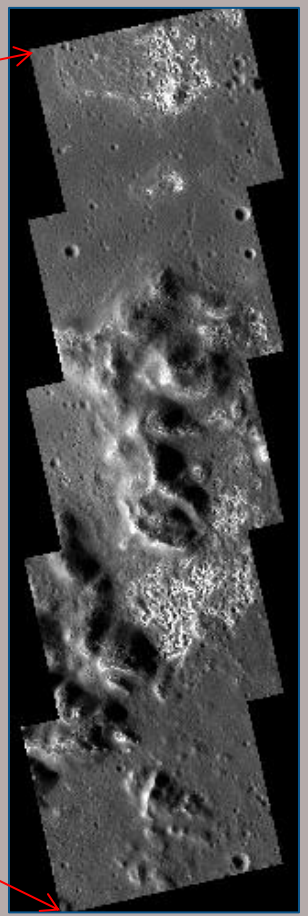
Considerations when Processing PDS BDRs and MDRs

- Use **pds2isis** to import all bands into an ISIS cube
- All quadrants must be reprojected to same center coordinate in order to merge into combined map product
- Special Perl script, **mdis_pds_proc**, provided to simplify PDS BDR and MDR processing
  - Script documentation contains additional information/help
- PDS BDR and MDR have additional information available for every pixel in backplanes
- Don't project "OBSERVATION ID" band with anything other than nearest neighbor (i.e., no interpolation)
  - Otherwise computes an average of surrounding OBSERVATION IDs – nonsense!
  - Project alone or exclude entirely
    - See **--bands** option in **mdis_pds_proc**
- The PDS BDR and MDR volumes are huge!
  - BDR is 69GB
  - MDR is 9GB

Raditladi Hollows



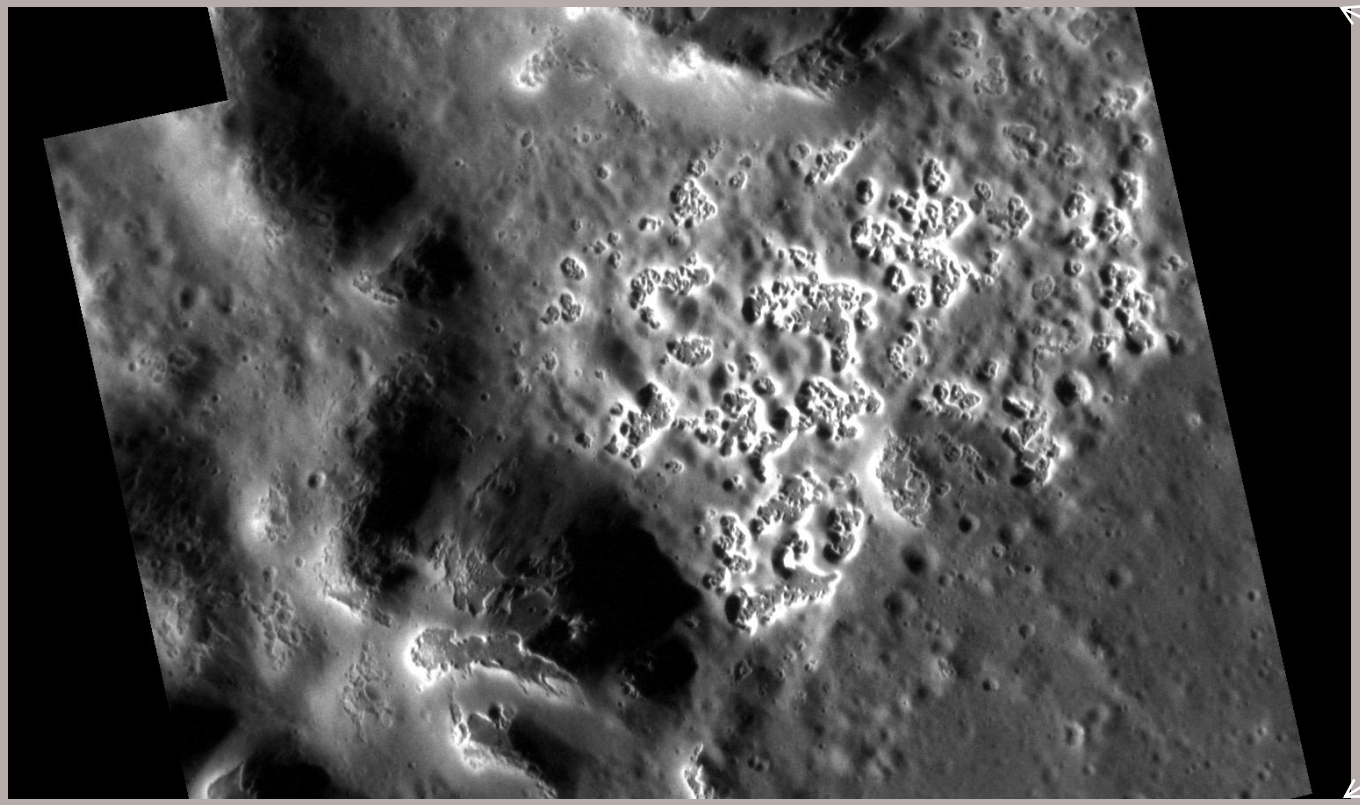
Finally, a Very High Resolution Mosaic

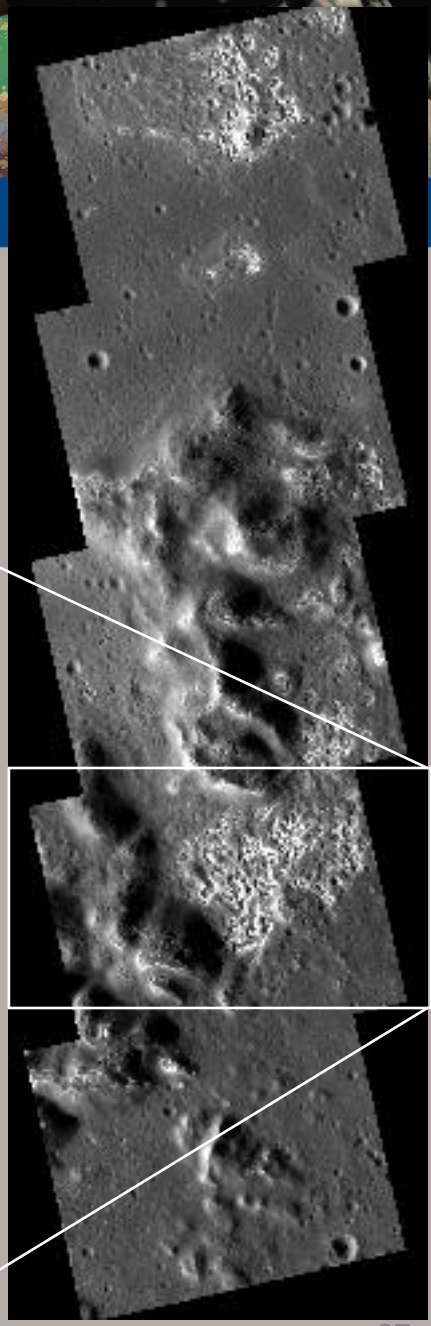


Very High Resolution NAC Mosaic

Raditladi Hollows

This MDIS NAC mosaic is ~17 <meters/pixel> and was acquired on 2011-08-04.





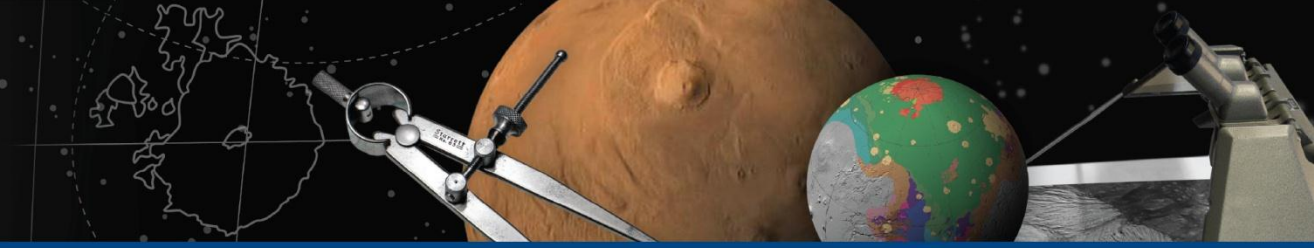
See the RaditladiHollows demo

Export and Application Support for ISIS3 Products

Trent Hare, thare@usgs.gov

Brief introduction to using data outside of ISIS.

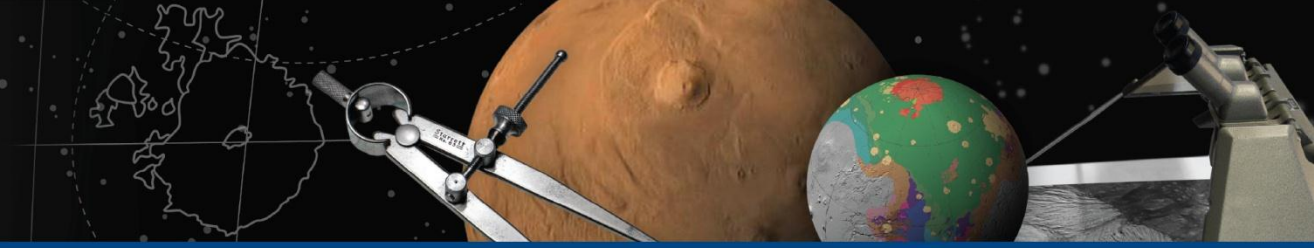
- For more visit Astrogeology's **MRCTR GIS Lab**



Exporting ISIS3 for Further Analysis

ISIS supports exporting images to several “graphical” formats using **isis2std**.

- PNG (8bit, 2GB)
- Jpeg (8bit, ~2GB)
- Tiff (8bit, 4GB)
- Jpeg2000 (8, 16bit, huge)

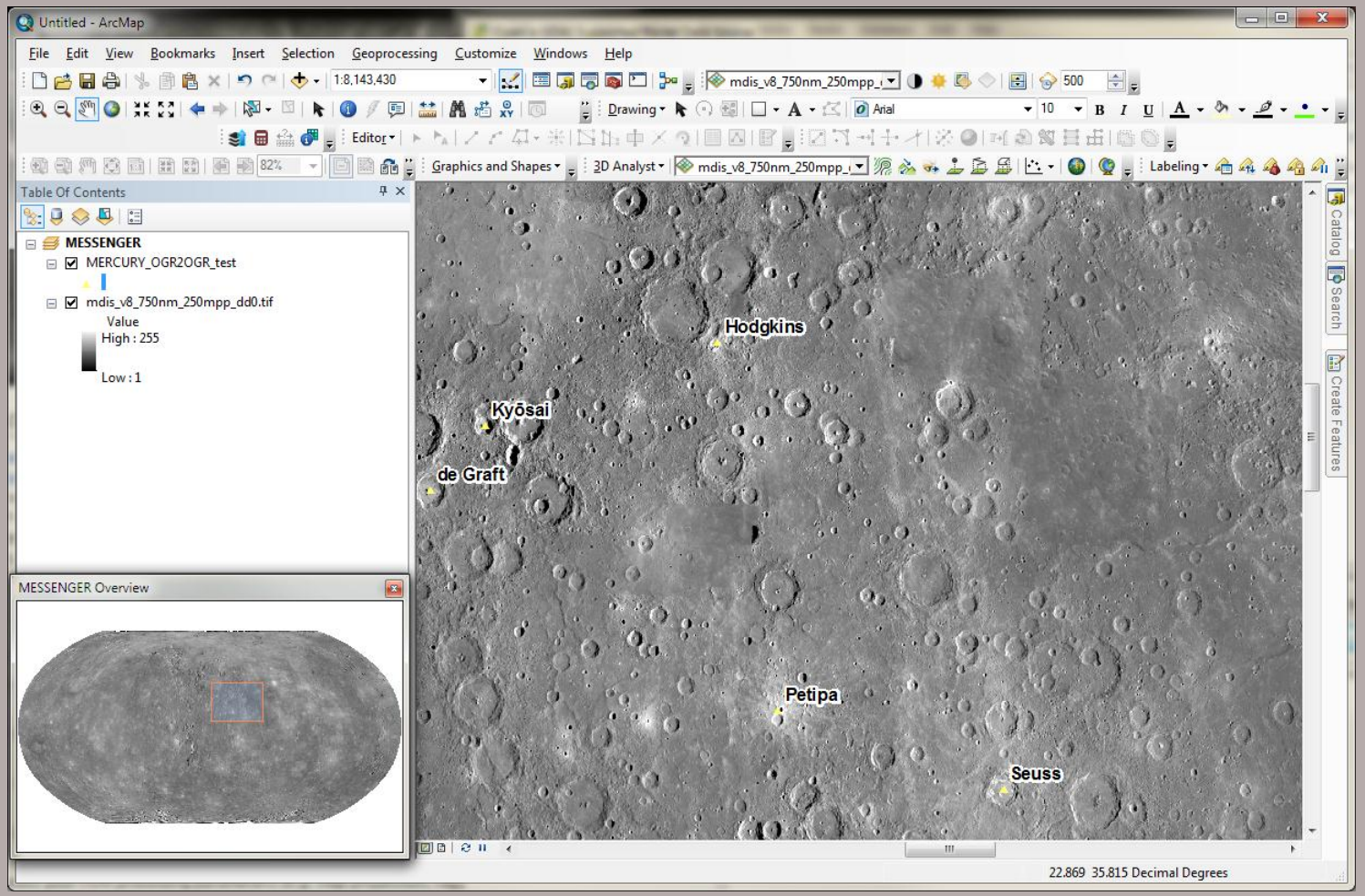


Exporting ISIS3 Data for Further Analysis

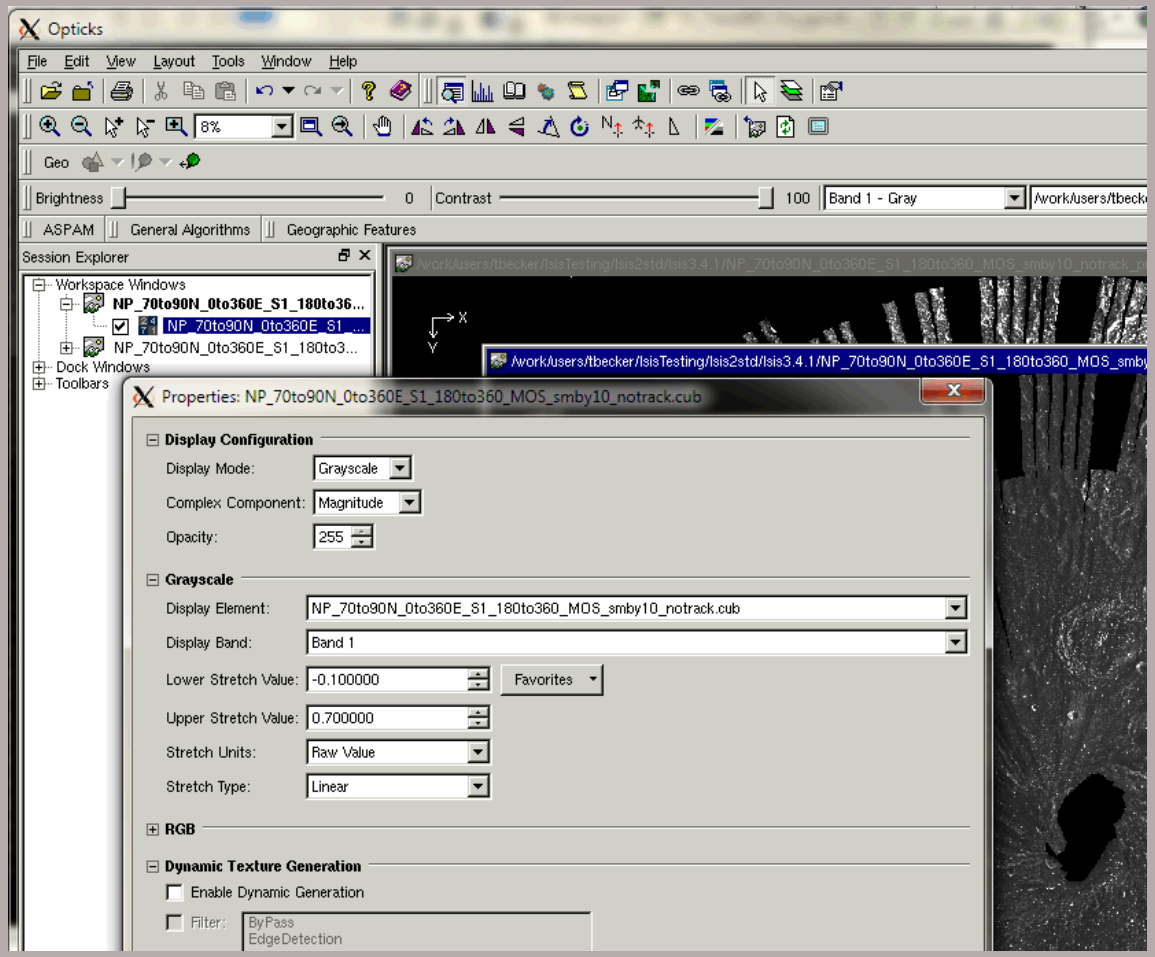
For higher bit types and “mapping” formats users can:

1. Use the ISIS3 format **directly** (apps that use GDAL)
   - ArcMap GIS
   - QGIS
   - Mirone (GMT, Mathlab)
   - Opticks
2. Convert to a standardized format (using GDAL tools)
   - GeoTiff (recommended, 8, 16,32bit, 2TB)
   - GeoJpeg2000 (8, 16bit, 2TB limit?)
   - ENVI (header w/ raw image, 8, 16, 32bit, limit?)
   - 100+ others

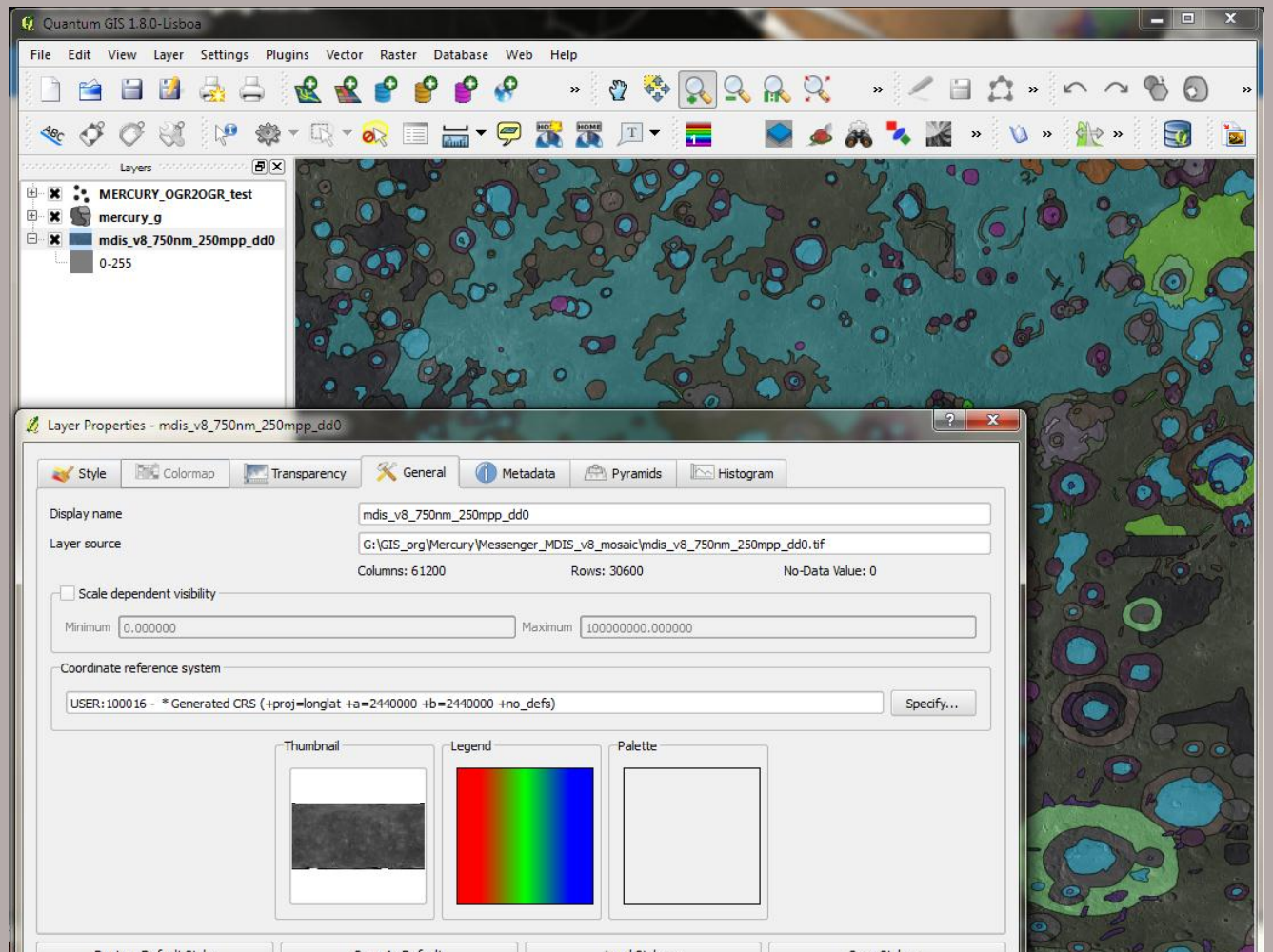
Direct Support Examples - ArcMap



Direct Support Examples - Opticks



Direct Support Examples - QGIS



No need for 16, 32? – convert to 8bit

- ISIS3 – "Stretch" seems to be the best. After run, bring use ISIS cub in ArcMap.

Stretch method 1

>stretch from=input.cub to=output_8bit.cub+8bit+1:254 USEPERCENTAGES=true pairs="0:1 100:254" null=0 lis=0 lrs=0 his=255 hrs=255

This allows you to specify input percentages for the mapping pairs. Thus when USEPERCENTAGES=true is set pairs="0:1 100:254" means:
map 0% to 1 (or the file's min value to 1) and 100% to 254 (file's max value).

Stretch method 2

This also means you can apply a recommended 0.5% clip to remove the potential extraneous lows and highs like:

> stretch from=input.cub to=output_8bit.cub+8bit+1:254 USEPERCENTAGES=true pairs="0:1 0.5:1 99.5:254 100:254" null=0 lis=0 lrs=0 his=255 hrs=255

No need for 16, 32 – convert to 8bit

- GDAL method
  - Manual method (more: http://bit.ly/pprlMK)

> gdalinfo –mm in.cub (returns min/max, now convert)

> gdal_translate –ot byte –scale min max 1 255 –a_nodata 0 in.cub out.tif

> gdal_translate –of PNG –ot byte –scale min max 1 255 –a_nodata 0 in.cub out.png

> gdal_translate –of JP2KAK –co quality=100 –ot byte –scale min max 1 255 –a_nodata 0 in.cub out.jp2

  - Cshell Script helper: http://bit.ly/oxIsQ7

GDAL for 32bit Map Projected ISIS3 (and PDS)

GDAL (binaries available using Fwtools, OSGeo4W, Mac - Kyng Chaos):

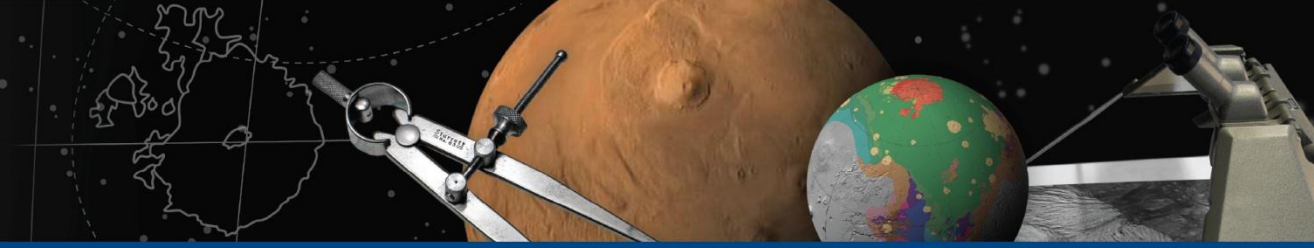
> gdal_translate –of GTIFF isis_ver3.cub isis_ver3.tif

GDAL Tips:

https://isis.astrogeology.usgs.gov/IsisSupport/index.php/topic,2172.0.html

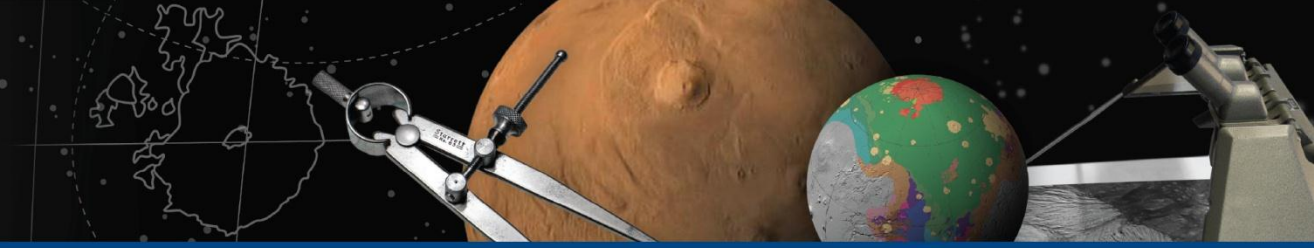
RULES of the GIS ROAD

- For ISIS processing
  - Best to set **same** projection and parameters for all
    - Note: optional to set same resolution
  - For visual (thematic) images, best to convert to 8bit
  - For "data" (e.g. DEM, Temperature -- 16,32 bit), set all ISIS Special Pixel Values to NULL (using specpix, stretch, bit2bit)
  - For global
    - If lonsys=360, then set clon=180
    - If lonsys=180, then set clon=0 *(better supported)*
  - Don't use *funky* projections



What is GDAL? (Geospatial Data Abstraction Library)

- GDAL is a “translator library for raster geospatial data formats”
- Open source
- Used in many applications: GRASS, UMN MapServer, Google Earth, ArcGIS 9.x, etc.
- Can handle many image formats for read and slightly fewer for write: AI Grid, Imagine, GeoTiff, JPEG, PNG, NetCDF, etc. (120 raster geospatial data formats)

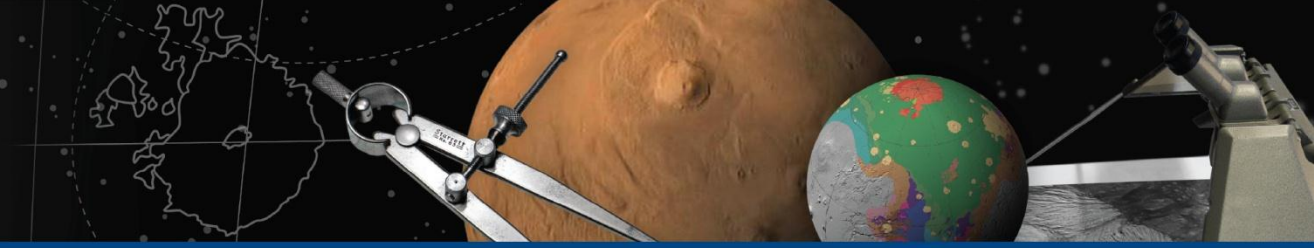


The End

GDAL
(Geospatial Data Abstraction Library)

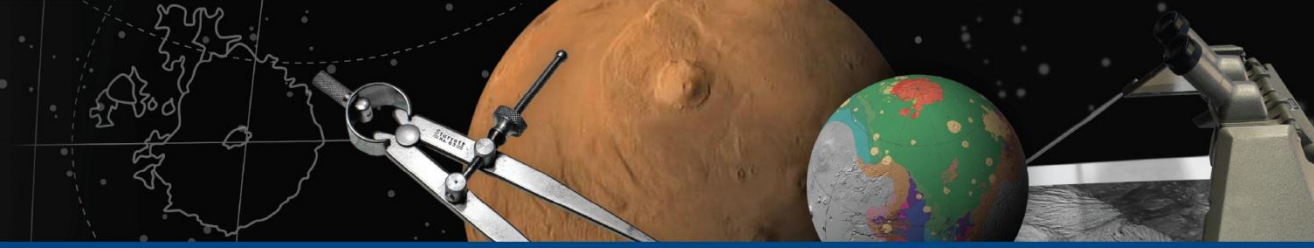
- Presents an “abstract data model” for processing spatial data
- Can be used directly from C/C++ and can be “wrapped” for use with Python, Perl, VB, C#, R, Java …
- Early developers have chosen Python as their scripting language and documentation is relatively good for this.

GDAL as of version 1.9.0 provides at least partial support for more than 120 raster geospatial data formats [ref]



Software that uses GDAL/OGR

- World Wind Java NASA's open source virtual globe and world imaging technology
- GRASS GIS
- OSSIM
- GvSIG
- JMap
- Quantum GIS
- MapServer
- Google Earth - A virtual globe and world imaging program.
- OpenEV
- SAGA GIS - a cross-platform open source GIS software
- R - an open source statistical software with extensions for spatial data analysis
- gdaltokmz, a Python module translating from GDAL-supported raster graphics formats to the Google Earth KMZ format
- ArcGIS 9.2 can use GDAL for customized raster formats
- TopoQuest - Internet topographic map viewer
- Orfeo toolbox - A satellite image processing library
- Biosphere3D – open source landscape scenery globe



NumPy (Numerical Python)

- An array/matrix package for Python
- Well suited for image processing – i.e. one function can operate on the entire array
- Slicing by dimensions and applying functions to these slices is concise and straightforward
- Nearly 400 methods defined for use with NumPy arrays (e.g. type conversions, mathematical, logical, etc.)