EPSC2017-517

About Jupiter's Reflectance Function in JunoCam Images

Europlanet

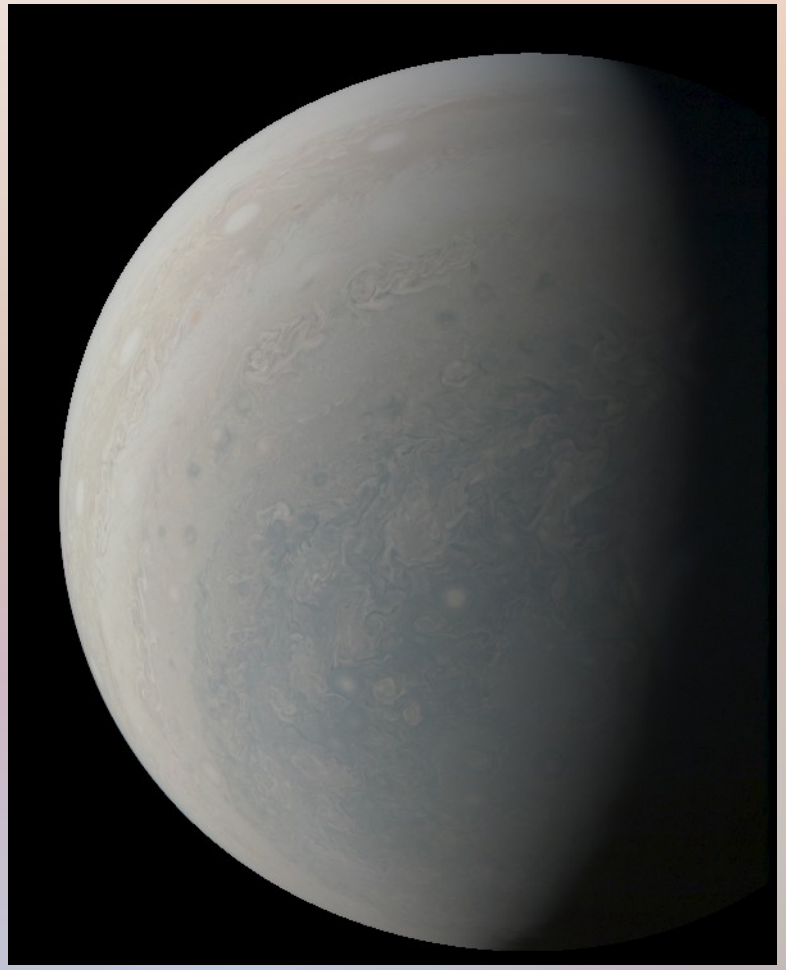
European Planetary Science Congress 2017
Riga
2017-09-20

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?



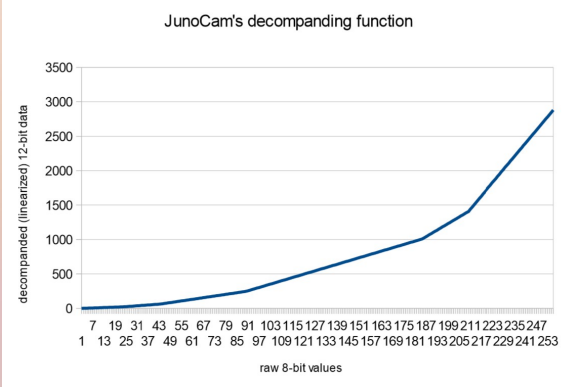
decompanded,
square-root encoded

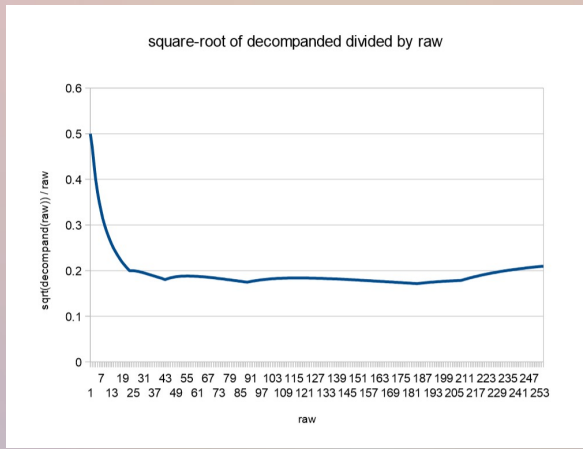
enhanced

Notion: decompanding

JunoCam encodes CCD-readout data in a non-linear way

raw encoding enhanced





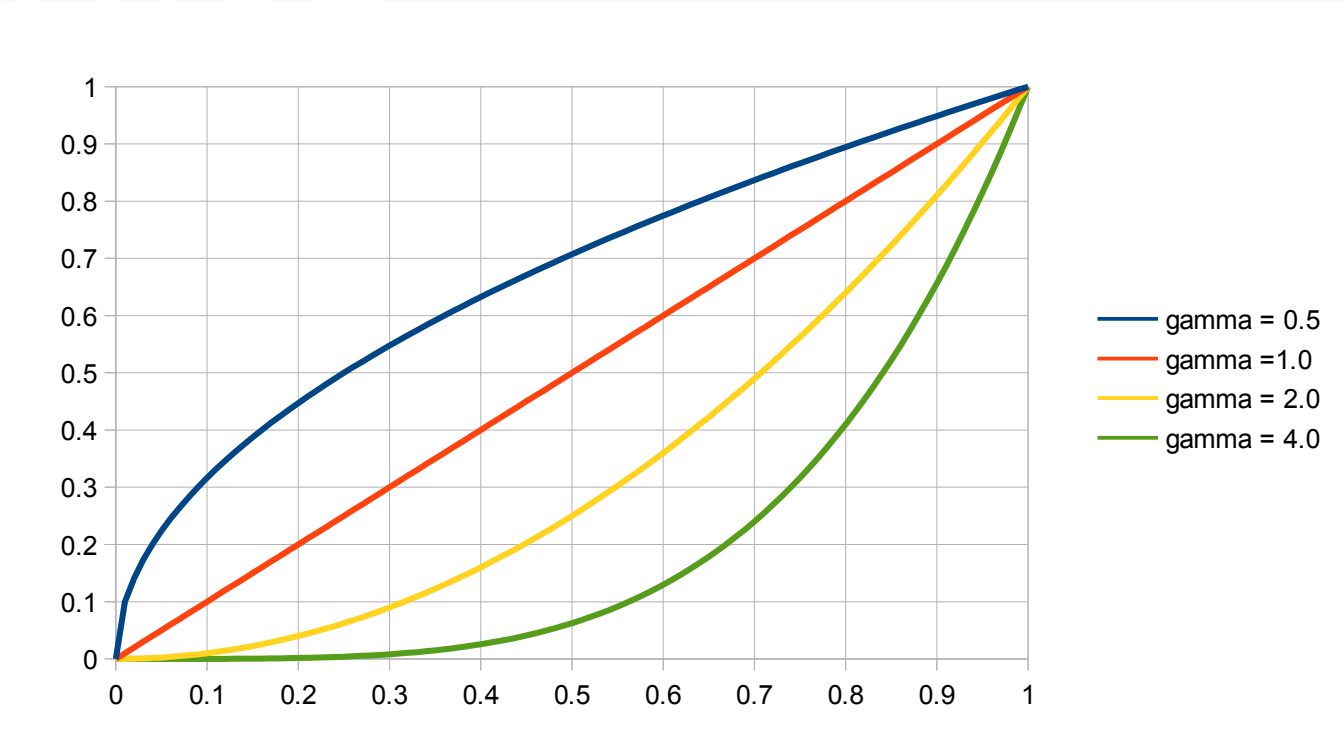
JunoCam's decompanding function, absolut linear and square-root relative to raw

decompanded and enhanced

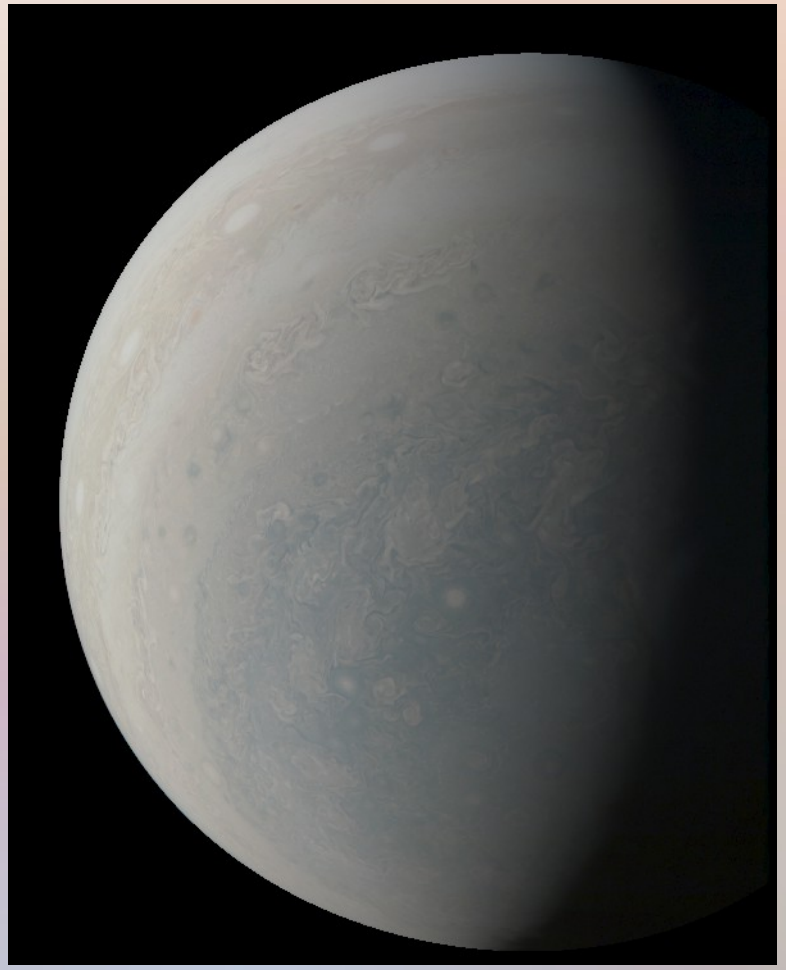
How do we get from a decompanded JunoCam image to an enhanced JunoCam image?

Approach: Gamma stretch

- A „gamma-stretch“ means applying the gamma-th power [usually to linear data]
- $f(x) = x^{\gamma}$



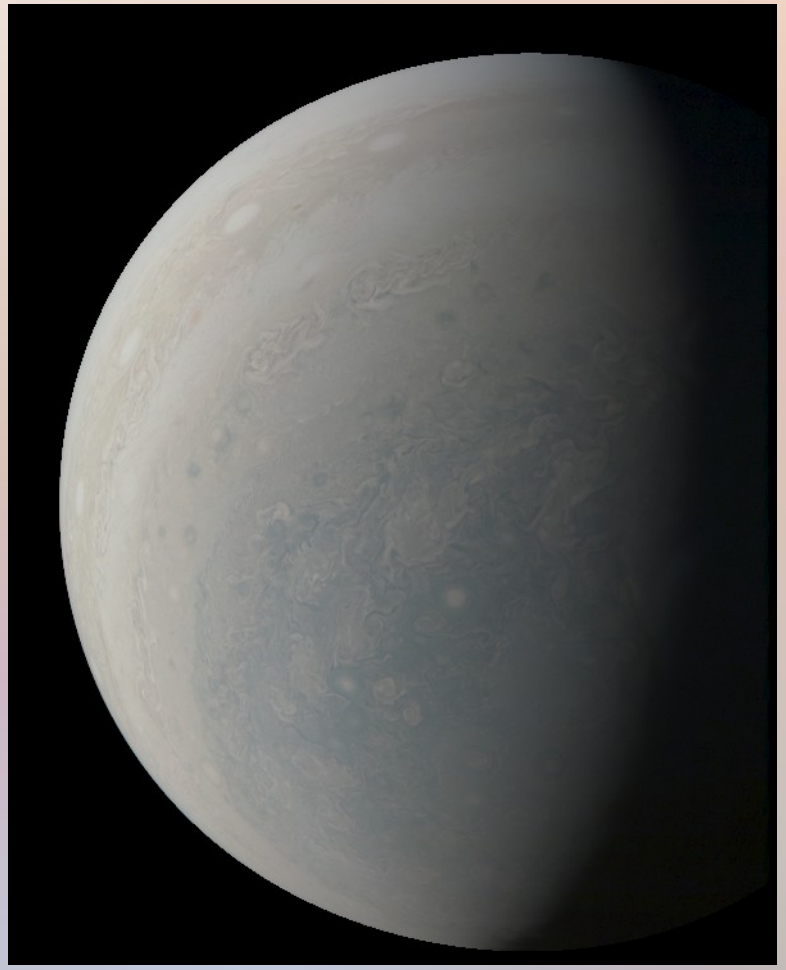
Stretching with gamma = 0.5 equivalent to square-root encoding



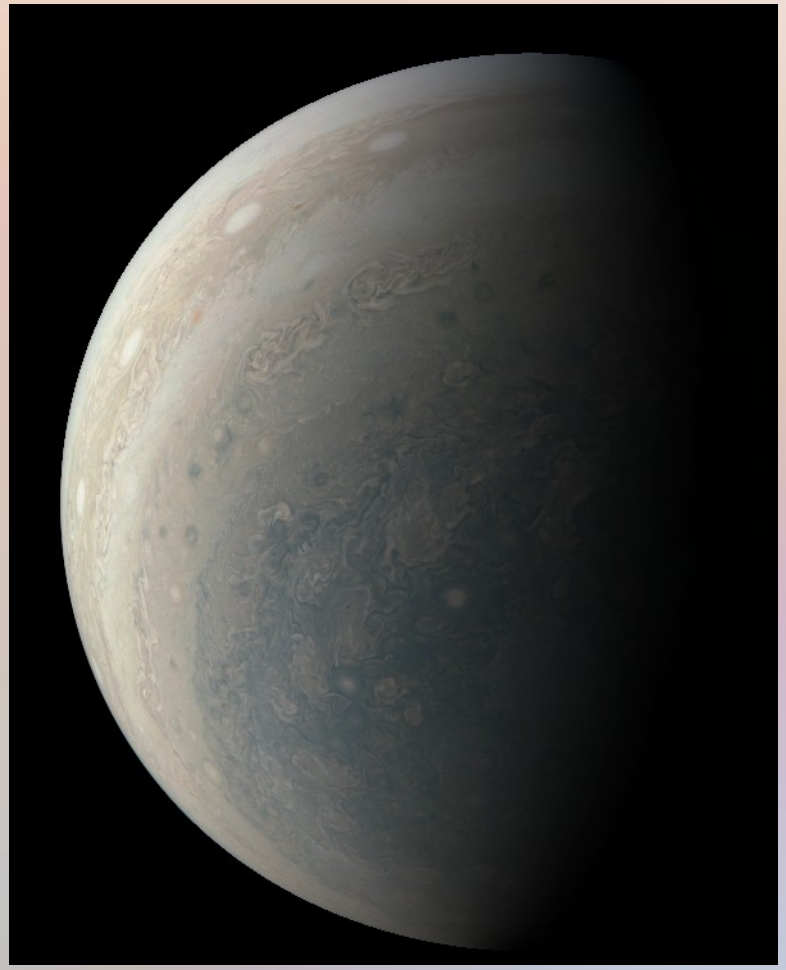
decompanded,
square-root encoded

gamma-stretched (trivial)

Stretching with gamma = 1.0

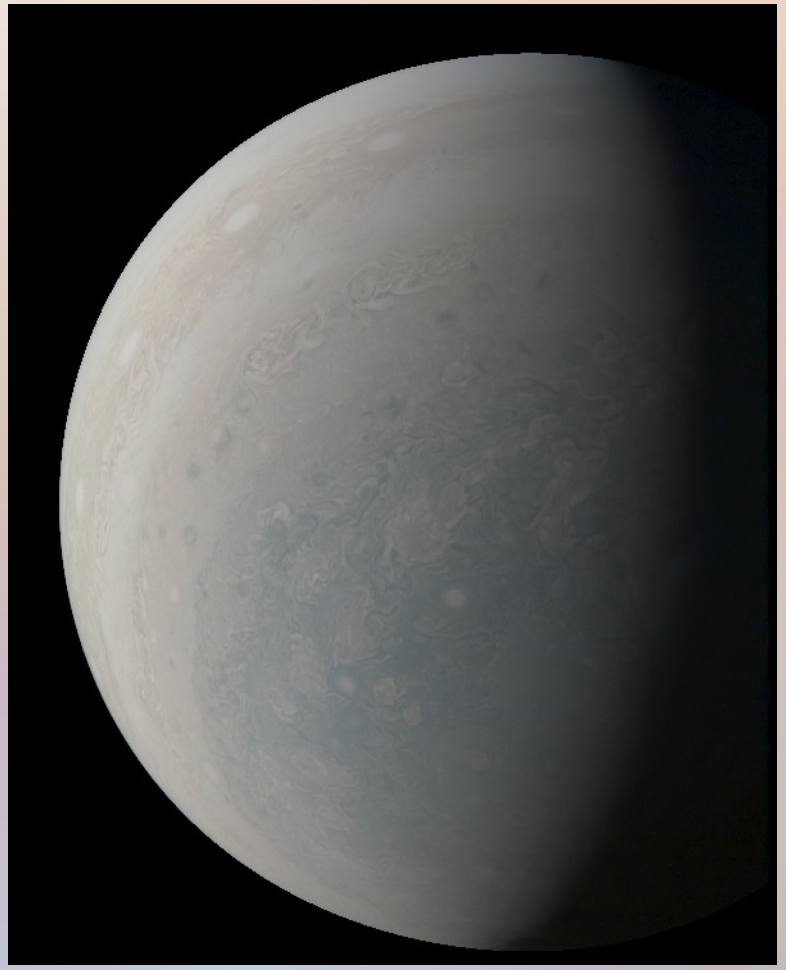


decompanded,
square-root encoded

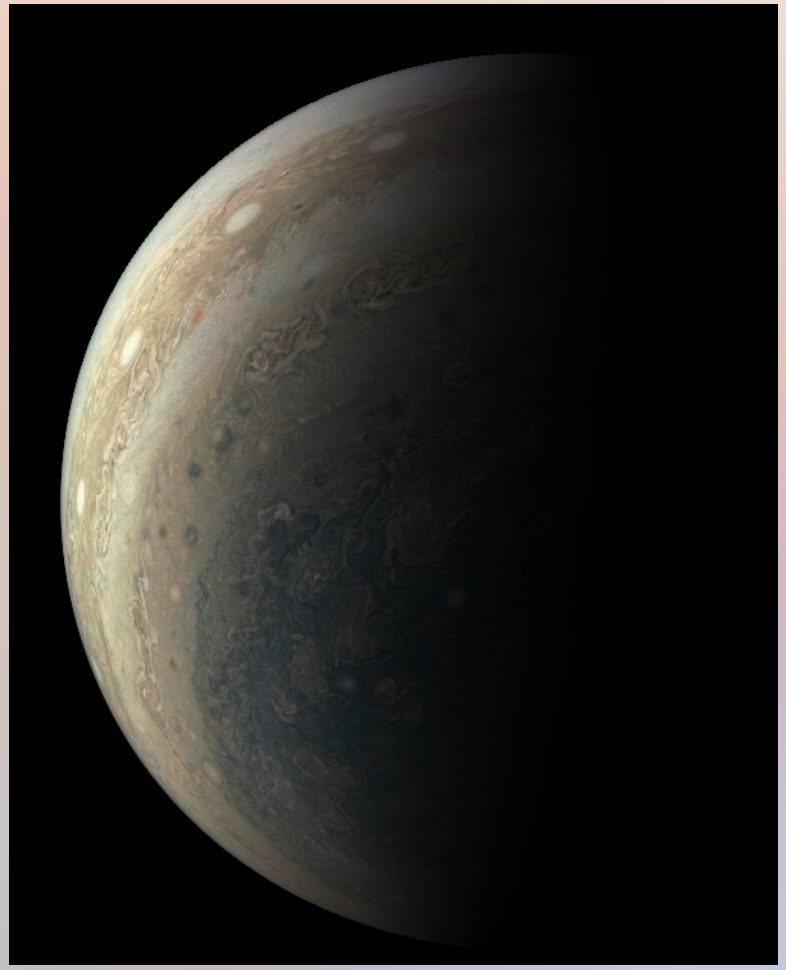


gamma-stretched
to linear radiometric values

Stretching with gamma = 2.0

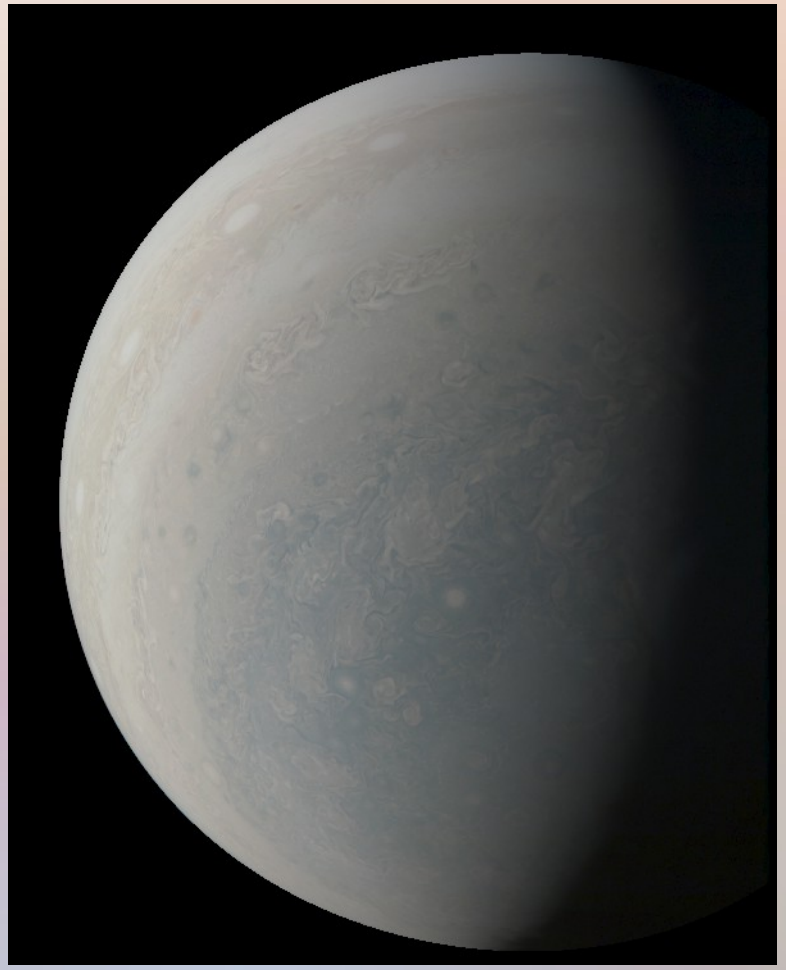


decompanded,
square-root encoded

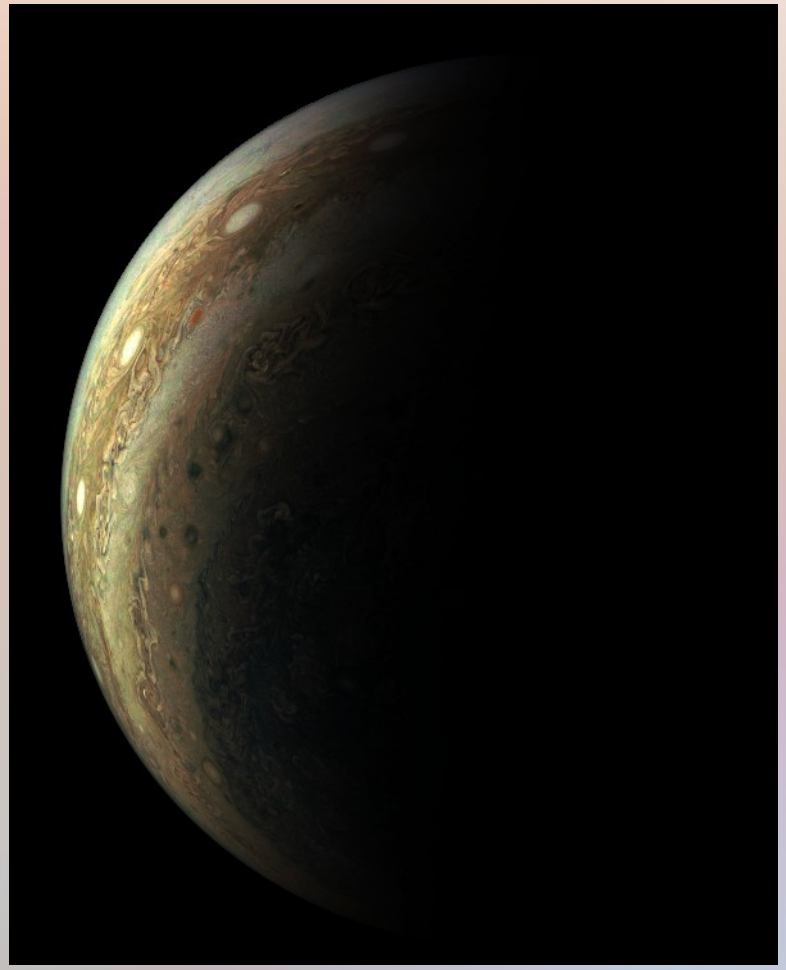


gamma-stretched

Stretching with gamma = 4.0

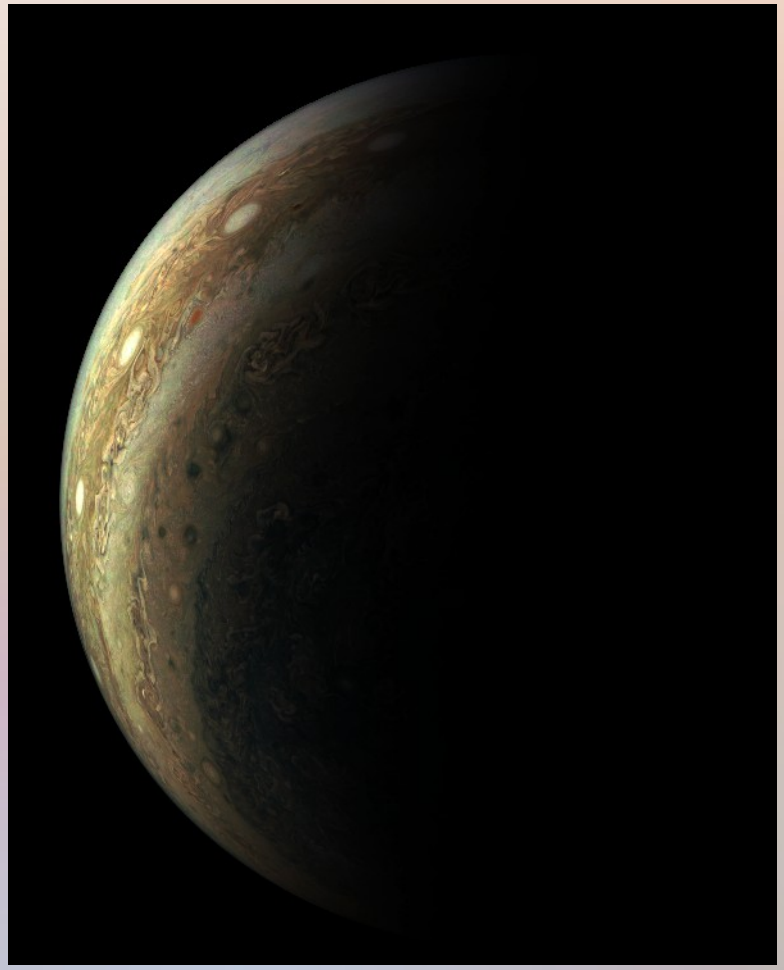


decompanded,
square-root encoded



gamma-stretched

Stretching with gamma = 4.0 compared to desired enhancement



gamma-stretched

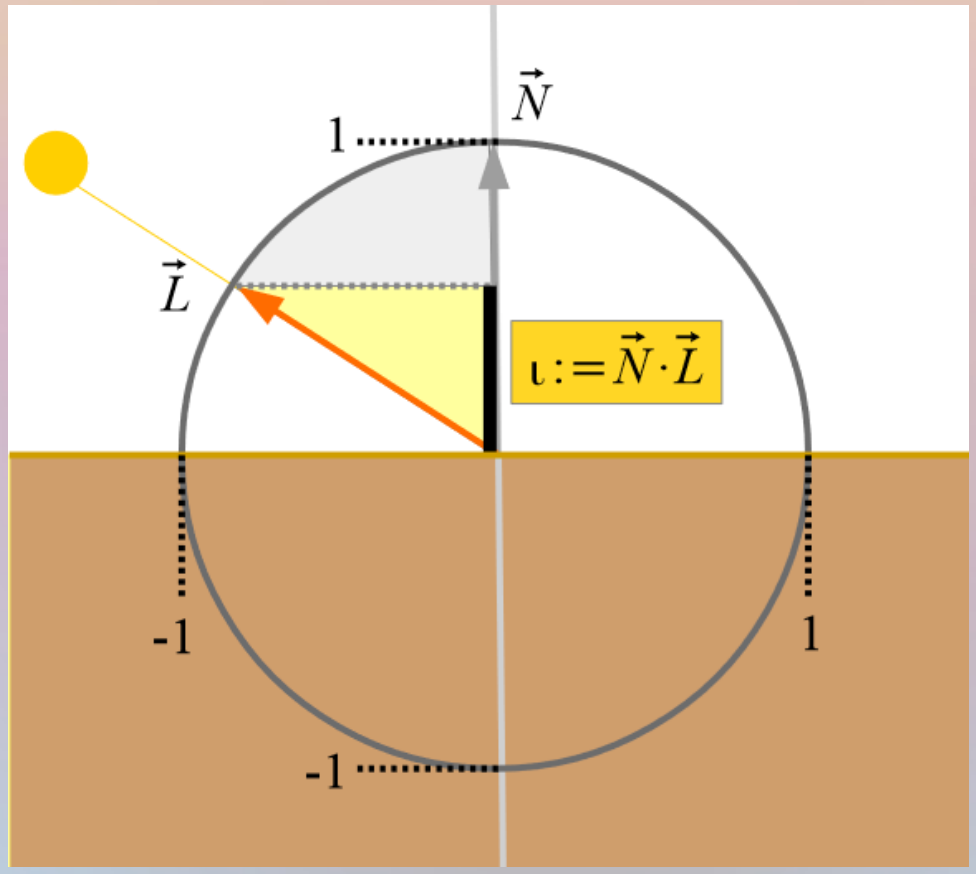
desired enhancement

Conclusion: illumination-adjustment required!

Basic Notions:

Cosine of Solar Incidence Angle

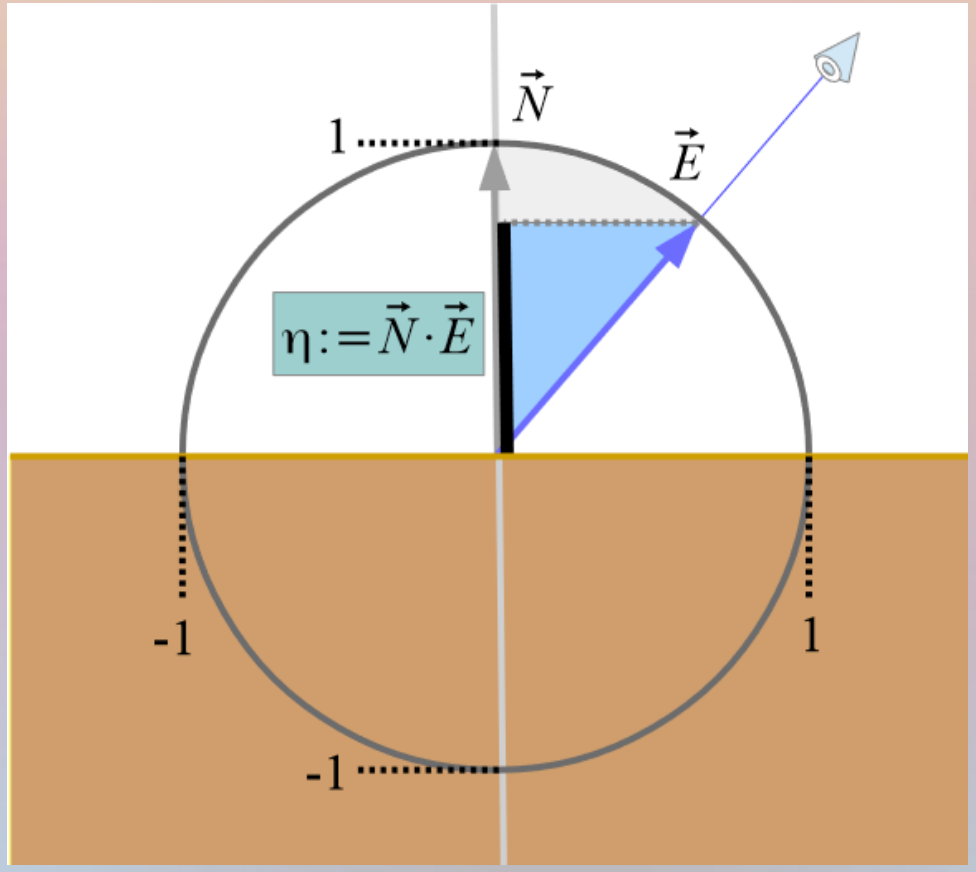
- Solar incidence angle: angle between surface normal and vector to point light source



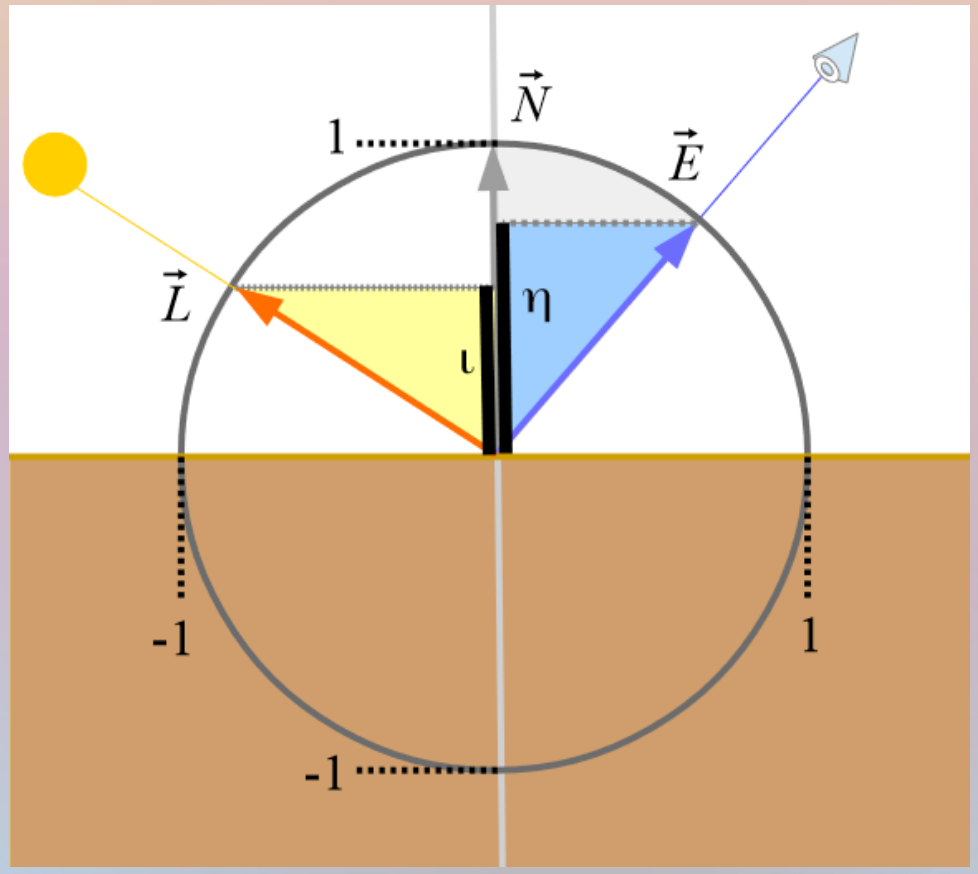
Basic Notions:

Cosine of Emission Angle

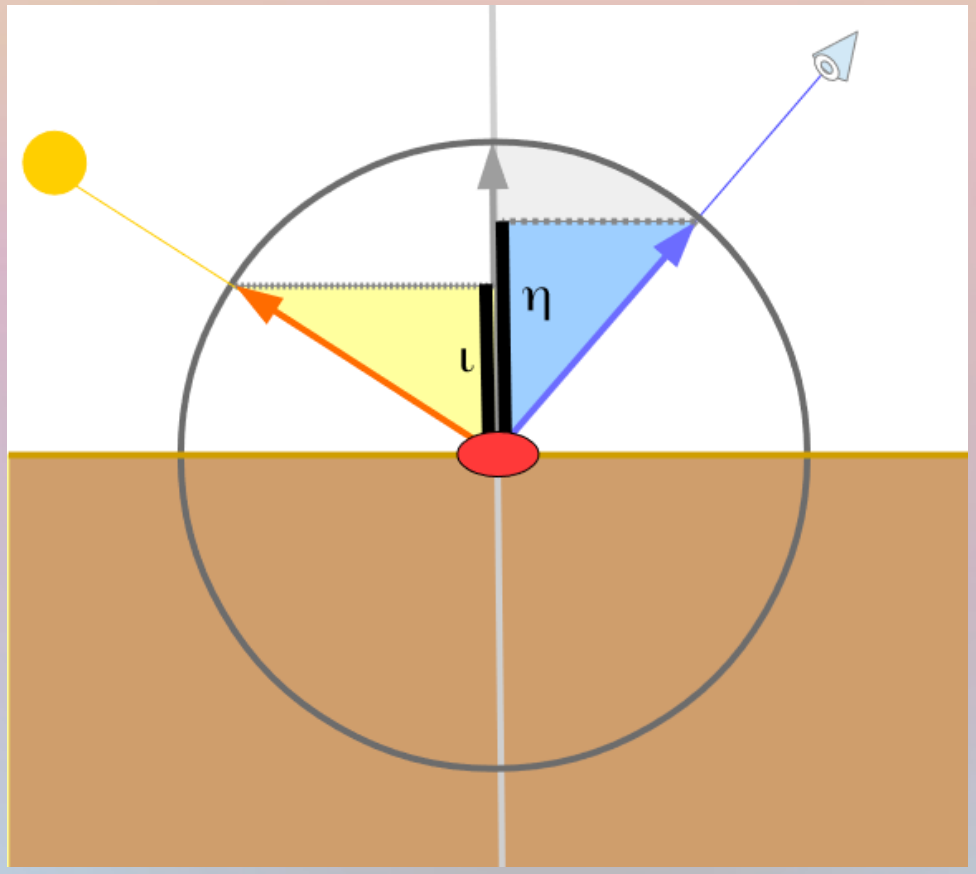
- Emission angle: angle between surface normal and vector to eye



Cosines of Solar Incidence and Emission Angle

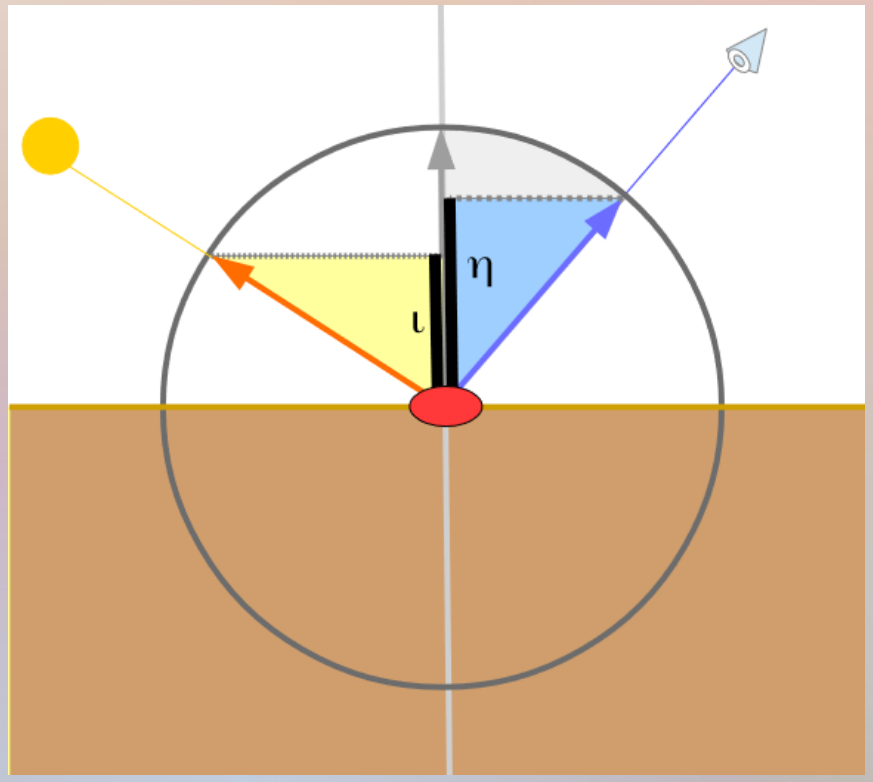


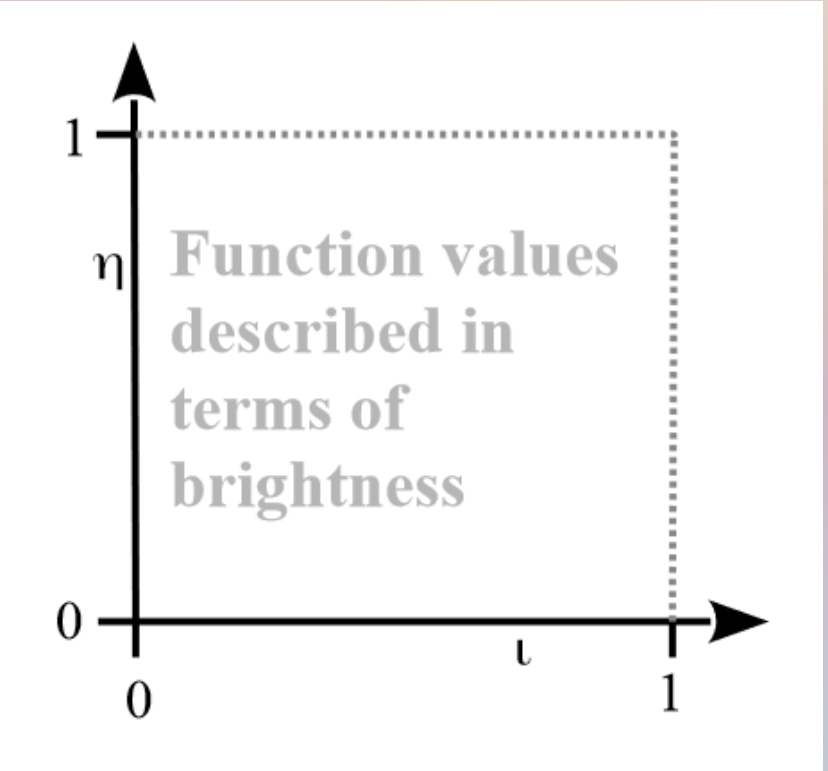
How does the appearence of a surface point change as a function of ι and η?



Global answers are given in terms of a reduced form of a BRDF

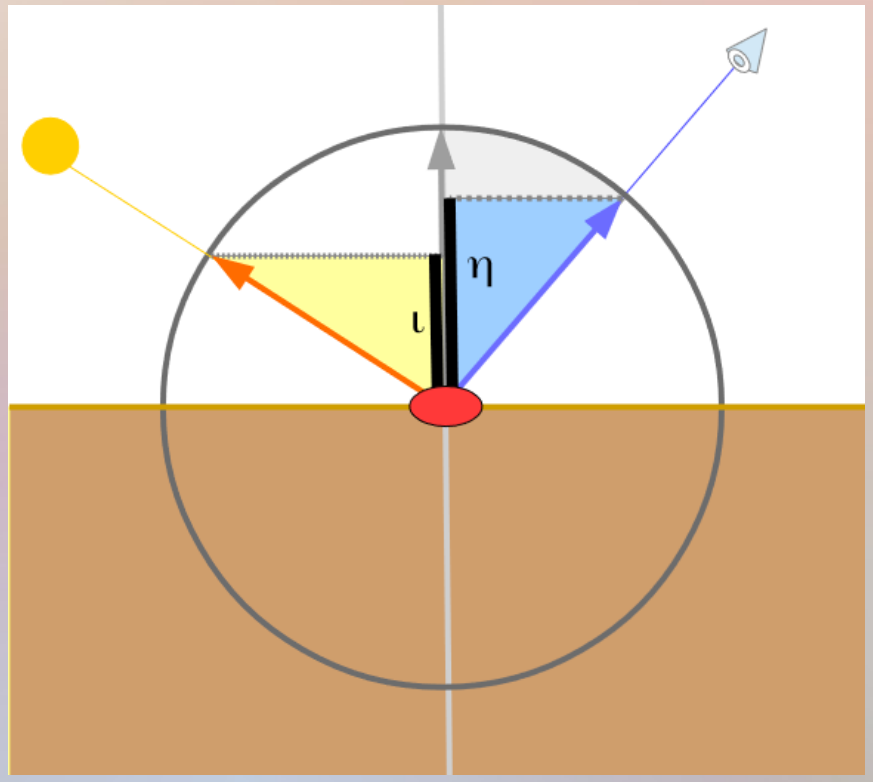
- BRDF: Bidirectional Reflectance Distribution Function
- Reduced to ι and η, in contrast to general BRDF, which is described in terms of vectors to point light and eye.





Example of a reduced BRDF

- But how does it apply to JunoCam images of Jupiter?
- How do we get ι and η for a given pixel of the decompanded image?

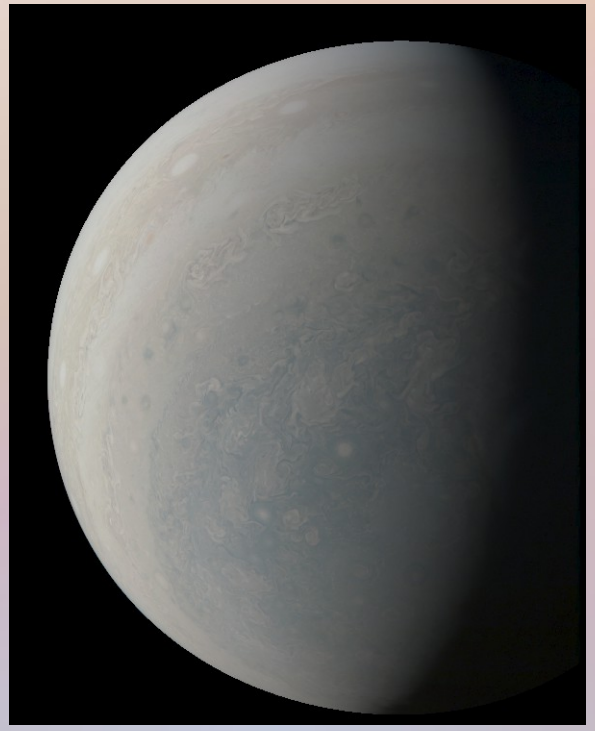


η

ι

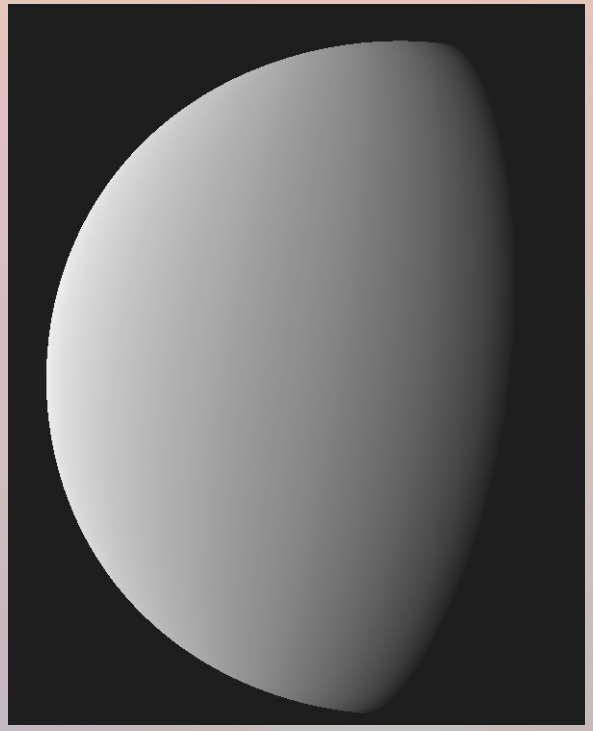
Decompanded image, solar incidence data, emission data

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



decompanded image, square-root encoded

ι



cos of solar incidence, square-root encoded

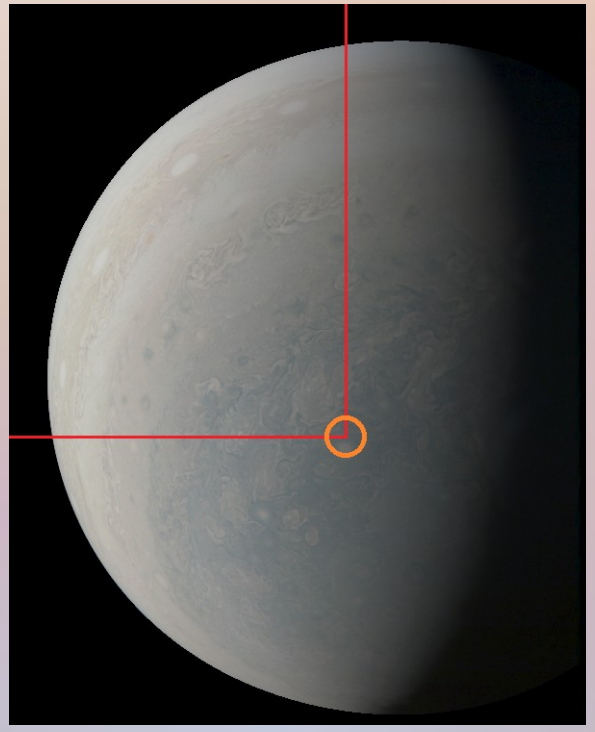
η

cos of emission angle, square-root encoded

For each pixel in decompanded image:

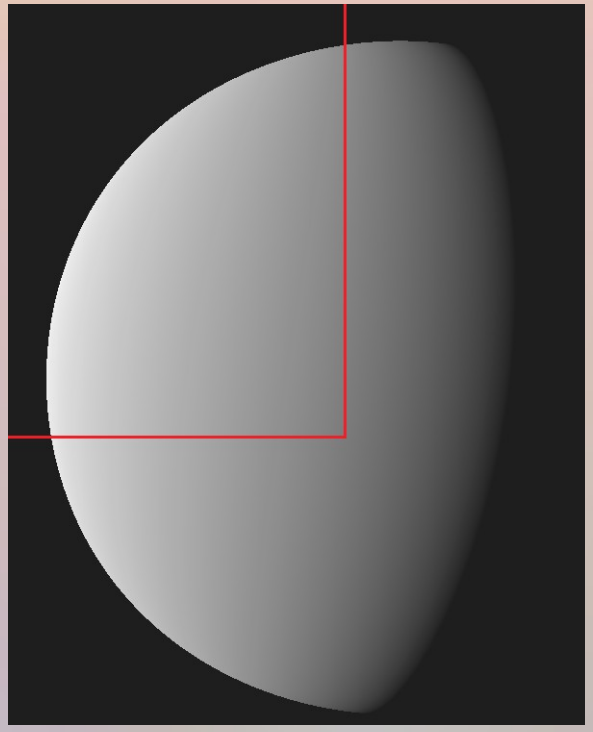
Lookup or calculate cosines of solar incidence angle and of emission angle [encoded as grey values, below]

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



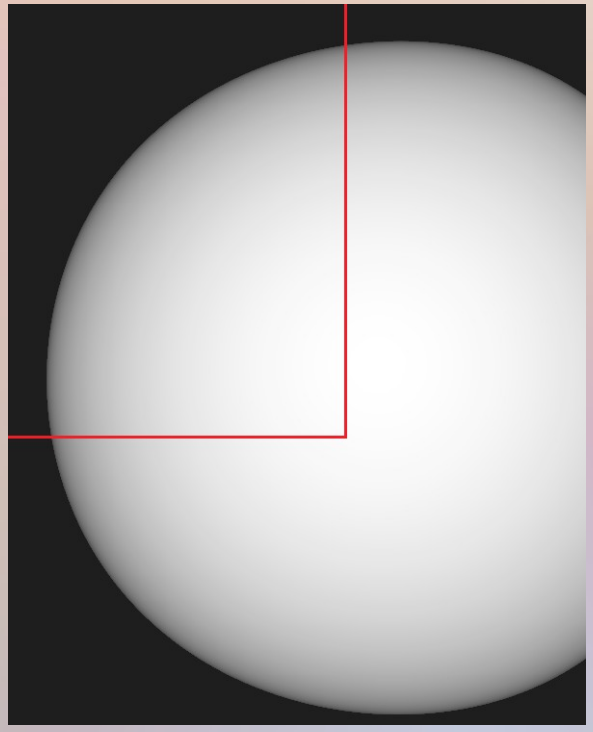
decompanded image,
square-root encoded

ι



cos of solar incidence,
square-root encoded

η

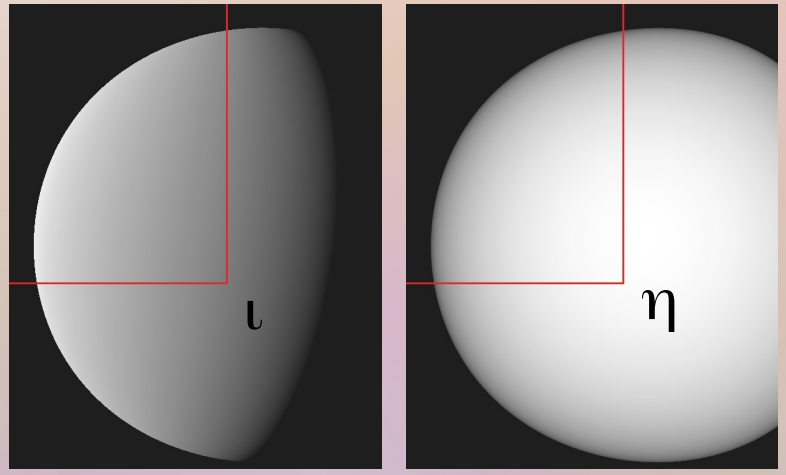


cos of emission angle,
square-root encoded

For each pixel in decompanded image:

Use cosines of solar incidence angle and of emission angle to

look-up or calculate model brightness

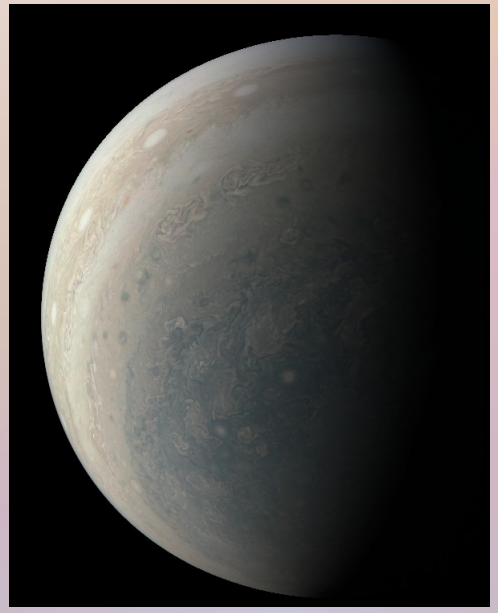


model brightness
as a function of ι and η

Result: Jupiter reference model

Divide decompanded image by Jupiter reference model

decompanded image



/

reference model

=

illumination-adjusted image

(gamma = 1, for all three images)

Decompanded colors divided by masked polynomial, stretched by gamma = 0.5

5th 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

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

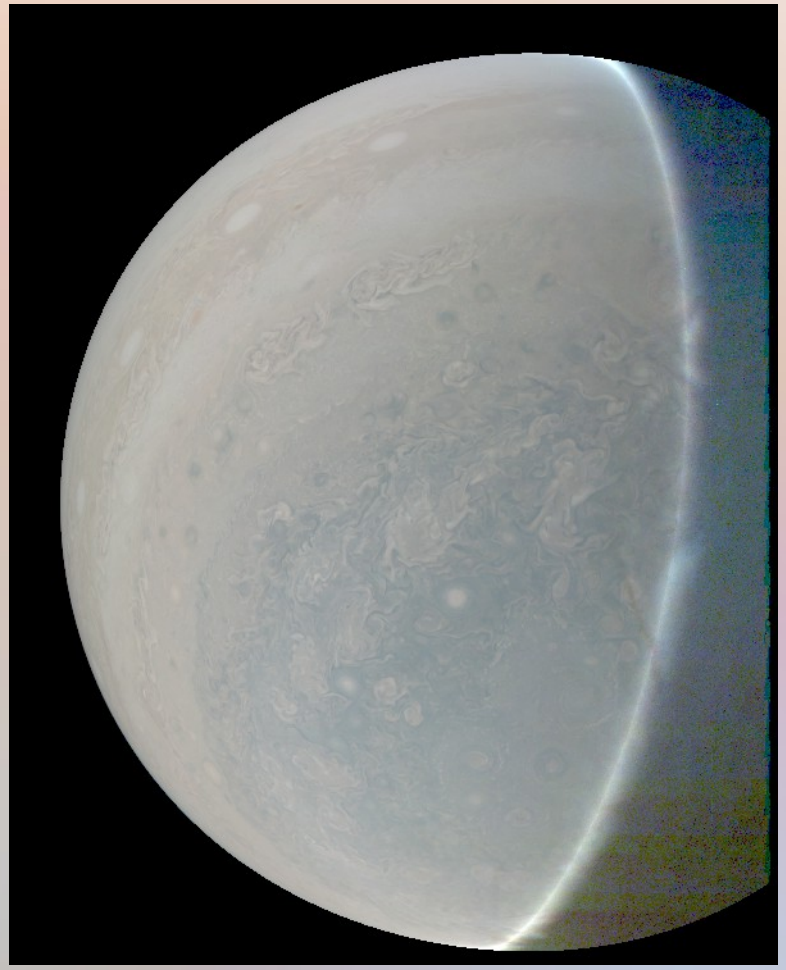
weighted **decompanded colors** divided by masked polynomial, and stretched by **gamma = 4.0**

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.

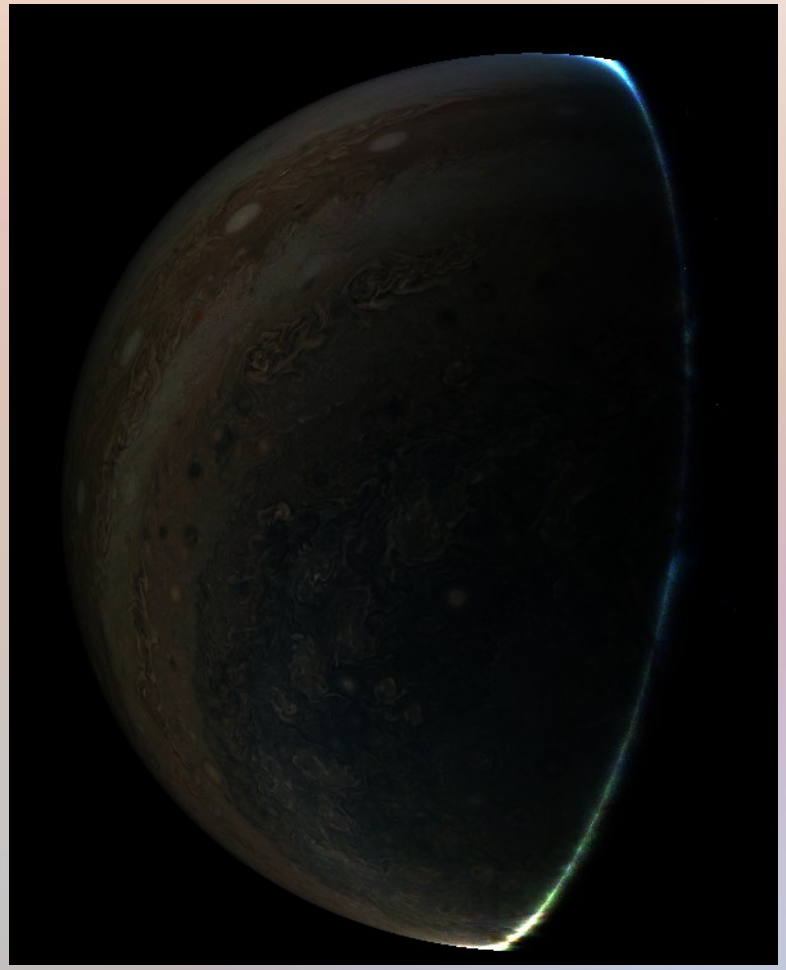


weighted **decompanded colors** divided by Lambert model, and stretched by **gamma = 0.5**

Lambert model, and gamma-stretch

$$f(\iota, \eta) := \iota$$

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 **gamma = 4.0**

Observation: Terminator too bright, zone near terminator too dark

Modified Lambert model, and gamma-stretch: Adding a constant darkenes terminator

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

weighted **decompanded colors** divided by modified Lambert model, and stretched by **gamma = 4.0**

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

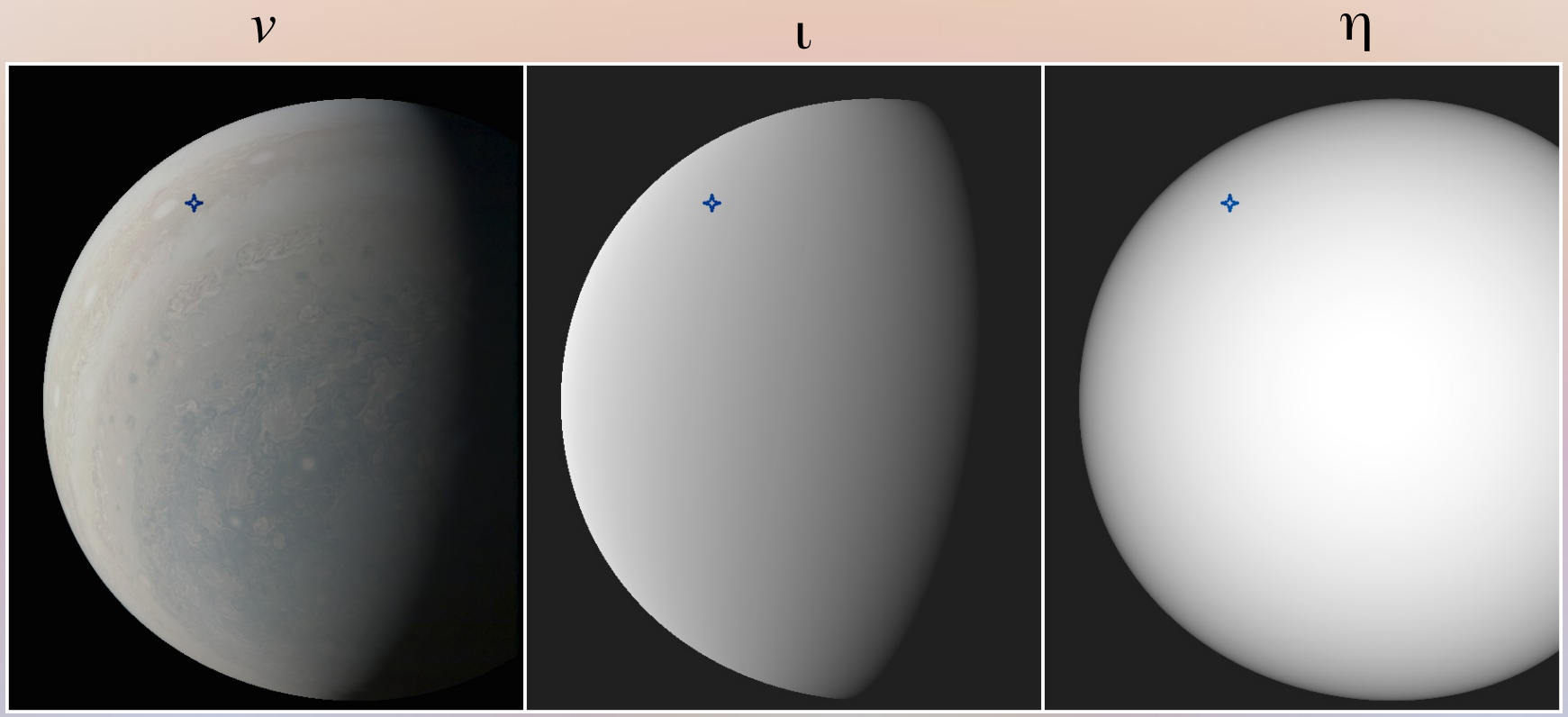
modified Lambert

desired enhancement

Conclusion: better, but more sophisticted illumination model desirable.

General approach: Monte-Carlo method

- Select random samples, each consisting of decompanded value v, ι, and η



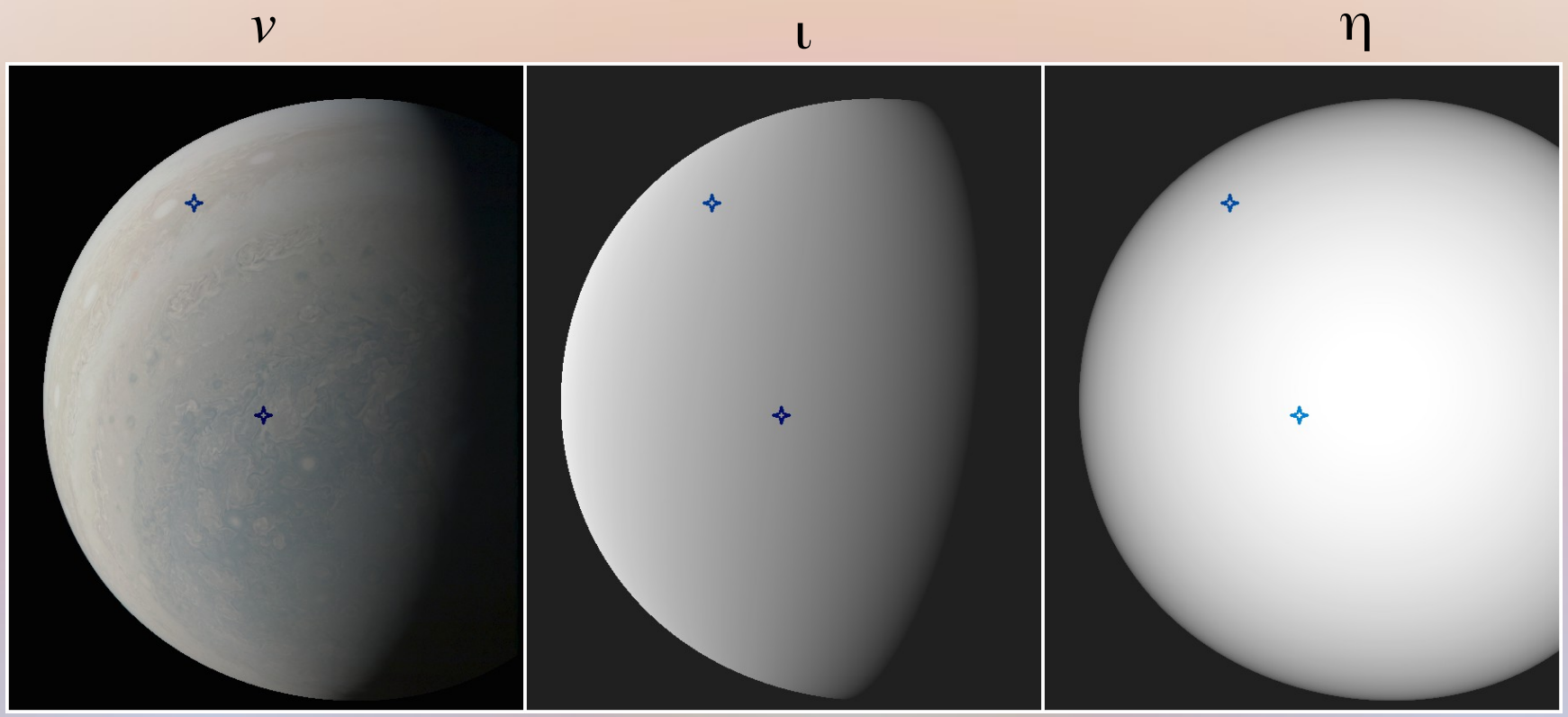
decompanded image, square-root encoded

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, ι, and η, exclude terminator and limb



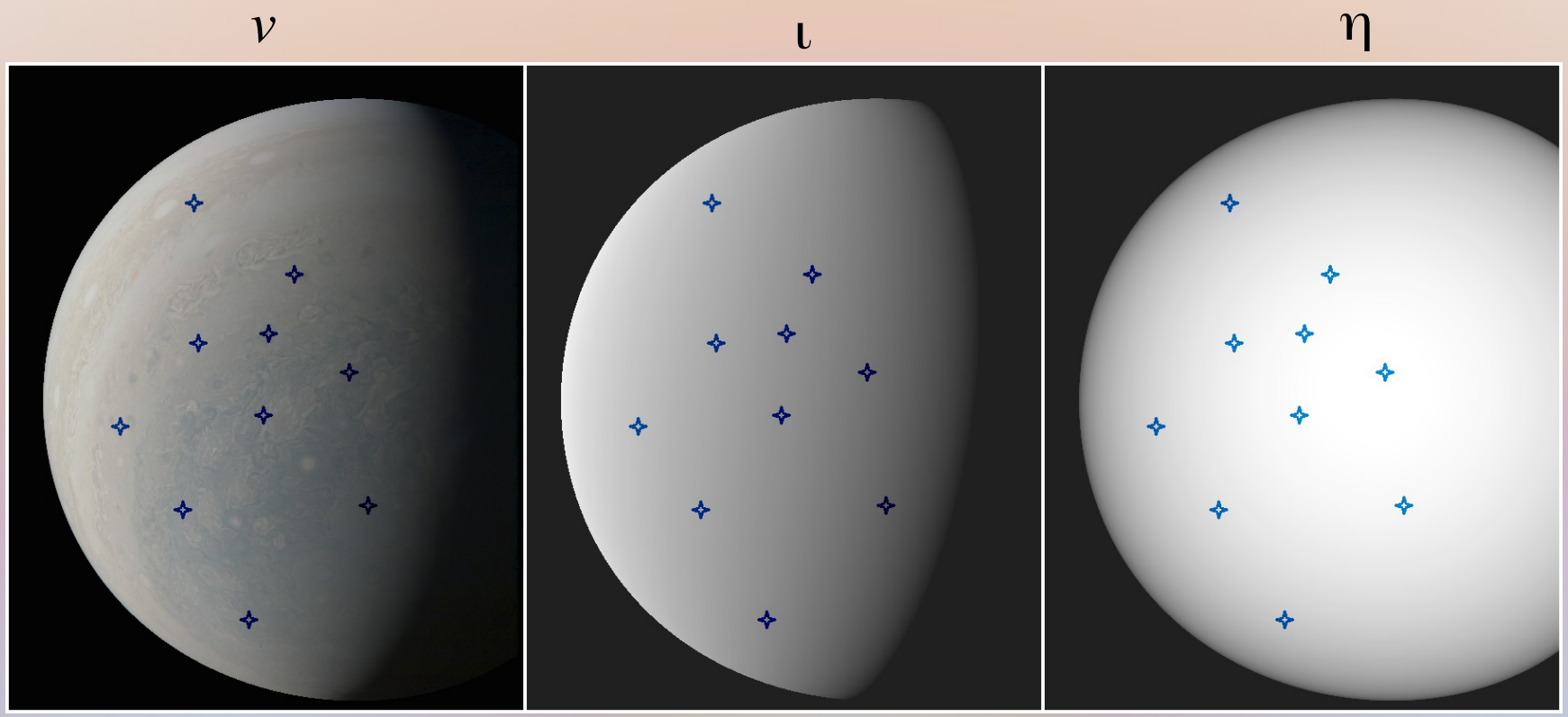
decompanded image, square-root encoded

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, ι, and η, exclude small ι, and η

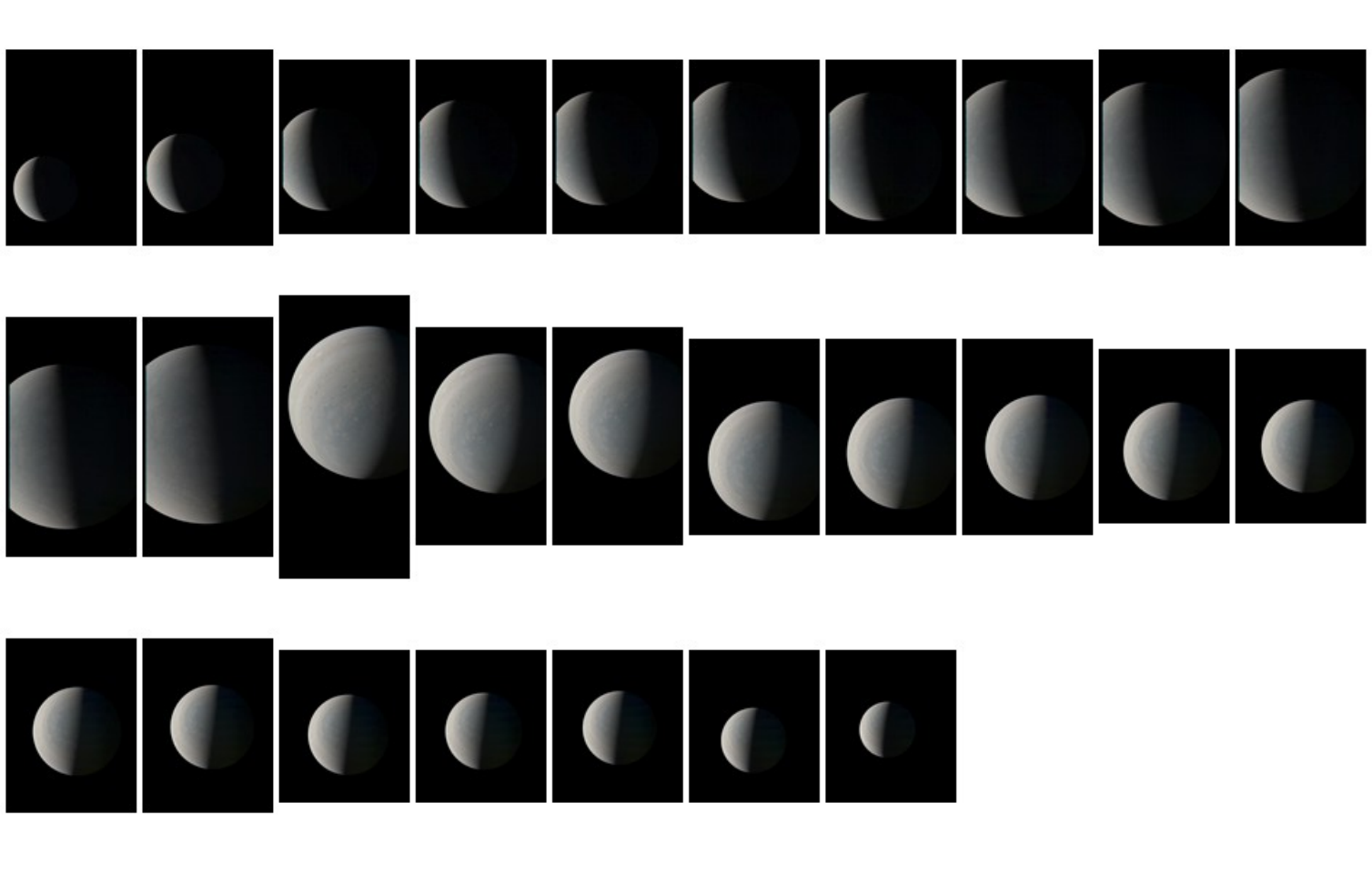


decompanded image, square-root encoded

cos of solar incidence, square-root encoded

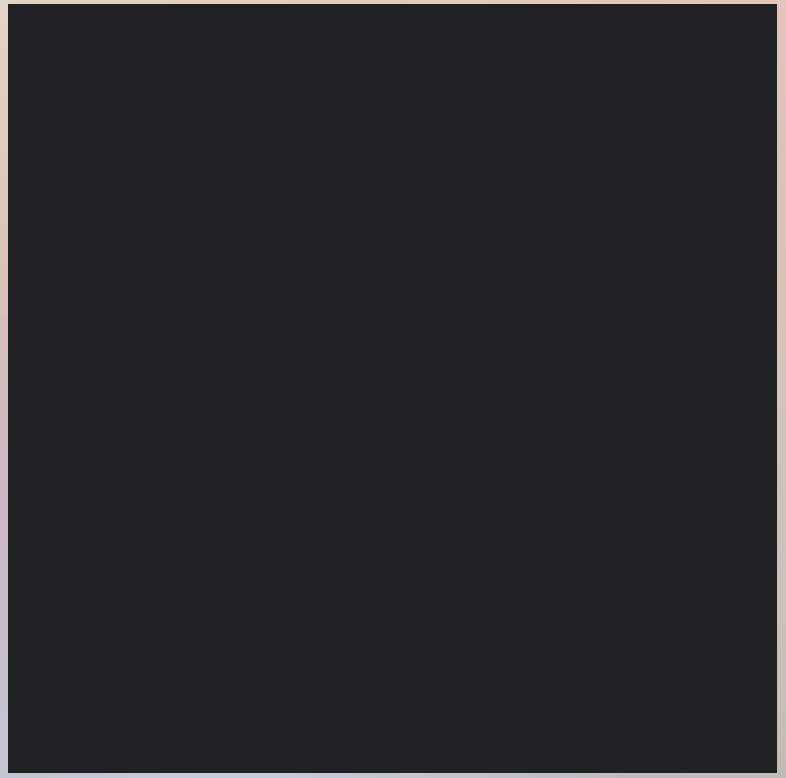
cos of emission angle, square-root encoded

1 million Monte-Carlo samples distributed over 27 Perijove-06 images (TDI 2)



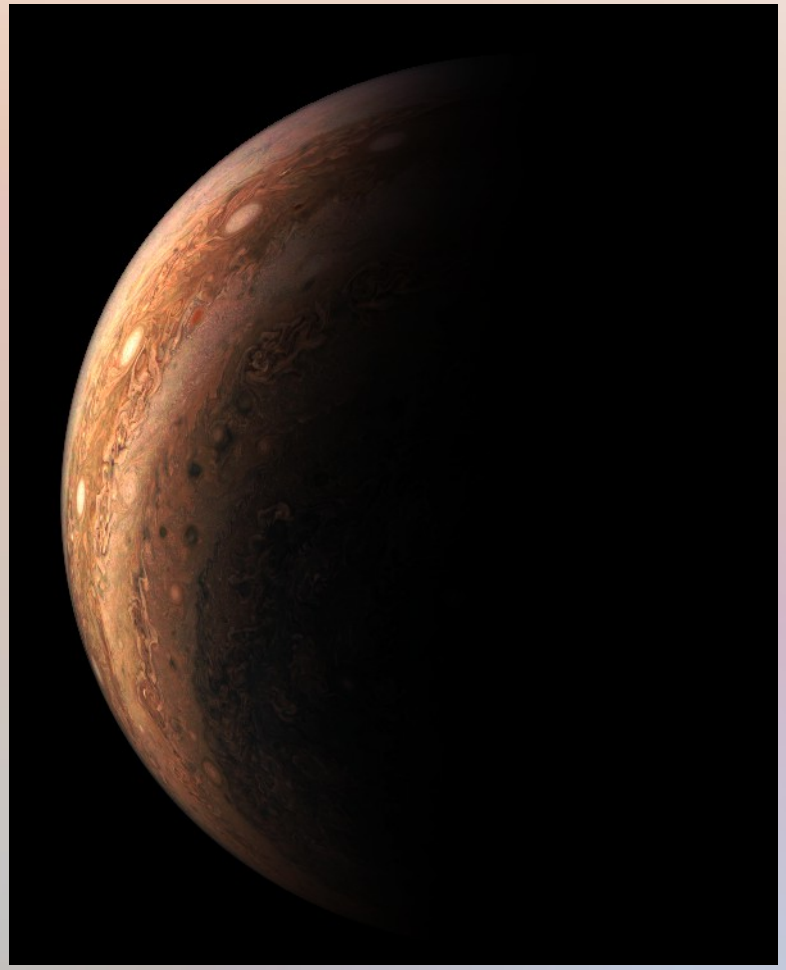
Best fit polynomial of degree 0 [constant] by least square sums of relative errors

$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot\iota_i^{\,j}\cdot\eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=0$$



cosine of emission angle

cosine of solar incidence angle

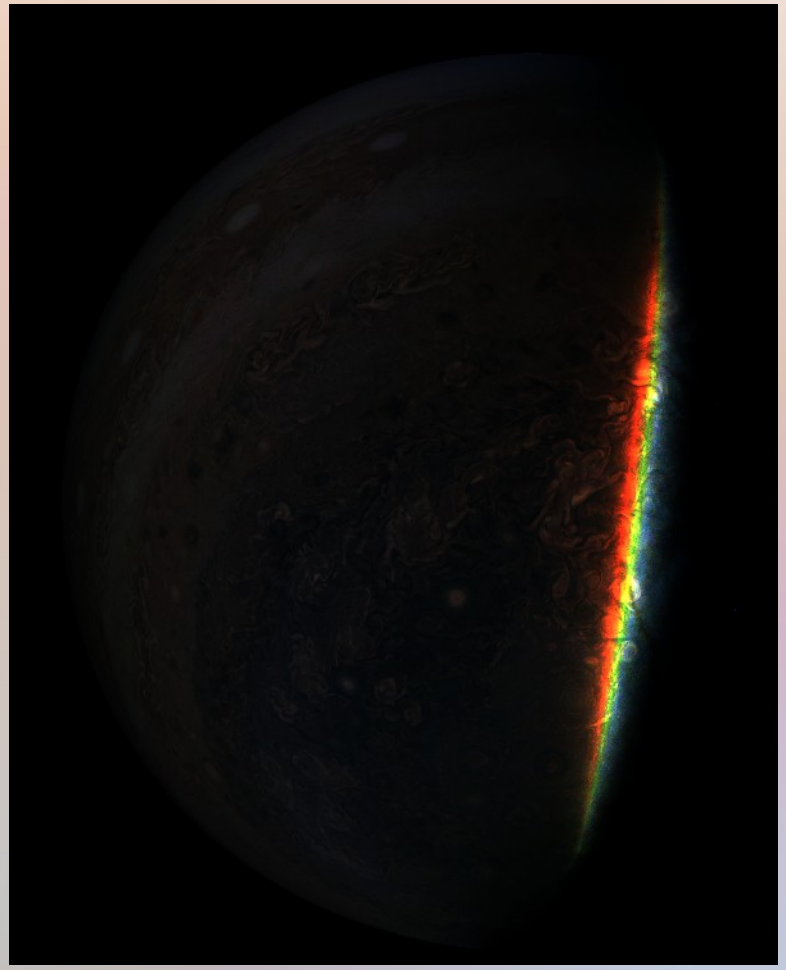


weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Note: Best-fit polynomials are calculated for each of the three color channles separately.

Best fit polynomial of degree 1 [plane] by least square sums of relative errors

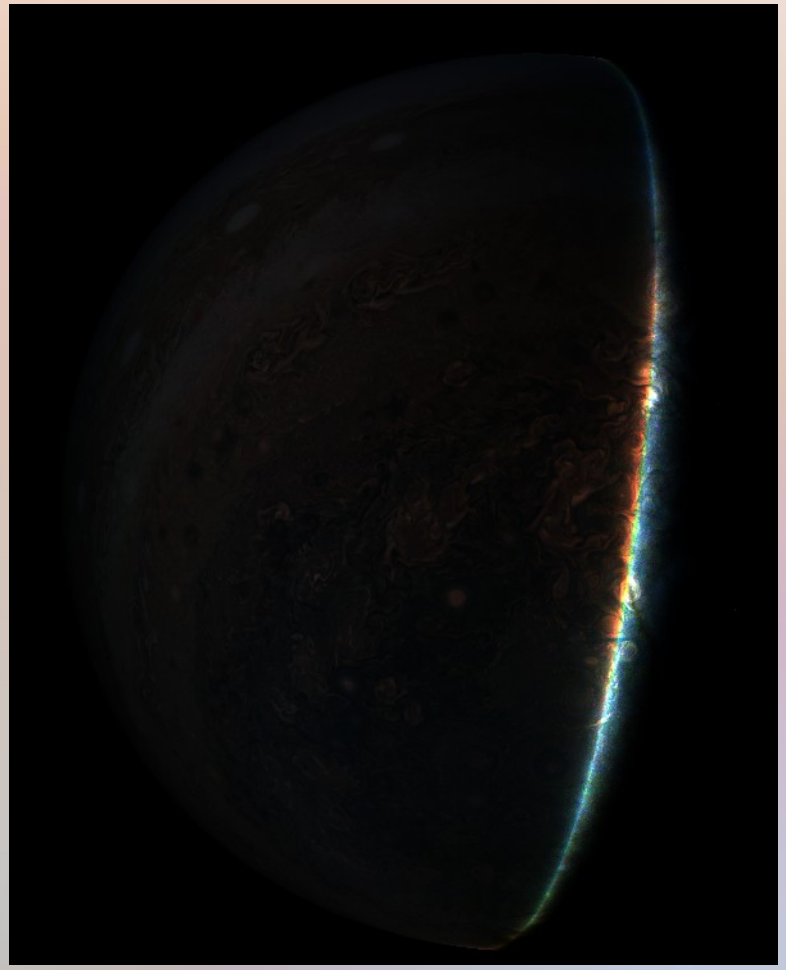
$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot\iota_i^{\,j}\cdot\eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=1$$



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

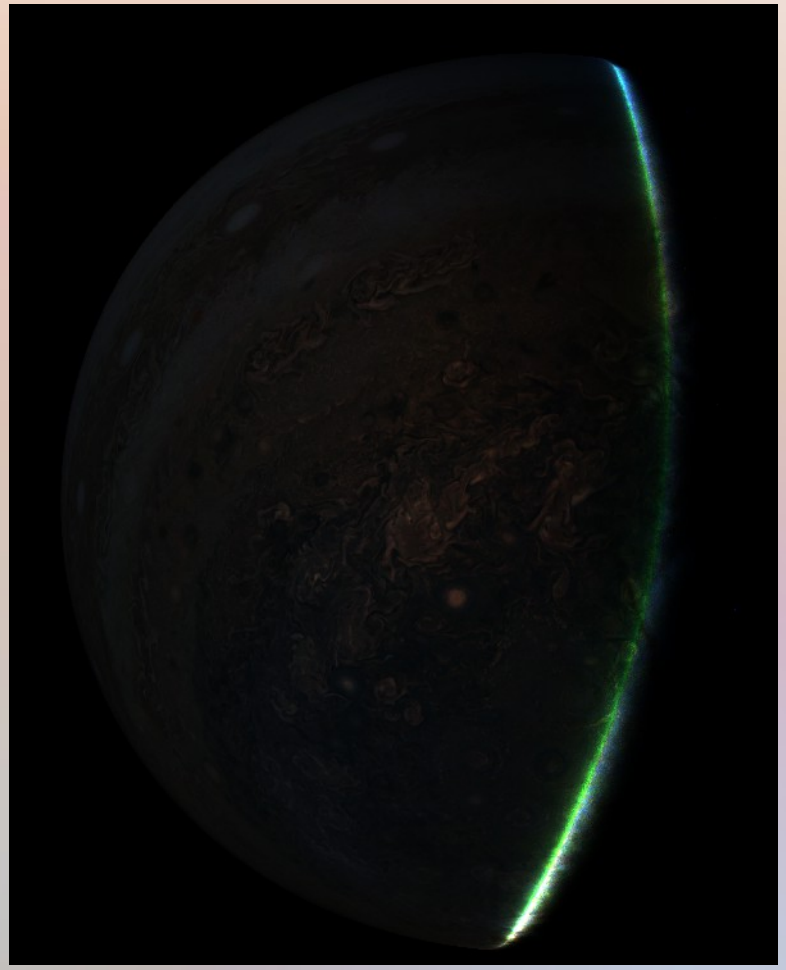
$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot\iota_i^{\,j}\cdot\eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=2$$



weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Best fit polynomial of degree 3 by least square sums of relative errors

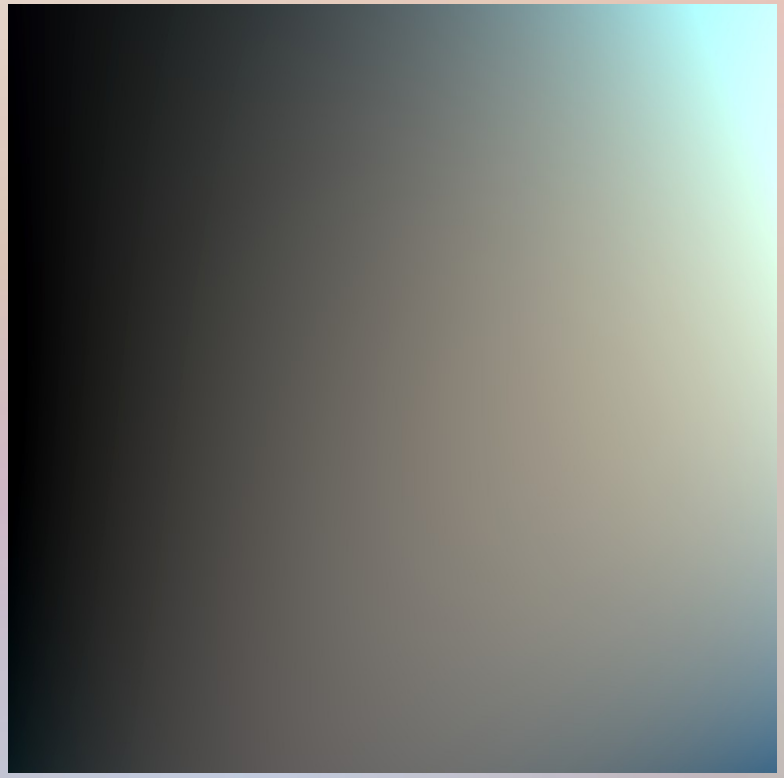
$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot\iota_i^{\,j}\cdot\eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=3$$

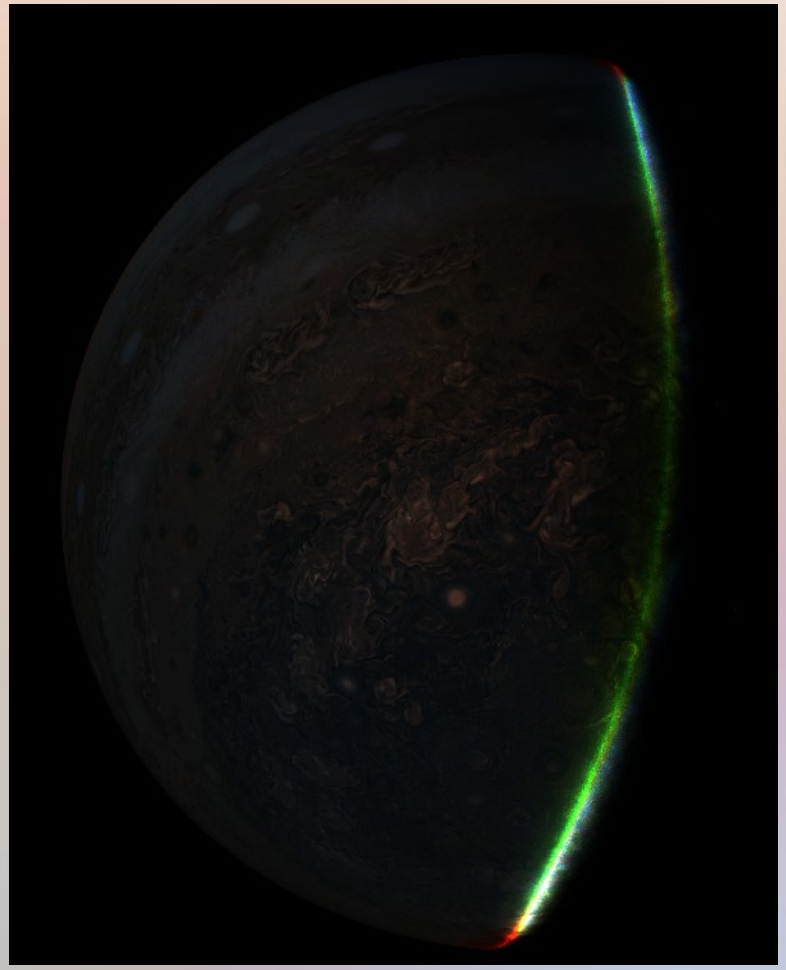


weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Best fit polynomial of degree 4 by least square sums of relative errors

$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot \iota_i^{\,j}\cdot \eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=4$$

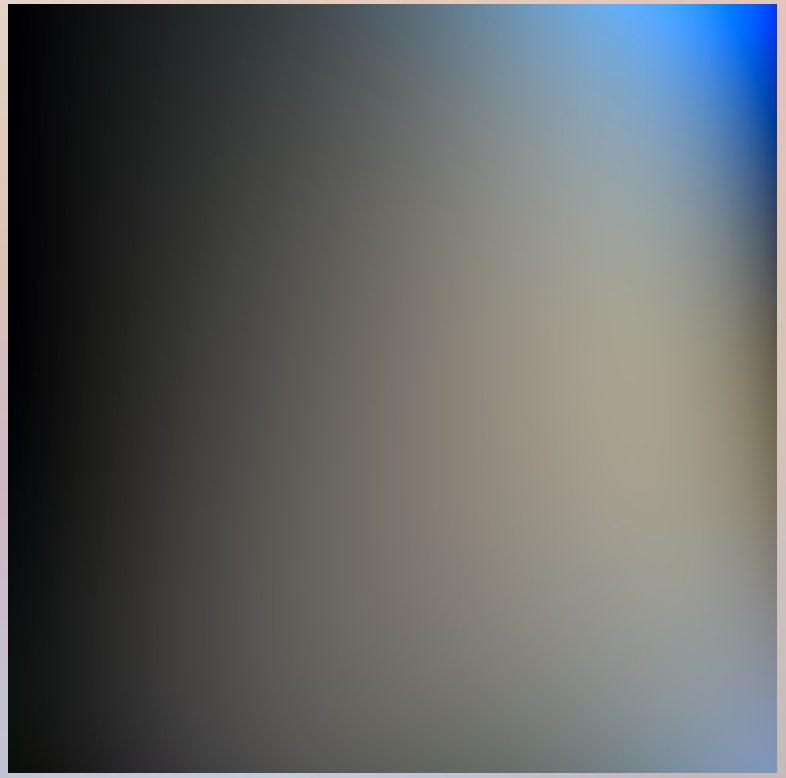


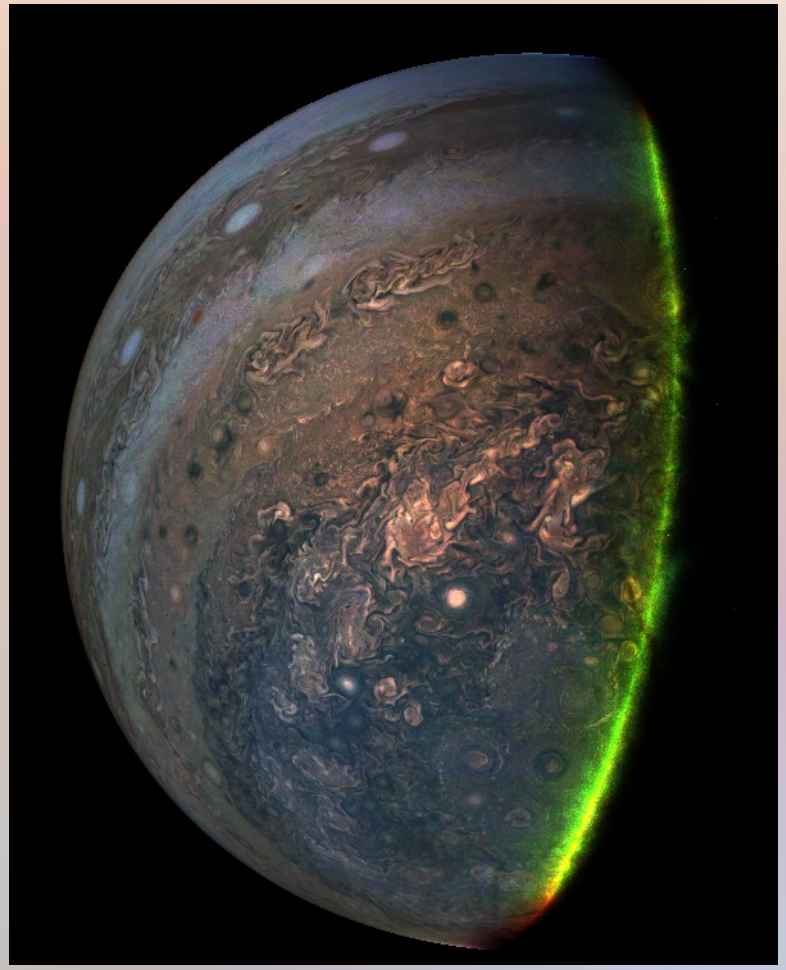


weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Best fit polynomial of degree 5 by least square sums of relative errors

$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot\iota_i^{\,j}\cdot\eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=5$$



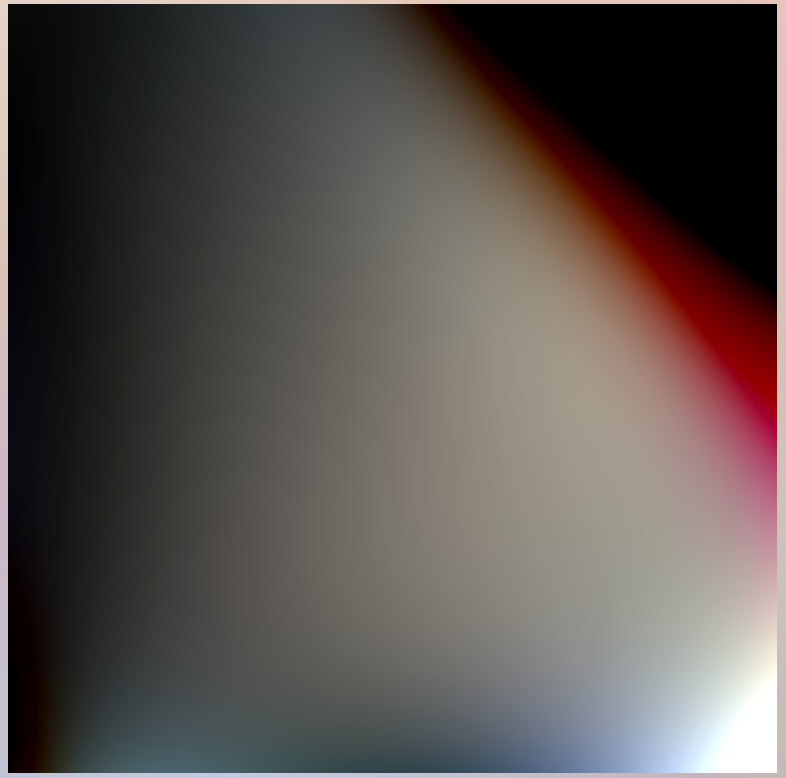


weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

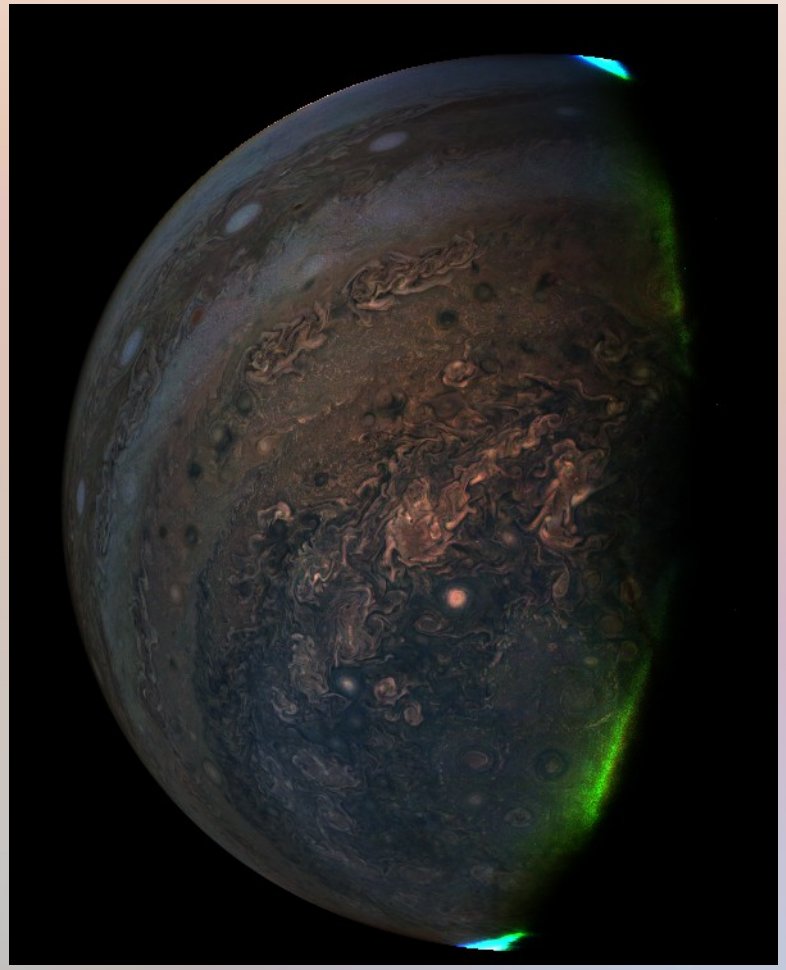
Best fit polynomial of degree 6 by least square sums of relative errors

$$\sum_{i=1}^{n}\left[\frac{\sum_{j=0}^{m}\sum_{k=0}^{m-j} a_{j,k}\cdot \iota_i^{\,j}\cdot \eta_i^{\,k}}{v_i}-1\right]^2 \overset{!}{=} min,\quad m=6$$

cosine of emission angle



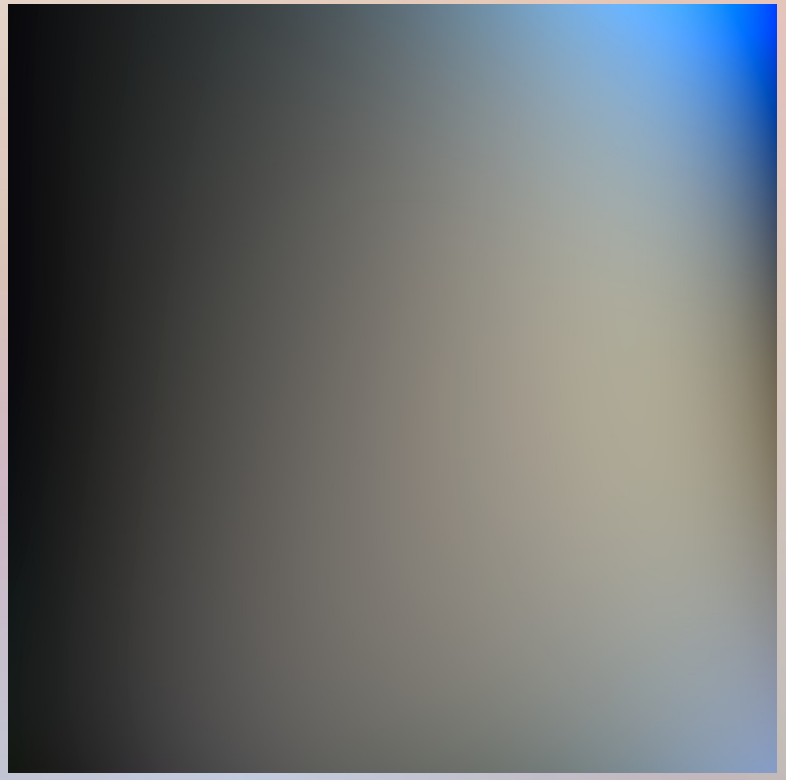
cosine of solar incidence angle



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



weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Use 5-th degree best fit polynomial of green channel for all three color channels

reduced BRDF

gamma-stretched texture

weighted **decompanded colors** divided by model, and stretched by **gamma = 4.0**

Initial goal accomplished.

A Jupiter model: Reduced BRDF + Texture

decompanded image

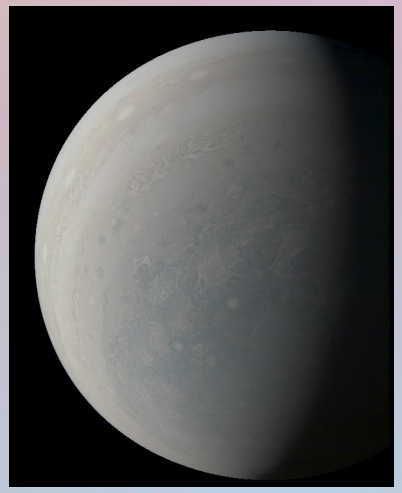
/

reduced BRDF

=

gamma-stretched texture

can be written as



=

*

Does this Jupiter model remove all illumination-induced effects?

Compare two images reprojected to the same perspective

PJ6 #124 decompanded

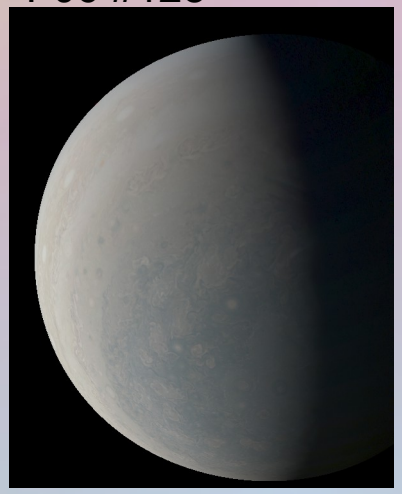
/

reduced BRDF

=

gamma-stretched texture

PJ6 #128

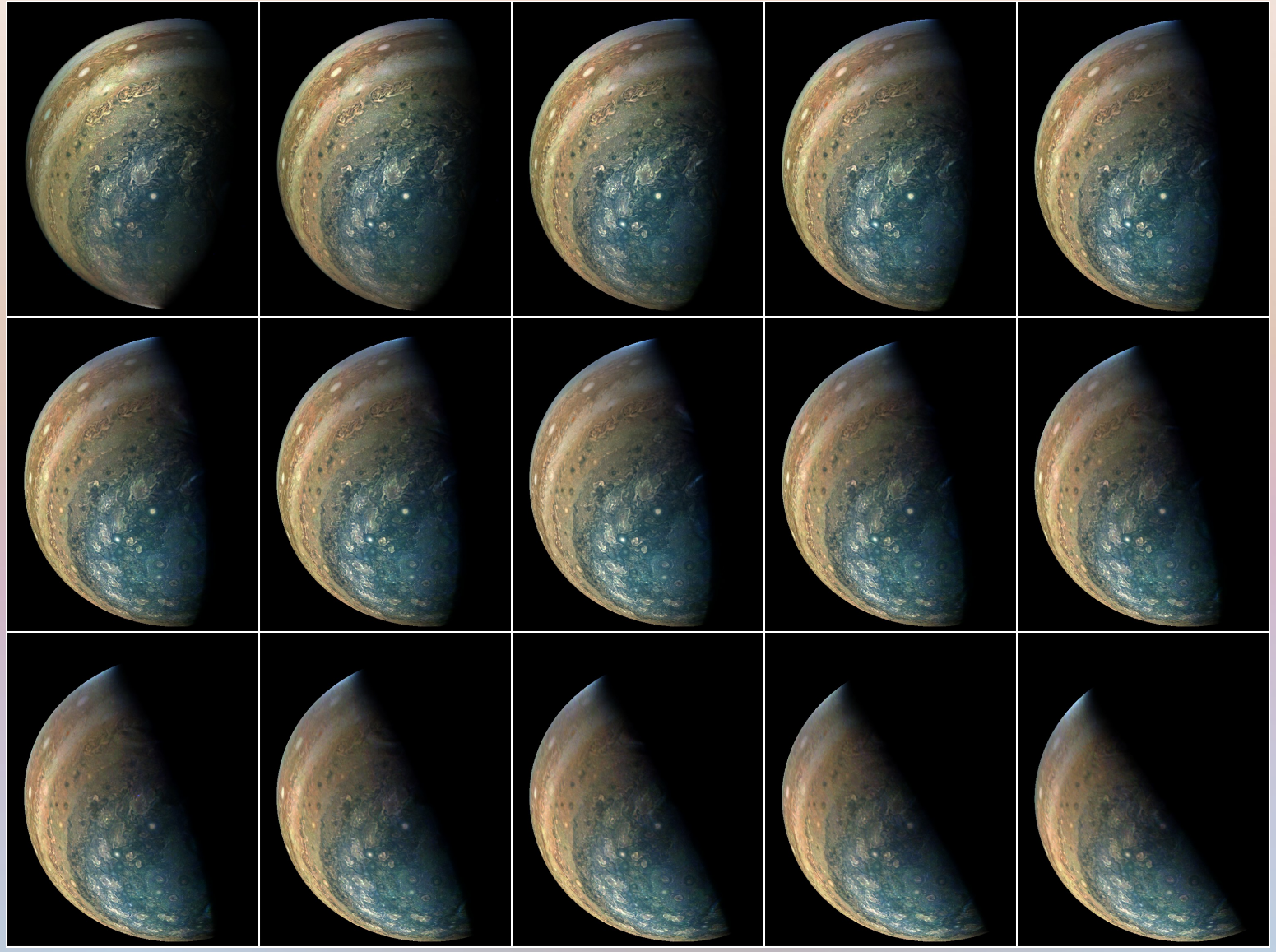


/

=

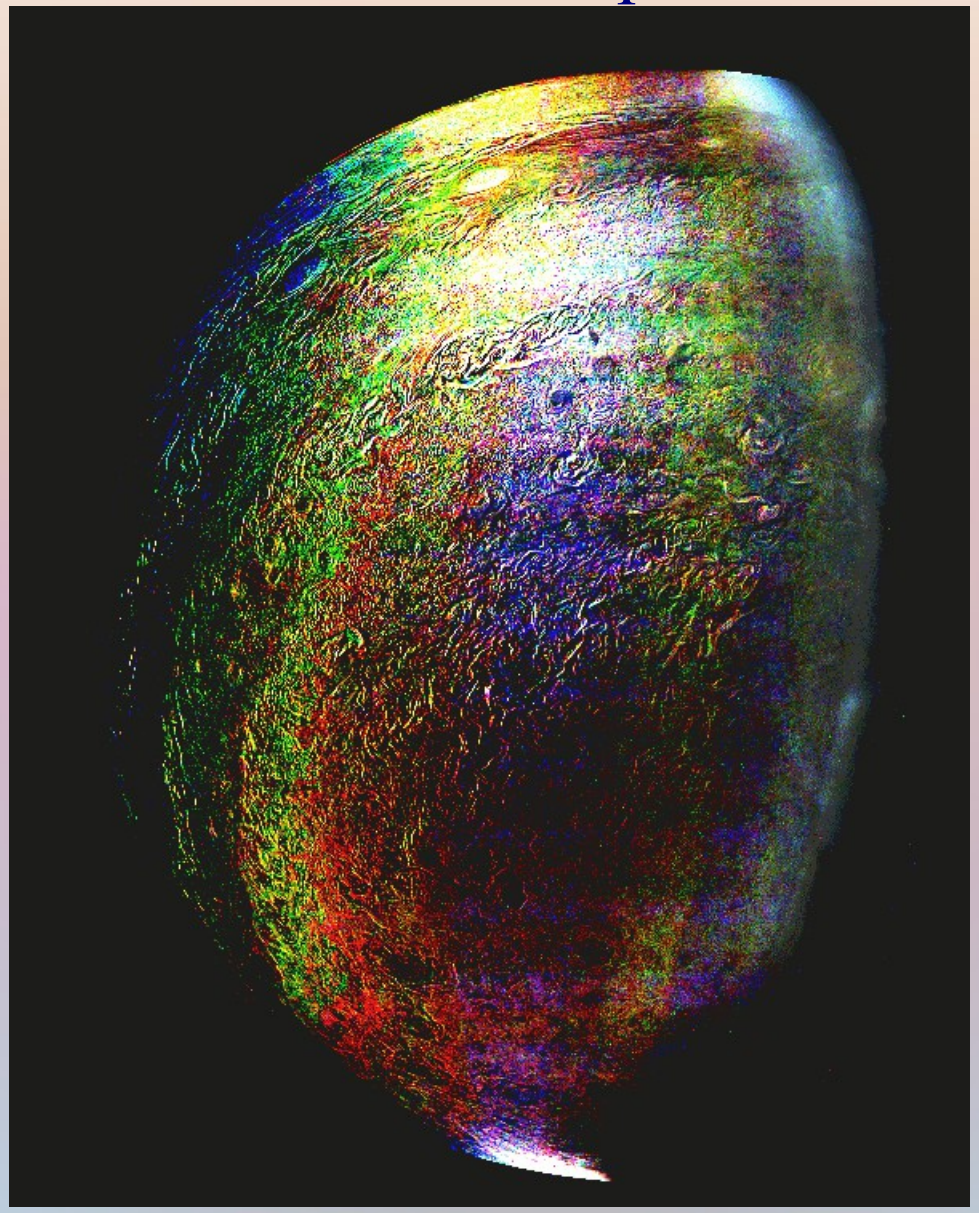
Observation: Textures differ by subtle haze features near the terminator

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();
```