

# **plate\_renderer with SPICE DSK subsystem**

Naru Hirata  
(ARC-Space, the University of Aizu)  
v20201012

# What SPICE can do?

## plate\_renderer



REAL image  
taken by  
Hayabusa



Time, Position,  
Attitude, Target,  
Shape, Lighting  
condition,  
Instrument

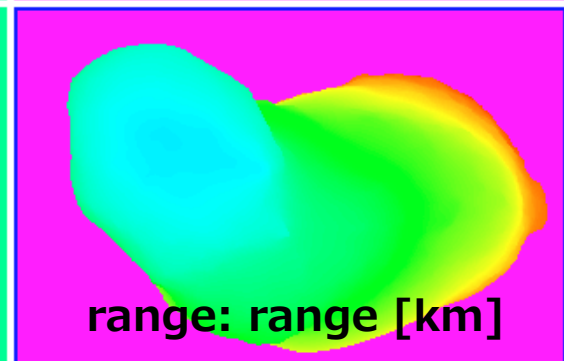
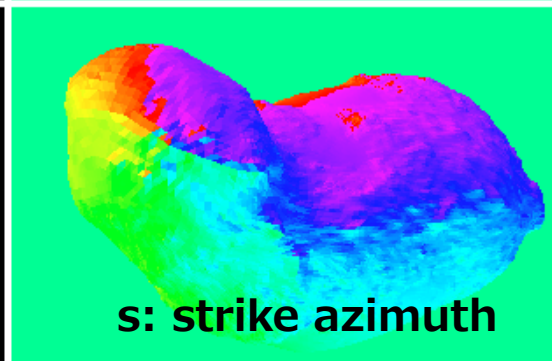
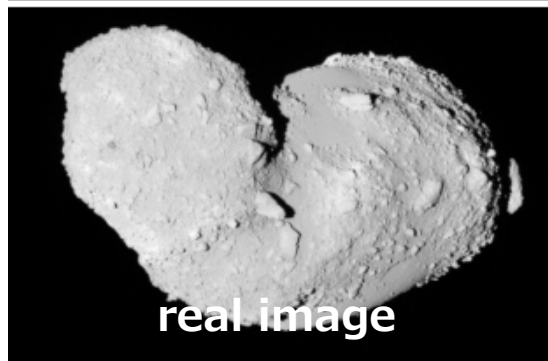
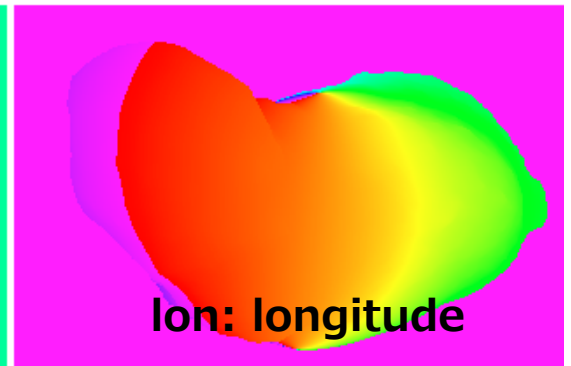
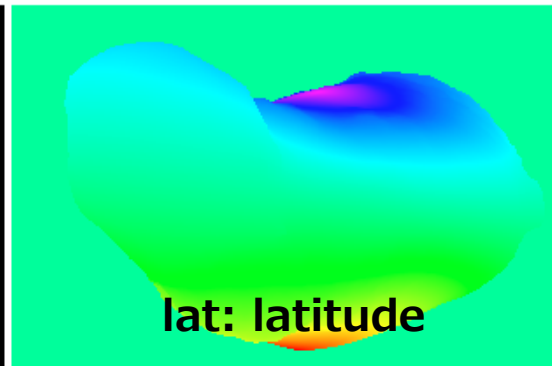
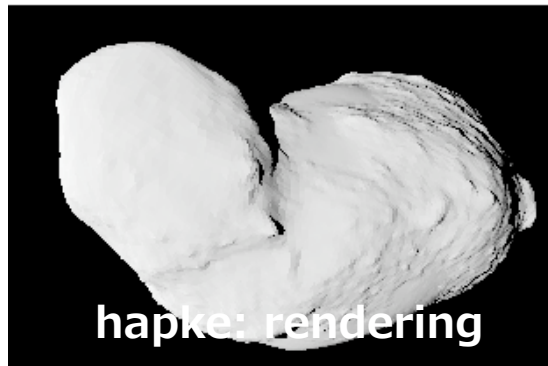
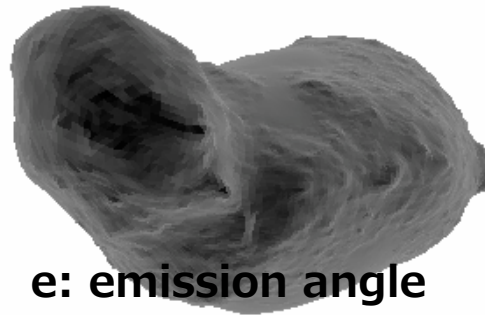


Reproduced  
image with  
SPICE

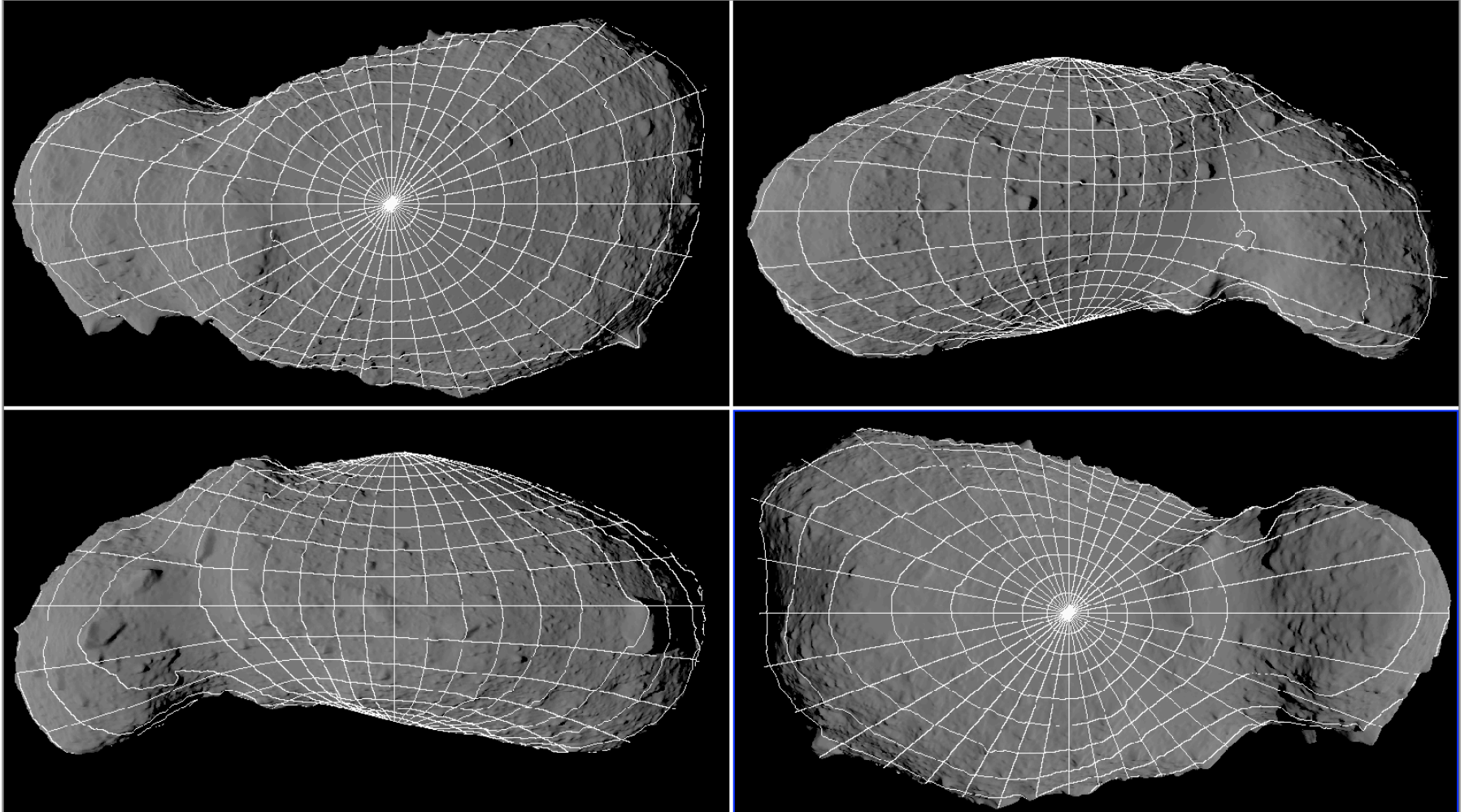
# plate\_renderer

- Tool for simulating camera observations on small bodies
  - Implementing a light scattering model of asteroids
  - Byproducts: geometric information at a given epoch or a given condition

# Outputs of plate\_renderer



# 4-sided view of Itokawa: with a free viewpoint mode



# How to compile plate\_renderer

- Modify Makefile to fit your local environment
  - Check and modify (if necessary) path in INCLUDES, LIBS
  - If you want to make a Hayabusa2 version, deactivate CFLAGS for Hayabusa1 and activate CFLAGS for Hayabusa2
- Type make command to produce plate\_renderer executable file
  - copy it to your working directory

# How to use

```
$ ./plate_rendere DSK_kernel kernels ...
```

- 1<sup>st</sup> argument should be DSK\_kernel file
- Other kernels are listed after dsk

- Options

- Selection of output files
- Draw the lat/lon grid
- Change the viewpoint
  - Simulate an actual situation of S/C (reading kernels): S/C VP mode
  - View from an arbitrary location: Free VP mode

# Example / 使用例

```
% ./plate_renderer kernels/Itokawa/dsk/hay_a_amica_5_itokawashape_v1_0_512q.bds $(cat
kernel_list_rendezvous.txt)
kernels/Itokawa/dsk/hay_a_amica_5_itokawashape_v1_0_512q.bds is loaded.
(snip)
Use HAYABUSA_AMICA
Input epoch (UTC): 2005-10-01T00:00:00 <- Type it
2005-10-01T000000
S/C position (Body-Fixed):
  X =      7.63712
  Y =     -1.97583
  Z =      0.05974
Solar position (Body-Fixed):
  X = 161050864.48813
  Y = -14520949.19513
  Z =  4732054.85776
The central LOS vector (body-fixed):
  X =     -0.97037
  Y =      0.24123
  Z =     -0.01401
line: 1000
```



# Command line options (cont.)

-v SC\_lat/SC\_lon/solar\_lat/solar\_lon

Free viewing point mode with provided sub-S/C lat/lon and sub-solar lat/lon

自由視点モードで視点の緯度経度，光源の緯度経度を指定

-erlnp

Output geometry info [e], range info [r], illumination conditions [l], lat/lon info [n], and polygon ID [p]

幾何条件データ[e]，距離[r]，日照条件[l]，緯度経度[n]，ポリゴンID[p]を出力

-All

Output all informations (same as -erlnp)

全ての情報出力（-erlnpを指定したのと同じ）

# How to work

- Loading kernels
- Selecting view mode
  - S/C VP mode: Reading geometries from kernels at the given epoch
  - Free VP mode: Computing geometries from the given conditions
- For all pixels
  - Check a footprint of the line-of-sight vector of a certain pixel on the shape model
  - Compute a lighting condition, If a footprint exists
    - check whether the footprint is shaded or not
  - Estimate a brightness at the footprint with Hapke's light scattering model
  - Output results as FITS format files

# User defined functions

- `int ONC_getlos_c( )`
  - Compute a line-of-sight vector of a given pixel
- `double hapke( )`
  - Compute brightness with Hapke's light scattering model
- `void get_plbore ( )`
  - Check a footprint of the line-of-sight vector of a certain pixel on the shape model
- `int get_cam_info ( )`
  - Read camera info. from ik

# Contact

- Please send your questions and comments to Naru Hirata ([naru@u-aizu.ac.jp](mailto:naru@u-aizu.ac.jp)) or our group mail address ([arcspace@u-aizu.ac.jp](mailto:arcspace@u-aizu.ac.jp)).