



**esac**

European Space Astronomy Centre (ESAC)  
European Space Agency (ESA)  
Camino Bajo del Castillo s/n  
Urb. Villafranca del Castillo  
28692 Villanueva de la Canada - Madrid  
SPAIN

# PSA PRODUCT USER GUIDE FOR AURORA-LPL\_TITAN-IMAGES\_DISR\_V1.0

**Prepared by** B. Grieger and E. Karkoschka  
**Reference**  
**Issue / Revision** 1 / 9  
**Date of Issue** 2021/03/04  
**Status**  
**Document Type** PUG  
**Distribution** -



## CHANGE LOG

Reason for change	Issue	Revision	Date
Started to collect data and to write the Product User Guide (PUG) for the Guest Storage Facility.	1	1	2020/11/26
Collected more image data. Replaced the notes by Erich with the newest once from the PDS website. A malformed rsync created unremovable stuff in dat/PGM_IMAGES/calibration.	1	2	2020/12/22
Wrote down some abstract math on re-de-compression.	1	3	2021/01/19
Noted a bit more on reconstruction of DCT coefficients.	1	4	2021/01/21
Brought roughly in ESAC format. Cleaned of superfluous stuff. Needed PNGs will should probably be saved.	1	5	2021/02/08
Back-uped the images for emulating the ESA look to comsim. Synchronized the preamble with the one from the GEOMAP PUG and put it in a dedicated file. Put the commands generating the title page in a macro. Took over from AIM_PLAN the automated generation of a change log from the CVS comments.	1	6	2021/02/11
Wrote the executive summary and part of the extended summary.	1	7	2021/02/11
Added a comprehensive explanation of the compression by Erich Karkoschka, along with some notes by myself.	1	8	2021/02/18
Added E. Karkoschka as author and put LPL in the dataset name. Completed draft of the extended introduction. Drafted the dataset introduction.	1	9	2021/03/04



## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Executive Summary . . . . .	4
1.2	Extended introduction . . . . .	4
1.3	AURORA-LPL_TITAN-IMAGES_DISR_V1.0 Introduction . .	5
1.4	Abbreviations and Acronyms . . . . .	6
1.5	Reference and Applicable Documents . . . . .	7
<b>2</b>	<b>Scientific Objectives</b>	<b>7</b>
2.1	Acknowledgements . . . . .	7
<b>3</b>	<b>Data Product Generation</b>	<b>7</b>
<b>4</b>	<b>Archive Format and Content</b>	<b>8</b>
4.1	Data provided in image format . . . . .	9
4.2	Data provided in ASCII table format . . . . .	11
4.3	Downlinked compressed data . . . . .	11
<b>5</b>	<b>Known Issues</b>	<b>11</b>
<b>6</b>	<b>Software</b>	<b>11</b>
<b>A</b>	<b>Notes on image processing by EK</b>	<b>12</b>
A.1	Overview . . . . .	12
A.2	Details c images . . . . .	13
A.3	Details e images . . . . .	15
A.4	Details g images . . . . .	20
<b>B</b>	<b>Compression</b>	<b>29</b>
B.1	My thoughts on the reconstruction of downlinked DCT coefficients . . . . .	29
B.2	Erich's explanation of compression (and my reflections) . . . .	31



# 1 Introduction

## 1.1 Executive Summary

The Descent Imager/Spectral Radiometer (DISR) consisted of 14 sub-instruments. We are here concerned with three particular sub-instruments, the three pancromatic imagers. The 606 successfully downlinked images are in principle available in the DISR dataset on the PSA, however, they are in inconvenient formats and the information is incomplete or difficult to find. Herewith, we provide a comprehensive collection of images in all processing levels, detailed information on on board processing and image compression and further on ground processing, and the originally downlinked DCT compression coefficients. This enables the reproduction of the complete processing chain, facilitating future image reprocessing.

## 1.2 Extended introduction

*<In addition to the table of contents on the previous page, you may wish to consider including a list of figures and/or a list of tables if applicable. The table of contents need not be included in the latter, as this list is intended to provide users a way to quickly find any data or informational tables provided in your PUG.>*

The DISR (pronounced *dizzer* by the PI team) aboard the Huygens probe consisted of 14 sub-instruments, four solar aureole cameras, two spectrometers covering the visible spectrum, two spectrometers in the near infrared, two violet photometers, a sun sensor, and three pancromatic imagers. The data from all these sub-instruments is in principle available in the DISR dataset on the PSA. However, at least concerning the three imagers, the information is not complete. This applies firstly to the images themselves:

- Not all available versions of further processed images are present.
- Some further processed versions are present but difficult to find in a cluttered 'EXTRAS' directory.
- The images in the regular 'DATA' directory are in inconvenient or unreadable formats.

Secondly, it applies to information about the image processing:



- Because of some on board processing, we do not have really raw images. No information on the on board processing is provided.
- The parameters of the compression applied to each image are not provided.
- The originally downlinked DCT compression coefficients, which represent the data closest to the original images that we have, are not provided.

With this dataset, we want to preserve the full information about the three imagers of the DISR aboard the Huygens probe. This entails replication of some data already available in the PSA and new data not available before, all in a well structured collection.

### 1.3 AURORA-LPL\_TITAN-IMAGES\_DISR\_V1.0 Introduction

*<The actual introduction of the PUG will be written to explain the purpose of this document with a quick overview, i.e. that it is intended as a guide on how to use the dataset that you are providing to the PSA. If you are referencing any other documents, they should be listed as part of the introduction as well, usually in section 1.3 (essentially like the bibliography of a peer-reviewed article.>*

This dataset contains products in two principal formats: images in PGM format and tables in ASCII text.

PGM images represent different version of the 606 successfully downlinked images or calibration data for the three different imagers (SLI, MRI, and HRI) like flat fields.

Some ASCII text files provide specific tables for each image and others provide tables with data for all images, one image per row.

It is important to note that we do not have the really raw images, because of some onboard processing before downlinking. However, we provide complete information about the onboard processing and also about the on ground processing that was conducted enhance the downlinked images.

For information about this dataset, contact

Björn Grieger



Aurora Technology B. V. for the European Space Agency (ESA)  
European Space Astronomy Centre (ESAC)  
Camino Bajo del Castillo s/n  
28692 Villanueva de la Caada  
Madrid  
Spain  
Bjoern.Grieger@esa.int  
T +34 91 81 31 107

or

Erich Karkoschka  
Lunar and Planetary Laboratory  
The University of Arizona  
1629 E. University Blvd.  
P. O. Box 210092 Tucson, AZ 85721-0092  
USA  
erich@pirlshell.lpl.arizona.edu

## 1.4 Abbreviations and Acronyms

- CCD — Charge Coupled Device, a type of 2D image sensor
- DCT — Digital Cosine transform, an image compression scheme
- DISR — Descent Imager/Spectral Radiometer, an instrument on the Huygens probe
- ESA — European Space Agency
- GSF — The Guest Storage Facility of the PSA
- HRI — High Resolution Imager of the DISR
- MRI — Medium Resolution Imager of the DISR
- PDS3 — Planetary Data System version 3, a format standard for planetary data
- PGM — Portable Graymap, a relatively simple image format
- PI — Principal Investigator, e. g., leader of an instrument team
- PSA — ESA's Planetary Science Archive
- SLI — Side Looking Imager of the DISR



## 1.5 Reference and Applicable Documents

*< If you are referencing any other documents, they should be listed here (i.e. this is the bibliography section). You may use the citation style of your choice, but please apply it consistently.>*

## 2 Scientific Objectives

*< This section should cover the scientific objectives which led to the observations and the creation of this dataset. Additionally, the research method and/or analytical techniques utilised in the creation of the data will be explained here. The reference papers listed in bibliographical detail in the Introduction section should be referred to here as well (e.g. Smith et al. [2099]), with a bit of explanation on how those papers apply to this dataset. This section should provide more of a general overview, while the next will cover more details.>*

### 2.1 Acknowledgements

*< In this subsection you should acknowledge the sources of your data. For example, if part of the source material came from an instrument on a spacecraft, make sure to acknowledge that team here. If any of the source data is publicly archived, such as in the PSA, make sure to mention the dataset names, and direct links to the data could be helpful to end users as well.>*

*< This is also the area in which you can specify to the archive user on how you would like your own work to be acknowledged if it is used by them for further research.>*

## 3 Data Product Generation

*< In this section more details will be provided on how the source data for these data products was gathered. Then, the specific scientific and/or engineering techniques that were applied to arrive at the final form of the data products should be explained in more detail. Giving explanations of each step is needed for the end users. Any calibration methods and parameters, and the processing level of the data will be discussed here as well. Comparisons to other results,*



*if applicable, should also be included in this section. Other topics to discuss can include instrument modes, assumptions made in the analysis, etc.>*

*<Finally, there shall be some discussion of how this dataset was reviewed and validated. For most external datasets (i.e. not those provided as part of regular deliveries by a mission's Principal Investigator teams) this can be covered by referencing and including the details of a published, peer-reviewed article which corresponds to these data products being provided to the PSA. In case no such article exists, a review would have to be conducted by a third party, which would need to be discussed in detail with the PSA prior to acceptance of the dataset.>*

## 4 Archive Format and Content

*<Here, please indicate to the users which archiving standard was used, e.g. PDS3 or PDS4, though it is possible your dataset may not use either of those 2 formats. The directory and file structures used in the dataset must be explained. For example, the DATA directory (in the case of PDS3) can include several levels of subdirectories, for separate instrument channels or to divide the data by different observation periods. The method which was used to subdivide the data should be explained here. All folders or collections are to be described in this section.>*

*<The structure on contents of the individual data products shall be explained here as well. For example, if you provide the data in binary format, an explanation of how to decode the data into human-legible form should be provided. Please keep in mind that this will be read by new scientists many decades from now who are not familiar with your data yet, thus provide as much detail as you can.>*

*<Additionally, instructions to the users on how to open the products is very important. For example, it is useful to reference the tools that are best to use for these products (e.g., ArcGIS for shape files, etc...), and any specific information on how to open the product. The user should be able to open the product easily following your guidelines.>*

I use 'dat' as the root directory for the dataset and put there symbolic links to all included data. So I can access it independently of the final name of the dataset. The final name is a symbolic link in 'out' pointing to 'dat'. This is all done from the 'Makefile'.





## 4.1 Data provided in image format

We have 606 downlinked images. The numbering is not consecutive because one of the two downlink channels did not work and about half of the images were lost. The images are provided in different processing levels. The orientation of lower level images followed the original pixel numbering from the CCD. This is called “a orientation”. These images are mirrored, so you have to flip them left/right. Higher level images are rotated by 180° with respect to the lower level images. This is called “e orientation”. These images need to be flipped up/down.

Details on the data format are provided below the list of images.

These are the available images:

- a images:** “raw” (but processed onboard, a orientation). The decompression results in pixel values in the range 0–255. These have been multiplied by 128 and then rounded to the nearest integer.
- c images:** calibrated but not smoothed (a orientation). Besides other processing detailed in section 3, these images have been “inversely square rooted” to restore the range from 8 bit to 12 bit. Thus, the value range is 0–4095. These values have been multiplied by eight and rounded to the nearest integer.
- e images:** further processed to remove compression artifacts which implies some smoothing (e orientation), same value range as c images [There are additional new versions of some images, with 1000 added to their image number.](#)
- g images:** absolutely calibrated and distortion corrected (e orientation) [There are two version, one in the V1.3 dataset and one in ‘CHUCK\\_NEW’. They differ, Erich’s note on g image processing may explain that. I take the new version. In his note on g images, EK writes that the new scaling yielded pixel values above 32767, and these have to be interpreted as unsigned two-byte integers.](#)
- m images:** “haze subtracted” (e orientation) [Have spurious white patches; maybe they go below zero? There are only 588.](#)
- j images:** m images divided by the atmospheric transmission (e orientation) [There are only 587.](#)



Then we have calibration data provided in image format individually for each image:

**d images:** modeled dark current (a orientation) [Only the first 136. Afterwards, the dark current is supposed to be zero.](#)

**f images:** improved flat fields (a orientation)

**k images:** atmospheric transmission (e orientation) [Only 587.](#)

All these images reside in respective directories named ‘?\_images’.

Some calibration data is the same for all images from the same imager, so there are only three different images for the three imagers SLI, MRI, and HRI (or maybe a few more). All these images are in a directory named ‘calibration’.

**Onboard flat fields:** normalized to 10000 average (a orientation), files ‘flatha’ (HRI), ‘flatma’ (MRI), and ‘flatsa’ (SLI) [In ‘DISR/FLATs’.](#)

**Maps of distortion corrected images:** (e orientation) files ‘g-001’ (SLI), ‘g-002’ (MRI), and ‘g-003’ (HRI), ‘g-004’, ‘g-005’, ‘g-006’

All images are provided in Portable Gray Map (PGM) binary format. The binary data is preceded by an ASCII header. The header is 19 characters long and contains the magic number for PGM binary (P5), the number of columns, the number of rows, and the maximum pixel value. Header fields are separated by blanks. For PGM binary, the standard requires exactly one white space character between the header and the binary data. For almost all images herein, a Line Feed is used to separate header and data. However, the a images have a Data Link Escape character instead, which may not be interpreted as white space. Therefore, some image viewers — like the Linux Image viewer — cannot read these images. The file editor emacs can. When writing your own code to read the images, just skip the first 20 bytes for reading the binary data. This should work for all images herein.

The actual binary data is in two byte big endian format. PGM readers expect the data row by row from left to right, starting with the top row. The higher processed images (labeled with e orientation above) were intended to be in the real world orientation, but accidentally they were written starting with the bottom row. Therefore, if you use a standard PGM reader, you have to flip the images up/down. The lower processed images (labeled with



a orientation above) follow the original pixel numbering of the CCD and are rotated upside down relative to the higher processed images, therefore you have to flip them left/right (and *not* up/down).

## 4.2 Data provided in ASCII table format

For each image, there is a “square routing” table for the onboard conversion from 12 bit to 8 bit.

There is one table providing the estimated compression parameters for each image.

File `timeh.dd` lists the image number of the HRI image and a transmitted value which is related to the pixel-independent dark value that is not included in the dark images.

The pixel-independent dark data number is  $0.9 + 0.25 * \text{value in table}$ . The image number of MRI and SLI images of the same exposure is 2 and 1 less than that of the HRI image.

## 4.3 Downlinked compressed data

A [DESCRIPTION](#) in the label files for the XDR files echoes header entries from the XDR file. These provide some information on the original ESOC files:

```
engineer:          Chuck See
ESOC_File:         C:\df3\15Jan05\ESOC_Files\o524sd__.1h_
```

## 5 Known Issues

*<Here is the place to report any known issues, if applicable. This information is important to provide to the user to avoid mis-usage of your products.>*

## 6 Software

*<If there are any commonly available software applications that would be useful in working with this dataset, they shall be listed here, including links*



*for users to download said software. Additionally, if you included your own custom code as part of the dataset delivery, details on how to use this code should be explained in this section.>*

## A Notes on image processing by EK

### A.1 Overview

The DISR images in this directory have been processed to three levels:

1) The 'Unsmoothed Images' have the least processing. Steps 1 through 12 below have been performed on them. Thus their photometric stretch has been square root expanded to return them to 12 bits of depth, they have been flat-field corrected to eliminate the camera's photometric distortions, the dark current has been removed using the imager model at the exposure temperature, the electronic shutter effect (induced by data clocking) has been compensated for, the images were further flat fielded to remove artifacts seen in the higher altitude images (which viewed the relatively flat upper atmosphere), the stretch was enhanced to increase the dynamic range and known bad pixels were replaced by their neighbors.

2) The 'E-Images' have been further processed to remove compressor induced artifacts, account for the imagers point spread functions (psf) and adjust for negative and saturated pixels. This helps reduce the annoying periodic wave pattern that occurs when features observed are near the imagers noise floor.

3) The 'G-Images' were even further processed to adjust for geometric distortions and intensity calibration into I/F (normalized intensity).

The images are named with a letter (c or e or g) and the 4-digit image number from 0002 to 1215.

E\_NEAR\_SURFACE Images # 1711-1721 correspond to original images #0711-0721 (with 1000 added to their image number). They show the effect of stray light that leaked directly from the Surface Science Lamp to the CCD, which was roughly subtracted for images # 0711-0721.

User should be aware that many images have saturated pixels, particularly



the first images taken on Titan's surface (image 723 and following), but also some images taken when the Calibration Lamps were on. Indeed some are completely saturated. Saturated pixels are indicated by a data number of 32000.

Orientation: The CCD was oriented such that the first data read out came from the top right corner. Subsequent data came from the top row, right to left. Following were the rows below, etc. This order was preserved for the C-images. Viewers that start displaying data from the bottom left show these images rotated by 180 degrees. Viewers that start displaying from the top left show these images flipped left to right.

The E- and G-images were rotated 180 degrees for a correct orientation with viewers starting at the bottom left. Viewers staring at the top left show them flipped top to bottom.

The easiest way to determine your viewers convention is to look at a surface SLI (like image 724). Proper orientation is with the sky to the top and the large broken rock in the foreground to the left of the scene.

There are 23 processing steps for the E-images and the 6 steps for the G-images. They are further described in their respective directories.

## A.2 Details c images

The 'Unsmoothed\_Images' (i.e. C-Images) are unchanged from 2005. Ergo, they are located in EXTRAS\PROCESSED\_IMAGES\2005\UNSMOOTHED\_IMAGES\_13NOV2005

This is a brief description of the processing done on the DISR raw images to produce the images seen in this directory ('UnSmoothed\_Images'). The smoothing steps were not performed on these images in order to preserve high frequency features.

In general, the on-board processing steps were reversed (square rooting, compression, flat fielding), and the instrument induced artifacts (shutter effect, dark current, bleed-through, bad pixels, etc) are removed. Thus, the C-images reflect the the removal of known instrumental effects.

The specific processing steps follow:



Processing of DISR images

Nov 13, 2005

Erich Karkoschka

- 1) I estimated the compressor threshold for each image based on the relative frequency for the three lowest amplitude bins.  
I tested the estimates with the compressor simulation with three images. The difference between actual and estimated threshold were 0, 0.125, and 0.25, which is good enough.
- 2) I changed the square-root lookup table into a real function.
- 3) For adaptive schemes, I changed the slope of the function in both outer section to a new slope, which is a quarter of the original slope plus three quarters of the slope in the middle section. This way, regions outside the limits are not as noisy as in the original images, but still noisier than in the middle region.
- 4) I applied the reverse of the square-rooting to the data.
- 5) I multiplied the data by the on-board flatfield.
- 6) For each exposure I created a dark current image according to the standard bias and the dark current image scaled by the measured dark current as function of time, called the dark model.  
I subtracted it from the data.
- 7) I subtracted a constant data number from the whole image: one quarter of the dark current number (which peaks at 34 DN).
- 8) I subtracted the expected contributions of light during the vertical transfer of data in the CCD. This is based on the assumption that each pixel sees the light at the actual position for the exposure time and the light of each pixel below (lower row number) for 2 microseconds each.
- 9) I subtracted the constant of 0.9 DN from the whole image.
- 10) For images taking after landing, I subtracted the following data numbers  $14.5 - 11.5 * \text{SIND}(360.*((228-J)/165.-((228-J)/300.)**2))$ , where J is the row number from 1 to 256 (in the SLI row 1 is above



the horizon).

- 11) I shifted the on-board flatfields by cubic interpolation for resampling. The assumed shift of flatfield features with CCD temperature T(K) in micropixels is:

SLI: X = 1485 (T-254.6)

Y = -569 (T-223.9)

MRI: X = 1678 (T-257.0)

Y = -563 (T-221.9)

HRI: X = 819 (T-260.8)

Y = -704 (T-228.0).

I modified the flatfields according to obvious flatfield artifacts seen in high-altitude images, typically by about 1 percent or a few percent. Most of them were adjustments next to the edge of the field of view. For the HRI, however, there was an adjustment of about 1 percent across the whole field of view. For the MRI, there was an adjustment of 2 percent over the top 50 rows. The latter two corrections are probably due to non-constant intensity of the integrating sphere.

I then divided the image by the adjusted flatfield.

### A.3 Details e images

The DISR images in this directory have been processed to the second level (E-Images):

C-Images: These "Unsmoothed Images" have the least processing. Steps 1 through 11 below have been performed on them. Thus their photometric stretch has been square root expanded to return them to 12 bits of depth, they have been flat-field corrected to eliminate the camera's photometric distortions, the dark current has been removed using the imager model at the exposure temperature, the electronic shutter effect (induced by data clocking) has been compensated for, the images were further flat fielded to remove artifacts seen in the higher altitude images (which viewed the relatively flat upper atmosphere), the stretch was enhanced to increase the dynamic range and known bad pixels were replaced by their neighbors.

E-Images: These "Smoothed Images" have been further processed to remove compressor induced artifacts, account for the imagers point spread functions (psf) and adjust for negative and saturated pixels. This helps reduce the annoying periodic wave pattern that occurs when features observed are near



the imagers noise floor. It also removes the compression artifacts at 16x16 pixel block boundaries. Note that the original version of November 2005 also removed some real features near the noise level in order to achieve full suppression of artifacts. The more sophisticated method applied here shows significantly more features near the noise level.

Following are the 22 steps for the E-images.

Processing of DISR images

Dec 13, 2017

Erich Karkoschka

With respect to the Nov 13, 2005 version, only 15) was changed, resulting in images with significantly more resolution and features visible without significantly affecting the suppression of compression artifacts.

- 1) I estimated the compressor threshold for each image based on the relative frequency for the three lowest amplitude bins.  
I tested the estimates with the compressor simulation with three images. The difference between actual and estimated threshold were 0, 0.125, and 0.25, which is good enough.
- 2) I changed the square-root lookup table into a real function.
- 3) For adaptive schemes, I changed the slope of the function in both outer section to a new slope, which is a quarter of the original slope plus three quarters of the slope in the middle section. This way, regions outside the limits are not as noisy as in the original images, but still noisier than in the middle region.
- 4) I applied the reverse of the square-rooting to the data.
- 5) I multiplied the data by the on-board flatfield.
- 6) For each exposure I created a dark current image according to the standard bias and the dark current image scaled by the measured dark current as function of time, called the dark model.  
I subtraced it from the data.
- 7) I subtracted a constant data number from the whole image: one





quarter of the dark current number (which peaks at 34 DN).

- 8) I subtracted the expected contributions of light during the vertical transfer of data in the CCD. This is based on the assumption that each pixel sees the light at the actual position for the exposure time and the light of each pixel below (lower row number) for 2 microseconds each.
- 9) I subtracted the constant of 0.9 DN from the whole image.
- 10) For images taking after landing, I subtracted the following data numbers  $14.5 - 11.5 * \text{SIND}(360.*((228-J)/165.-((228-J)/300.)**2))$ , where J is the row number from 1 to 256 (in the SLI row 1 is above the horizon).
- 11) I shifted the on-board flatfields by cubic interpolation for resampling. The assumed shift of flatfield features with CCD temperature T(K) in micropixels is:
  - SLI: X = 1485 (T-254.6)  
Y = -569 (T-223.9)
  - MRI: X = 1678 (T-257.0)  
Y = -563 (T-221.9)
  - HRI: X = 819 (T-260.8)  
Y = -704 (T-228.0).

I modified the flatfields according to obvious flatfield artifacts seen in high-altitude images, typically by about 1 percent or a few percent. Most of them were adjustments next to the edge of the field of view. For the HRI, however, there was an adjustment of about 1 percent across the whole field of view. For the MRI, there was an adjustment of 2 percent over the top 50 rows. The latter two corrections are probably due to non-constant intensity of the integrating sphere.

I then divided the image by the adjusted flatfield.
- 12) I replaced the data numbers at bad pixels by the average of the data numbers of the nearest good pixels. Bad pixels were those which were replaced before the application of the on-board flatfield. Also I added to the list of bad pixels about 100 pixels each in the SLI and HRI next to the edge of the field of view, which did not seem to produce consistent data numbers.
- 13) At this point, the image should be free of artifacts.



I compared the discrete Fourier transform of the resulting data with the original transforms. For each Fourier coefficient, the original data has a minimum and maximum possible value. The standard processing adopts the mean between the minimum and maximum. Because the processing steps 1) through 10), the minimum and maximum values will be quite different. The fact that some processing steps are non-linear in the Fourier coefficients is neglected here. The amplitudes of the Fourier coefficients are changed in such a way that amplitudes are reduced. The (absolute) smaller amplitudes are reduced the most to values close to their (absolute) minimum. Large amplitudes are essentially unchanged to the mean between the minimum and maximum possible value. This smoothes features near the noise level with are not artifacts.

- 14) I increased the smoothing in 16x16 pixel blocks where the average spatial frequency of transmitted coefficients is high. Such a high value means usually that most of the data are noise, while real features typically have more low frequency coefficients.
- 15) I did a similar smoothing operation for 32x32 pixel blocks, shifted by 16 pixels in x and y. The inner 16x16 blocks are in four such blocks, and the results of the four calculations are averaged with increased weighting towards the center of a 32x32 pixel block. This smoothes discontinuities at 16x16 pixel block boundaries. It especially decreases amplitudes of spatial frequencies which are non-zero at only one of the four 16x16 blocks in a 32x32 pixel block. In that case, the pattern does not abruptly stop at the boundary, but decreases smoothly in amplitude across the boundary. Because the 32x32 pixel block data do not have minimum and maximum possible values. The amount of smoothing can be set. For the current run I used a smoothing factor  $SF = 1.5$ . With respect to the previous version I increased the smoothing of high frequencies near the noise level. This increase was tailored individually for each 16x16 pixel block according to the strengths of flatfield features, which caused image artifacts through image compression.
- 16) I increased the smoothing in 32x32 pixel blocks at areas where the variation of data number with position is much lower than the mean variation in that block. Typically, the compressor gets the strongest variations right but makes small errors where variations are much smaller than the average ("ringing").



This method decreased somewhat the ringing near strong features. All the smoothing operations together take out 88 percent of the coefficients transmitted to Earth, or reduce them to less than half of their original amplitude. Most of these coefficients are high frequency noise or high frequency patterns which are difficult to interpret if low spatial frequencies are missing.

- 17) For the last six HRI before landing, the lamp is on and creates a bright sloping background signal. I estimated this signal by adding the images and smoothing them. I subtracted this signal from these six images. I estimate that the subtraction of the about 1000 DN is accurate to 50 DN and may be as good as 20 DN.
- 18) I deconvolved the images before touch-down according to the program of April 2001. This program calculates a synthetic PSF for each location, based on an analysis of PSFs taken during testing before launch. I used the following widths (FWHM) of the PSFs for the center and edge (at top center or bottom center): WIC = 1.5 (pixels) and WIE = 1.5 (SLI), 2.0 (HRI), and 2.5 (MRI). For the images taken from the surface, I changed the PSFs according to the defocus calculated from the distance, which is based on the assumption that the SLI window was 47 cm above the surface and the nose was pointing 1.7 degrees up. For these images, I took WIC = WIE, only dependent on the imager: 1.0 for the SLI, 1.5 for the MRI, and 4.5 for the HRI. The main purpose of deconvolution is not a sharpening but a transformation into consistent PSFs. Since the original PSFs have FWHM near 1.5 over parts of the field of view, the resulting images are not sharper. However, circular features become circular again. This is most noticeable near the corners of the MRI where all small features have radial streaks in raw images.
- 19) I flipped the images right/left and up/down.
- 20) I set negative data numbers to zero.
- 21) I investigated saturation. Saturation in the processed images occurred at data numbers somewhat below 4000 due to flatfield division, dark current, photon accumulation during transfer, and due to the compressor. Saturation essentially also occurred at the point where the adaptive square-root scheme has a discontinuity in the slope. Sometimes, the slope changes by more than a factor



of 50. If the signal-to-noise ratio was 200 just below that point, this ratio may be only 3 just above this point, which is a useless measurement. Thus, all data numbers above that point were reduced to the same DN, which is listed in the file header.

22) I multiplied the data numbers by eight and rounded them to integers.

23) I wrote the image in pgm format, which is the simplest of the standard image formats such as pgm, jpeg, tiff, and postscript.

-----  
Further notes....

The TIFF images were derived from the PGM images using IDL\*. For Photometry the PGM's should take precedence. The TIFF images are provided to allow ready viewing on different platforms. The TIFF images are stretched (min to max) from 0 to 65535, or 16 BitsPerPixel (BPP). No pixels were discarded.

ASCII tabular data is also provided in the 'TABLE' directory. They contain the pixel values from the PGM images by row and column in ASCII form.

The filenames contain the beginning observation mission time in seconds, and the observation altitude in Kilometers (KM) or meters (M) as specified.

E\_NEAR\_SURFACE Images # 1711-1721 correspond to original images #0711-0721 (with 1000 added to their image number). They show the effect of stray light that leaked directly from the Surface Science Lamp to the CCD, which was roughly subtracted for images # 0711-0721.

\*Interactive Data Language V6.1

## A.4 Details of images

This is a brief description of the processing done on the DISR raw images to produce the images seen in this directory ('G-IMAGES').

In general, the on-board processing steps were reversed (square rooting, compression, flat fielding), and the instrument induced artifacts (shutter effect, dark current, bleed-through, bad pixels, etc) removed. The images



were then smoothed and deconvolved using point spread functions from laboratory measurements. Linearity and optical aberrations were compensated. They were then geometrically corrected and photometrically calibrated in reflectivity before being converted to 16 bit TIFF images. They are also output in PGM format.

Most of the High Resolution Images (HRI's) taken immediately after probe landing were hard saturated due to the Surface Science Lamp reflecting off Titan's surface. After about 2000 seconds, the autoexposure routine corrected this problem so that only the right part of the HRI images was saturated. The saturated MRI's (from 9008 to 10911 seconds, DDB time) have not been removed from this archive.

The specific processing steps follow:

Processing of DISR images                      Dec 13, 2017                      Erich Karkoschka

Note: the only change with respect to the 2005 version is the different scaling and the use of updated E-images. The new scaling gives data numbers above 32767 for 16 SLI images above 40 km altitude. These data numbers have to be interpreted as unsigned two-byte integers

- 1) I estimated the compressor threshold for each image based on the relative frequency for the three lowest amplitude bins.  
I tested the estimates with the compressor simulation with three images. The difference between actual and estimated threshold were 0, 0.125, and 0.25, which is good enough.
- 2) I changed the square-root lookup table into a real function.
- 3) For adaptive schemes, I changed the slope of the function in both outer section to a new slope, which is a quarter of the original slope plus three quarters of the slope in the middle section. This way, regions outside the limits are not as noisy as in the original images, but still noisier than in the middle region.
- 4) I applied the reverse of the square-rooting to the data.
- 5) I multiplied the data by the on-board flatfield.



- 6) I created a images according to Andy's equation for each image and subtracted it from the data.
- 7) I subtracted a constant data number from the whole image: one quarter of the dark current number (which peaks at 34 DN).
- 8) I subtracted the expected contributions of light during the vertical transfer of data in the CCD. This is based on the assumption that each pixel sees the light at the actual position for the exposure time and the light of each pixel below (lower row number) for 2 microseconds each.
- 9) I subtracted the constant of 0.9 DN from the whole image.
- 10) For images taking after landing, I subtracted the following data numbers  $14.5 - 11.5 * \text{SIND}(360.*((228-J)/165.-((228-J)/300.)**2))$ , where J is the row number from 1 to 256 (in the SLI row 1 is above the horizon).
- 11) I shifted the on-board flatfields by cubic interpolation for resampling. The assumed shift of flatfield features with CCD temperature T(K) in micropixels is:

```

SLI:  X = 1485 (T-254.6)
      Y = -569 (T-223.9)
MRI:  X = 1678 (T-257.0)
      Y = -563 (T-221.9)
HRI:  X =  819 (T-260.8)
      Y = -704 (T-228.0).
  
```

I modified the flatfields according to obvious flatfield artifacts seen in high-altitude images, typically by about 1 percent or a few percent. Most of them were adjustments next to the edge of the field of view. For the HRI, however, there was an adjustment of about 1 percent across the whole field of view. For the MRI, there was an adjustment of 2 percent over the top 50 rows. The latter two corrections are probably due to non-constant intensity of the integrating sphere.

I then divided the image by the adjusted flatfield.

- 12) I replaced the data numbers at bad pixels by the average of the



data numbers of the nearest good pixels. Bad pixels were those which were replaced before the application of the on-board flatfield. Also I added to the list of bad pixels about 100 pixels each in the SLI and HRI next to the edge of the field of view, which did not seem to produce consistent data numbers.

- 13) At this point, the image should be free of artifacts. I compared the discrete Fourier transform of the resulting data with the original transforms. For each Fourier coefficient, the original data has a minimum and maximum possible value. The standard processing adopts the mean between the minimum and maximum. Because the processing steps 1) through 10), the minimum and maximum values will be quite different. The fact that some processing steps are non-linear in the Fourier coefficients is neglected here. The amplitudes of the Fourier coefficients are changed in such a way that amplitudes are reduced. The (absolute) smaller amplitudes are reduced the most to values close to their (absolute) minimum. Large amplitudes are essentially unchanged to the mean between the minimum and maximum possible value. This smoothes features near the noise level with are not artifacts.
- 14) I increased the smoothing in 16x16 pixel blocks where the average spatial frequency of transmitted coefficients is high. Such a high value means usually that most of the data are noise, while real features typically have more low frequency coefficients.
- 15) I did a similar smoothing operation for 32x32 pixel blocks, shifted by 16 pixels in x and y. The inner 16x16 blocks are in four such blocks, and the results of the four calculations are averaged with increased weighting towards the center of a 32x32 pixel block. This smoothes discontinuities at 16x16 pixel block boundaries. It especially decreases amplitudes of spatial frequencies which are non-zero at only one of the four 16x16 blocks in a 32x32 pixel block. In that case, the pattern does not abruptly stop at the boundary, but decreases smoothly in amplitude across the boundary. Because the 32x32 pixel block data do not have minimum and maximum possible values. The amount of smoothing can be set. For the current run I used a smoothing factor  $SF = 1.5$ . With respect to the previous version I increased the smoothing of high frequencies near the noise level.



- 16) I increased the smoothing in 32x32 pixel blocks at areas where the variation of data number with position is much lower than the mean variation in that block. Typically, the compressor gets the strongest variations right but makes small errors where variations are much smaller than the average ("ringing"). This method decreased somewhat the ringing near strong features. All the smoothing operations together take out 88 percent of the coefficients transmitted to Earth, or reduce them to less than half of their original amplitude. Most of these coefficients are high frequency noise or high frequency patterns which are difficult to interpret if low spatial frequencies are missing.
- 17) For the last six HRI before landing, the lamp is on and creates a bright sloping background signal. I estimated this signal by adding the images and smoothing them. I subtracted this signal from these six images. I estimate that the subtraction of the about 1000 DN is accurate to 50 DN and may be as good as 20 DN.
- 18) I deconvolved the images before touch-down according to the program of April 2001. I used the following widths (FWHM) of the PSFs for the center and edge (at top center or bottom center): WIC = 1.5 (pixels) and WIE = 1.5 (SLI), 2.0 (HRI), and 2.5 (MRI). For the images taken from the surface, I changed the PSFs according to the defocus calculated from the distance, which is based on the assumption that the SLI window was 47 cm above the surface and the nose was pointing 1.7 degrees up. For these images, I took WIC = WIE, only dependent on the imager: 1.0 for the SLI, 1.5 for the MRI, and 4.5 for the HRI. The main purpose of deconvolution is not a sharpening but a transformation into consistent PSFs. Since the original PSFs have FWHM near 1.5 over parts of the field of view, the resulting images are not sharper. However, circular features become circular again. This is most noticeable near the corners of the MRI where all small features have radial streaks in raw images.
- 19) I flipped the images right/left and up/down.
- 20) I set negative data numbers to zero.
- 21) I investigated saturation. Saturation in the processed images occurred at data numbers somewhat below 4000 due to flatfield division, dark current, photon accumulation during transfer,





and due to the compressor. Saturation essentially also occurred at the point where the adaptive square-root scheme has a discontinuity in the slope. Sometimes, the slope changes by more than a factor of 50. If the signal-to-noise ratio was 200 just below that point, this ratio may be only 3 just above this point, which is a useless measurement. Thus, all data numbers above that point were reduced to the same DN, which is listed in the file header.

- 22) I multiplied the data numbers by eight and rounded them to integers.
- 23) I wrote the image in pgm format, which is the simplest of the standard image formats such as pgm, jpeg, tiff, and postscript.

---

Calibration of DISR images                      Nov 13, 2005                      Erich Karkoschka

- 1) I defined a geometric solution which attaches an azimuth and nadir angle to the center of each pixel. The solution is based on laboratory images of a pattern and on laboratory images of a point source in specified directions. For the SLI and HRI, the solution was adjusted based on simultaneous descent images in overlapping areas between the SLI and MRI and between the MRI and SLI. This solution is estimated to be accurate to 0.03 degrees. It differs from the previous solution by about 1 degree or less. The improvements over the previous solution include correcting a processing error for the SLI, avoiding a blow-up of the solution towards the corners of the field of view, adjusting laboratory measurements for a periodic error and a backlash of the mount used in the laboratory, and using descent images to adjust the relative pointing. The latter correction was only 1-2 pixels.
- 2) I resampled all images according to the geometric solution by cubic interpolation. The calibrated images are gnomonic projections with the zero-azimuth meridian half way between the two central columns and the following nadir angles half way between the two central rows:  $N_{Ac} = 70.3 \text{ deg (SLI)}, 31.3 \text{ deg (MRI)}, 14.5 \text{ deg (HRI)}$ .



The central scale in the calibrated images is:

0.22 deg/pix (SLI), 0.125 deg/pix (MRI), 0.062 deg/pix (HRI), or in radians: SC = 0.0038397/pix (SLI), 0.0021817/pix (MRI), 0.0010821/pix (HRI).

- 3) Equations to convert between column x, row y and azimuth AZ, nadir angle NA:

xc and yc are the center pixels. For example, if x varies between 0 and 127 and y varies between 0 and 255, xc=63.5, yc=127.5.

$$x = xc + \sin AZ / [SC * (\sin NAc \cos AZ + \cos NAc / \tan NA)]$$

$$y = yc + (\cos AZ \tan NA - \tan NAc) / [SC * (\cos AZ \tan NA \tan NAc + 1)]$$

$$\tan AZ = (x - xc) / [(y - yc) \cos NAc + \sin NAc / SC]$$

$$\tan NA = \text{SQRT}\{ (x - xc)^2 + [(y - yc) \cos NAc + \sin NAc / SC]^2 \} / [\cos NAc / SC - (y - yc) \sin NAc]$$

(The azimuth AZ is positive towards the right)

- 4) The pixel interpolation routine gives data numbers for the edge of the original field of view if one asks for a pixel outside the original field of view. Thus, a few columns and rows near the edge of the calibrated images have such data numbers, which should be ignored. Therefore, for each of the imagers, an image is given outlining the usable field of view. These images are called g-001 (SLI), g-002 (MRI), g-003 (HRI), g-004 and g-005 (half HRI frames). The edge of the original field of view is where data numbers in that image are near 1000. The location of reliable data far from the edges have data number 10000 in these five images. As the edges are approached, reliability decreases due to imperfections at the edges of the CCD, indicated as a decrease if data number from 10000 to 1000 in these five images. One pixel off the edge these data numbers reach zero, indicating zero reliability.
- 5) I scaled data numbers of MRI and HRI images so that simultaneous images give similar data numbers in overlapping areas. Based on my investigations of descent images, I multiplied HRI data numbers by a constant 0.424, and I divided MRI data numbers by  $0.989 + 0.0000032 * (T - 180)^2$  where T is the CCD temperature in K. Then the data numbers were converted in DN/ms by dividing by the exposure time in ms. Finally, I divided the data numbers by



( $742+0.13*T$ ) to convert them to I/F where  $\pi*F$  is the solar flux at the top of Titan's atmosphere. These numbers are multiplied by 100000 and rounded to integers. Thus, in the calibrated images, a data number of 100,000 corresponds to  $I/F = 0.1$ . For the MRI and HRI images after landing, data numbers are scaled down by a factor of 4 in order to avoid saturation, so that a data number of 25,000 corresponds to  $I/F = 0.1$ .

The sensitivity estimate of  $742+0.13*T$  derives mostly from laboratory measurements with a small adjustment for non-constant intensity of the integrating sphere.

- 6) The wavelength distribution of photons received from a gray surface at the top of Titan's atmosphere is estimated for two temperatures in the following table, normalized to a sum of unity. For other CCD temperatures, linear interpolation in temperature is sufficient. Since the CCD temperature decreased from 260 K during the beginning of the descent to 170 K at landing, the early images probed Titan's light centered at 785 nm, the low-altitude ones at 769 nm wavelength. Because the illumination near the surface has steep spectral signatures from methane absorption in the atmosphere, small shifts of wavelength can significantly change the probed reflectivity.

Wavl (nm)	170 K	270 K	CCD temperature
610	0.00001	0.00000	
620	0.00003	0.00002	
630	0.00018	0.00006	
640	0.00224	0.00049	
650	0.01506	0.00444	
660	0.03212	0.01740	
670	0.04253	0.03223	
680	0.04800	0.04045	
690	0.05105	0.04470	
700	0.05066	0.04523	
710	0.05006	0.04521	
720	0.04919	0.04479	
730	0.05002	0.04596	
740	0.04991	0.04648	
750	0.05019	0.04763	
760	0.04795	0.04665	
770	0.04370	0.04344	



780 0.04054 0.04089  
790 0.03764 0.03846  
800 0.03588 0.03705  
810 0.03461 0.03632  
820 0.03349 0.03573  
830 0.03218 0.03509  
840 0.03021 0.03367  
850 0.02619 0.03000  
860 0.02404 0.02792  
870 0.01981 0.02394  
880 0.01729 0.02147  
890 0.01492 0.01911  
900 0.01278 0.01692  
910 0.01081 0.01475  
920 0.00908 0.01287  
930 0.00779 0.01147  
940 0.00654 0.01000  
950 0.00551 0.00877  
960 0.00459 0.00769  
970 0.00377 0.00669  
980 0.00302 0.00577  
990 0.00231 0.00485  
1000 0.00167 0.00399  
1010 0.00110 0.00312  
1020 0.00067 0.00239  
1030 0.00035 0.00176  
1040 0.00015 0.00126  
1050 0.00007 0.00087  
1060 0.00005 0.00062  
1070 0.00003 0.00044  
1080 0.00001 0.00034  
1090 0.00000 0.00026  
1100 0.00000 0.00018  
1110 0.00000 0.00011  
1120 0.00000 0.00004

Mean 768.6 786.5 nm  
RMS 77.0 83.9 nm

-----  
Further notes....



The TIFF images were derived from the PGM images using IDL\*. For Photometry the PGM's should take precedence. The TIFF images are provided to allow ready viewing on different platforms. The TIFF images are stretched (min to max) from 0 to 65535, or 16 BitsPerPixel (BPP), therefore they do not retain the photometry calibration. No pixels were discarded.

ASCII tabular data is also provided in the 'TABLE' directory. The contain the pixel values from the PGM images by row and column in ASCII form.

The filenames contain the beginning observation mission time in seconds, and the observation altitude in Kilometers (KM) or meters (M) as specified.

The G\_IMAGE\_MAPS provide the near-edge intensity rolloff for the various image types, as well as for the uncompressed Side Looking Imager strips.

\*Interactive Data Language V6.1

## B Compression

### B.1 My thoughts on the reconstruction of downlinked DCT coefficients

$$R(DQ(Cx)) = a \quad (1)$$

$$a = R(D\chi) \quad (2)$$

$$DQ(Ca) = b \quad (3)$$

From this I would naively derive

$$DQ(C \cdot (a - (b - a))) \approx a, \quad (4)$$

but this seems not to be true.

$$DQ(Cb) = b \quad (5)$$

$$DQ(Cx) = x \quad (6)$$

$x$ : Sseeked unknown (float) image that compresses into the true downlinked DCT coefficients



$C$ : DCT compression (linear operation)

$Q$ : Quantization (non-linear operation)

$D$ : DCT decompression (linear operation)

$a$ : A image, decompressed from true coefficients and rounded to 15 bit

$\chi$ : True downlinked coefficients

$R$ : Rounding to 15 bit (non-linear operation)

$b$ : Result of re- and de-compression of  $a$

We are looking for something that redecompresses into itself and that rounded to 15 bit gives the A image. So:

$$DQ(Cx) = x \quad (7)$$

$$R(x) = a \quad (8)$$

The former condition is fulfilled by  $b$ , the latter by  $a$ . An image that results from a decompression, like  $b$ , should always redecompress into itself. So, what could be a procedure?

1. Start with  $x = b$ .
2. Tear  $x$  a little bit towards  $a$ , take into account the rounding to 15 bit in some way, maybe add some randomness, to obtain  $y$  (which does *not* redecompress into itself).
3. Get new value  $x = DQ(Cy)$  (which *does* redecompress into itself).
4. Repeat from step 2.

This could work if the new  $x$  is closer to  $a$  than the old one, but is it? Nope!

But is  $b$  of any significance at all? It deviates from  $a$  in an arbitrary direction determined by the rounding.

**Note:** All weighted means (with positive weights) of  $a$  and  $b$  redecompress into  $b$ ! This holds even with negative weight for  $a$ , unless it gets  $\leq -0.5$ , then the deviation from  $a$  increases. The deviation also increases with any negative weight for  $b$ .



## B.2 Erich's explanation of compression (and my reflections)

I am also very grateful that you care to give the exact DISR imager data to the future of mankind. I can give you some details about the compression. Perhaps this can help you to understand why the compression and decompression does not seem to behave as it should. My response may be completely different than Lyn's which hopefully means that we are complementing each other perfectly. The compressor is not easy to understand, so don't worry if you have not figured it out. I went into the details of the compressor so I could get the most of the available data. If you want to preserve complete information, you probably have to do the same. If you just want to preserve most of the information, it gets so much easier since you do not have to read any further.

The compressor on Huygens gets an image with 8-bits pixels from the imager via the square-rooter. It gets the instruction to reduce the number of bits by a certain factor. It tries to do it, I think, by trial and error and then determines a certain pair of quantization and threshold to get somewhat close to the desired compression ratio.

It does a DCT which creates 256 real coefficients in each 16x16 block. It then combines coefficients between a certain range into the same transmitted symbol or sequence of bits, and the width of the ranges vary among the 256 coefficients. You can call this symbol a number, but I call it a range because it is a more accurate description of the information it contains. For example, one transmitted symbol may indicate that the coefficient was between -2.5 and 2.5. One symbol may indicate that another coefficient was between -0.5 and 0. One symbol may indicate that the coefficient was between 0.5 and 2.5. Actually, this is true for 252 of the 256 coefficients. The coefficient 0,0 (average of all 256 pixel values) was transmitted accurately while the three highest-frequency coefficients were ignored.

[The width of the ranges varies among the coefficients? I was thinking it is determined by the quantization value, the same for all coefficients.](#)

[With which precision was the average transmitted?](#)

The decompression software that we got (is this the one from Peter Rueffer?) rounded those three ranges to 0, 0, and 2, respectively. Then it did the inverse DCT. Finally it rounded the pixel values to integers. The rounding in two steps is not optimal since it creates a biased image. The loss of information about the width of the range is significant. In science, the mean of a measurement does not mean much as long as you do not know the accuracy. This software rounded ranges of -2.5 to 2.5, -0.5 to 0, and 0 to 0.5 to the same integer of zero. This loss of information means that you may not be able to distinguish between three possible different interpretations based on the decompressed images but you might



be able to distinguish them based on the data stream. The decompression software is appropriate in a sense that it creates errors that are smaller than compression errors or CCD noise. For a purist, it is imperfect since compression errors and CCD cannot be avoided while any further errors can be avoided.

It seems inconsistent that the range 0.5 to 2.5 is rounded to 2 and not to 1. Is there a rule? Other wise, we would need a list of all ranges and rounding values. Maybe it's a typo? Probably not, is the same ios stated later again.

How is a range  $-2.5$  to  $2.5$  possible considering the table of limits below. Wait, no, its clear, just "not transmitted" can impose that range. But... that would require a  $q = 6$  in Erich's terms, and that should not exist as they should come in powers of two.

To work with the ranges of DCT coefficients instead of numbers I first had to estimate what the quantization and threshold was because this information was not transmitted to Earth. I created several methods of estimating them based on the transmitted data and tested them on several simulated images. I finally found a method that always gave the correct quantization and gave the correct threshold in more than 95 % of the tests and was off by 1 unit in the remaining tests. This method created the list I gave you.

I then did a reverse DCT on the 606 images using the midpoint of each range, i.e. 0, -0.25, and 1.5 for the three coefficients mentioned above. I rounded the resulting pixel values to 7 decimal bits which produced the 15-bit images you have. If you do a DCT on these images you should get coefficients that are very close to multiples of quarter integers. Because ranges end at integers or half-integers, mid-points must be multiples of  $1/4$ . If you believe my estimated quantization and threshold and you know how the DCT is implemented, you can uniquely determine the range based on the quarter-integer coefficients. Thus, the 15-bit images have essentially all the information as the original data stream. You could in principle determine all possible input images that resulted in the transmitted data. My 15-bit images would be close the the average of all those possible input images. The original decompression software gives a possible image, but not so much the average of possible images because the rounding in two steps introduces a significant bias.

There are several important points here:

- If Erich did a reverse DCT on the 606 images, he must have the data streams of them. **I want it!** (and also his software, along with the original compression software)
- A DCT of the 15-bit images should give coefficients at the midpoint of the original ranges, thus applying the compression with the correct quantization should reproduce the original data stream.





- Couldn't this be used to check the quantization value by comparing original and reproduced data stream and determining it unambiguously by try and error?
- I understand now why compressing and decompressing a 15-bit image does not always reproduce the image: Erich's decompression uses the midpoint of ranges while the decompression I have rounds the midpoint to integers before the DCT.
- I think the 15-bit image is not just close to the average of all possible images, it equals it exactly (within 15-bit precision), because the reverse DCT of the midpoints of the ranges without rounding is a linear operation.
  - A useful product could be de-square rooted 15-bit images.
  - An error range for each pixel value could be estimated by randomly generating possible images.
- I would not say that the original decompression gives a possible image, because if you compress it, it results in a different data stream, so it is not compatible with the original data stream.

Basically, if the quarter-integer coefficient is zero, the absolute value of this amplitude was below the threshold minus 0.5. Otherwise, the sign and the rounded integer was transmitted for quantization 1. So +0 and -0 were two different transmitted symbols (bits). For quantization 2 the limits were 0.5, 2.5, 4.5, ... For quantization 4 the limits were 1.5, 5.5, 9.5, etc.

I do not understand a couple of points here:

- “the threshold **minus** 0.5” is probably a typo.
- Something I *do* understand after some thought: For quantization 1, if the coefficient value was -0.4, sign “-” and rounded integer “0” would be transmitted. For quantization 2, nothing would be transmitted.
- So, I understand that these are the range limits:

$q$	Limits						
1	-2.5	-1.5	-0.5	0	0.5	1.5	2.5
2	-4.5	-2.5	-0.5		0.5	2.5	4.5
4	-9.5	-5.5	-1.5		1.5	5.5	9.5

The case  $q = 1$  seems to be special at first, but it may be not. If we determine the threshold magnitude below which a coefficient is not transmitted as  $\frac{q-1}{2}$ , this fits all three cases.



- My compression routine prints out the message:

```
q      : quantization: 0 = 0.125
                        1 = 0.250
                        2 = 0.500
                        3 = 1.000
                        4 = 2.000
                        5 = 4.000
```

How could this relate to Erich's scheme? I suppose that the compressor does its computations with two byte integers, so the one byte input values are multiplied by eight before any computation. Erich's quantization values refer to these integers, while the message of the compressor software refers to the original one byte values.

In principle, if a quadruple of coefficients got transmitted as non-zero, one knows that one of the four coefficients of the quadruple must have been at least at the threshold or the quadruple would not have been transmitted. This can be additional information if the threshold cutoff falls within a quantization interval. However, even then, usually two or more coefficients were of the same absolute size and it is unknown which of these coefficients was the one that triggered the threshold, or whether several or all coefficients triggered the threshold. It would have been very complicated to implement such a complicated logical structure in further processing, so I think I deleted this information. I doubt that anybody wants to dig so deeply into the compression scheme.

This can (and should) be used in a forward way to check if an image is compatible with the data.

If one of the coefficients of a quadruple is at least the threshold, the whole quadruple is transmitted. I suppose that coefficients below the minimum absolute quantization value are still dropped.

Everybody who looked at the DISR images noted that they are not a good estimate of how Titan looks like. Most noticeable were the sharp boundaries at 16x16 pixel blocks and the high-frequency features that change shape from block to block. So I created images that come closer to what nature looks like by varying each coefficient within the allowed range. I tried a few methods. Some still had high-frequency features that obviously came from the compression, but at lower amplitudes. Some were so smooth that everything artificial looking was obliterated. Marty liked the latter ones, and those were put into the PDS archive. At some point they are called E-images.

After Marty retired I continued to work on mosaics with better spatial resolution. I found that it was better to use images that were not smoothed as much. They



show more real features. They also show a little of compression artifacts. However, since I do mosaics by averaging several images, the real features remain while the compression artifacts mostly average out.

[The mosaic that I have on this polar exponential grid is probably an early version, so I would like to have the latest one.](#)

I also investigated the distribution of coefficients. For example, The data of one image may tell you that 98 % of the high-frequency coefficients were between -0.5 and 0.5. Based on CCD noise, one might calculate that 98 % of these coefficients would be between -0.5 and 0.5 if the image scene were completely bland. This tells you that real features must be generally causing coefficients perhaps up to  $\pm 0.2$ , and only CCD noise brings them up beyond 0.5. So it makes sense to display this image with high-frequency coefficients up to around  $\pm 0.2$  even if this is clearly outside the possible range of 0.5 to 2.5 or -2.5 to -0.5 for some coefficients. Of course, the original software displayed 98 % of these coefficients as 0 and 2 % as  $\pm 2$ . This creates a few high-frequency features that have nothing to do with reality. My first estimate of  $\pm 1.5$  from the middle of the range is not much better (the 15-bit images).

[That implies that one should also model noise when trying to model a true image and checking for compliance with the data. But one might need to check several realization of noise and only accept images where a certain fraction yields compliant images.](#)

I also investigated the different blocks and realized that there were a few blocks for each imager that showed a particularly bad behavior with high-frequency patterns. In each case there was one pixel in the block that got bad during the cruise or that had a strong flatfield feature that shifted slightly so that the flatfield Huygens had available was not quite right. I could create much better images by using different methods for different blocks. This way, many more real images could be shown in most blocks while the few bad blocks could be smoothed enough so that the compression artifacts would not dominate. It turns out that a handful of bad pixels and Marty's desire to suppress compression artifacts created a set of DISR images in the current archive that lacks much of the fine detail that was imaged. Chuck submitted the improved E-images to the PDS two years ago, but the public has not seen them. Included there is a description of my other improvements with respect to the available PDS version.

[This is the 2017 update! So I should replace my E-images.](#)

So, Bjoern, you can see that preserving the DISR data with all its details has not been finished and your work is important to get it done.



*END OF PUG*