Rosetta, 2nd review, for map projected DUST

-Trent Hare, thare@usgs.gov

I am only reviewing the images found here

rogia\_1022/ro-c-gia-5-map-dust-v1.0/data

* flybys\_day\_side\_2015\_02\_05/
* perihelion\_2015\_06\_01/
* post\_equinox\_2015\_04\_14/
* post\_perihelion\_2015\_11\_01/
* prelanding\_bound\_orb\_2014\_08\_06/

Summary: The dust maps for this archive look much more consistent and well formed. This will make for a much easier archive for scientist use and compare against. With a few fixes below I was easily able to load these using multiple PDS3 readers and a few GIS mapping applications. Below are the only remaining issues I had with the current maps. Optional: the seidelmannetal2002 reference is somewhat outdate as the IAU has recently released a new version (in 2018) called: Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2015

1. All PDS3 image labels have a **malformed** START\_TIME and END\_TIME value. For example:

START\_TIME = 2015-02-04 04:42:14.553

STOP\_TIME = 2015-03-28 16:55:22.999

This is not valid since there is not a single string (string with no spaces). The format should be:

START\_TIME = 2015-02-04T04:42:14.553

This breaks a couple PDS3 image readers like the Python planetaryimage and GDAL. I have seen some archive with quotes around separated values but that is also not valid. Add “T” between the two strings will fix it per PDS3 guidelines:

<https://pds.nasa.gov/datastandards/pds3/standards/sr/StdRef_20090227_v3.8.pdf#page=74>

1. When testing these labels on LINUX, the reference to the image **filename is listed in all caps** yet the files are stored in the archive using lowercase. I recommend that if lowercase filenames are going to be used in the archive, then

^IMAGE = "GIA\_SPEED\_STD\_ESC1\_ESC2.IMG"

Should probably be:

^IMAGE = "gia\_speed\_std\_esc1\_esc2.img"

This breaks the Python PDS3 reader planetaryimage. Both GDAL and NASAView are tolerant.

1. Again, the labels are much better defined than the first review. Unfortunately, there still seems to be some confusion on how to define the map projection parameters. It seems either the radii are defined incorrectly or the **MAP\_SCALE is calculated wrong**. The MAP\_RESOLUTION looks good.

I assume the different radius values per data directory are correct and define a best-fit sphere to approximate the distance from the center of the body to the dust layer. If these are correct, the MAP\_SCALE can easily be calculated based on the radius used. For example, for the data products in flybys\_day\_side\_2015\_02\_05/ define a radius of 15.0 km. Using this value, and the fact the defined projection is "SIMPLE CYLINDRICAL", the MAP\_SCALE should be calculated as

MAP\_SCALE = radius <km> \* 2 \* PI (circumference in km) / LINE\_SAMPLES

Thus, for the 4 products in the flybys directory:

MAP\_SCALE = 9.204e-2 <km/pixel>

This defines the size of the pixel in map projected units (km) at the equator. Because a map projection defines a Cartesian coordinate system, this value is use for all pixels (in the X and Y directions), even at the poles. Applications which understand the defined map projection will understand how to deal with this. I recall there was confusion on how this could be true for polar areas. While the polar areas are grossly distorted, it is correct. A free mapping application to visuall test these files include QGIS (or possibly the Python viewer TuiView).