About Jupiter's Reflectance Function in JunoCam Images

Europlanet

European Planetary Science Congress 2017
Riga
2017-09-20

G. Eichstädt (1), G. S. Orton (2), T. Momary (2), C. J. Hansen (3), and M. Caplinger (4)

(1) Independent scholar, Stuttgart, Germany (ge@junocam.pictures)
(2) Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California USA
(3) Planetary Science Institute, Tucson, Arizona, USA
(4) Malin Space Science Systems, San Diego, California, USA
Motivation:
How do we get from a decompanded JunoCam image to an enhanced JunoCam image?

decompanded, square-root encoded  enhanced
Notion: decompanding
JunoCam encodes CCD-readout data in a non-linear way

- raw encoding enhanced
- JunoCam's decompanding function, absolut linear and square-root relative to raw
- decompanded and enhanced
How do we get from a decompanded JunoCam image to an enhanced JunoCam image?

**Approach: Gamma stretch**

- A "gamma-stretch" means applying the gamma-th power [usually to linear data]
- \( f(x) = x^\gamma \)
Stretching with $\gamma = 0.5$ equivalent to square-root encoding

decompanded, square-root encoded

gamma-stretched (trivial)
Stretching with $\gamma = 1.0$

decompressed, square-root encoded

gamma-stretched to linear radiometric values
Stretching with gamma = 2.0

decompanded, square-root encoded
gamma-stretched
Stretching with gamma = 4.0

decompanied, square-root encoded

gamma-stretched
Stretching with gamma = 4.0 compared to desired enhancement

Conclusion: illumination-adjustment required!
Basic Notions:

Cosine of Solar Incidence Angle

- Solar incidence angle: angle between surface normal and vector to point light source

\[ v := \vec{N} \cdot \vec{L} \]
Basic Notions:

**Cosine of Emission Angle**

- Emission angle: angle between surface normal and vector to eye

\[ \eta = \vec{N} \cdot \vec{E} \]
Cosines of Solar Incidence and Emission Angle
How does the appearance of a surface point change as a function of $\iota$ and $\eta$?
Global answers are given in terms of a reduced form of a BRDF

- BRDF: Bidirectional Reflectance Distribution Function
- Reduced to $\iota$ and $\eta$, in contrast to general BRDF, which is described in terms of vectors to point light and eye.
Example of a reduced BRDF

- But how does it apply to JunoCam images of Jupiter?
- How do we get $\iota$ and $\eta$ for a given pixel of the decompanded image?
Decompanded image, solar incidence data, emission data

- Solar incidence data and emission data are provided in terms of the cosines $\iota$ and $\eta$ of the angles, calculated by intersecting sun and observer vectors with Jupiter MacLaurin spheroid.

![Decompanded image](image1.png) ![Cos of solar incidence angle](image2.png) ![Cos of emission angle](image3.png)

decompanded image, square-root encoded  
cos of solar incidence, square-root encoded  
cos of emission angle, square-root encoded
For each pixel in decompanded image:

Lookup or calculate cosines of solar incidence angle and of emission angle [encoded as grey values, below]

- Jupiter's one-bar MacLaurin spheroid, as well as SPICE positions of spacecraft, sun, and Jupiter are used for the calculation.

![decompanded image](image1)
![cos of solar incidence angle](image2)
![cos of emission angle](image3)

decompanded image, square-root encoded

cos of solar incidence, square-root encoded

cos of emission angle, square-root encoded
For each pixel in decompanded image:
Use cosines of solar incidence angle and of emission angle to look-up or calculate model brightness

model brightness as a function of $\iota$ and $\eta$

EPSC
2017
RIGA
Result: Jupiter reference model
Divide decompanded image by Jupiter reference model

\[
\frac{\text{decompanded image}}{\text{reference model}} = \text{illumination-adjusted image}
\]

\[(\gamma = 1, \text{for all three images})\]
Decompanded colors divided by masked polynomial, stretched by $\gamma = 0.5$

$5^{th}$ degree polynomial of 2 parameters visualized as a grey scale image, masked by text

weighted decompanded colors divided by masked polynomial, and stretched by $\gamma = 0.5$
Decompanded colors divided by masked polynomial, stretched by $\gamma = 4.0$
Approach seems to work; how do we obtain a reduced BRDF?

Simplest non-trivial BRDF: Lambert model

\[ f(\iota, \eta) := \iota \]

The Lambert illumination model is linear with the cosine of the solar incidence angle, and ignores the emission angle.

weighted *decompanded colors* divided by Lambert model, and stretched by \( \text{gamma} = 0.5 \)
The Lambert illumination model is linear with the cosine of the solar incidence angle, and ignores the emission angle.

Observation: Terminator too bright, zone near terminator too dark.
Modified Lambert model, and gamma-stretch:
Adding a constant darkenes terminator

\[ f(\iota, \eta) := 0.05 + 0.95 \iota \]

Observation: Zone near terminator still too dark, part of limb too bright
modified Lambert model compared to desired enhancement
both stretched with gamma = 4.0

modified Lambert desired enhancement

Conclusion: better, but more sophisticated illumination model desirable.
General approach: Monte-Carlo method

- Select random samples, each consisting of decompanded value $v$, $\imath$, and $\eta$

$\nu$  $\imath$  $\eta$

decompanded image, square-root encoded  cos of solar incidence, square-root encoded  cos of emission angle, square-root encoded
General approach: Monte-Carlo method

- Select random samples, each consisting of decompanded value $v$, $\iota$, and $\eta$, exclude terminator and limb.

| $v$ | $\iota$ | $\eta$ |
|--------------------------------|
| decompanded image, square-root encoded | cos of solar incidence, square-root encoded | cos of emission angle, square-root encoded |
General approach: Monte-Carlo method

- Select random samples, each consisting of decompanded value $v$, $\iota$, and $\eta$, exclude small $\iota$, and $\eta$.

- Images:
  - $v$: decompanded image, square-root encoded
  - $\iota$: cos of solar incidence, square-root encoded
  - $\eta$: cos of emission angle, square-root encoded
1 million Monte-Carlo samples distributed over 27 Perijove-06 images (TDI 2)
Best fit polynomial of degree 0 [constant]
by least square sums of relative errors

\[
\sum_{i=1}^{n} \left[ \sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot v_i^j \eta_i^k \cdot \left( \frac{1}{v_i} - 1 \right) \right]^2 = \min, \ m=0
\]

Note: Best-fit polynomials are calculated for each of the three color channels separately.
Best fit polynomial of degree 1 [plane] by least square sums of relative errors

$$\sum_{i=1}^{n} \left[ \sum_{j=0}^{m} \sum_{k=0}^{j} a_{j,k} \cdot \psi_i^j \cdot \eta_i^k \right] \cdot \psi_i - 1 \right]^2 = \min, \ m = 1$$

- cosine of solar incidence angle
- weighted decompanded colors divided by model, and stretched by gamma = 4.0
Best fit polynomial of degree 2 [paraboloid] by least square sums of relative errors

\[
\sum_{i=1}^{n} \left( \frac{\sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot \psi_i \cdot \eta_i^k}{\psi_i} - 1 \right)^2 = \text{min}, \ m=2
\]

cosine of solar incidence angle

weighted **decompanded colors** divided by model, and stretched by \( \text{gamma} = 4.0 \)
Best fit polynomial of degree 3 by least square sums of relative errors

\[
\sum_{i=1}^{n} \left( \sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot \nu_i^j \cdot \eta_i^k \cdot \nu_i - 1 \right)^2 \leq \min, \ m=3
\]

cosine of solar incidence angle

weighted \textbf{decompanded colors} divided by model, and stretched by \textbf{gamma} = 4.0
Best fit polynomial of degree 4 by least square sums of relative errors

\[
\sum_{i=1}^{n} \left[ \sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot \nu_i^j \cdot \eta_i^k \cdot \nu_i \right]^2 = \min, \ m=4
\]
Best fit polynomial of degree 5
by least square sums of relative errors

\[ \sum_{i=1}^{n} \left( \sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot \eta_{i,j} \cdot \xi_{i,k} \right)^2 - 1 = \min, \ m=5 \]

- cosine of solar incidence angle
- weighted decompanded colors divided by model, and stretched by gamma = 4.0
Best fit polynomial of degree 6 by least square sums of relative errors

\[ \sum_{i=1}^{n} \left[ \sum_{j=0}^{m} \sum_{k=0}^{m-j} a_{j,k} \cdot \eta_{i}^{j} \cdot v_{i}^{k} - 1 \right]^{2} \]  
\[ = \min, \ m=6 \]

- Cosine of solar incidence angle
- Weighted decompanded colors divided by model, and stretched by gamma = 4.0
- Oscillation of reference model is going to become evident near the terminator
Best fit polynomial of degree 5 looked best, add a small constant to remove bright terminator.
Use 5-th degree best fit polynomial of green channel for all three color channels

cosine of solar incidence angle

reduced BRDF

weighted *decompanded colors* divided by model, and stretched by \( \text{gamma} = 4.0 \)

Initial goal accomplished.
A Jupiter model: Reduced BRDF + Texture

decompressed image / reduced BRDF = gamma-stretched texture

can be written as

decompressed image / reduced BRDF = gamma-stretched texture

decompressed image / reduced BRDF = gamma-stretched texture
Does this Jupiter model remove all illumination-induced effects?

Compare two images reprojected to the same perspective

PJ6 #124 decompanded / reduced BRDF = gamma-stretched texture

PJ6 #128 / =

Observation: Textures differ by subtle haze features near the terminator
Repeat this with 15 Perijove-06 images
Animation makes it a little more evident
Ongoing: Do there exist ways to separate the haze layer from the cloud tops?
if (we_have_some_time_left_over)
{
    Show_Perijove_08_Flyby_Movie();
}
Anwer_Questions();
Return_to_Seat();