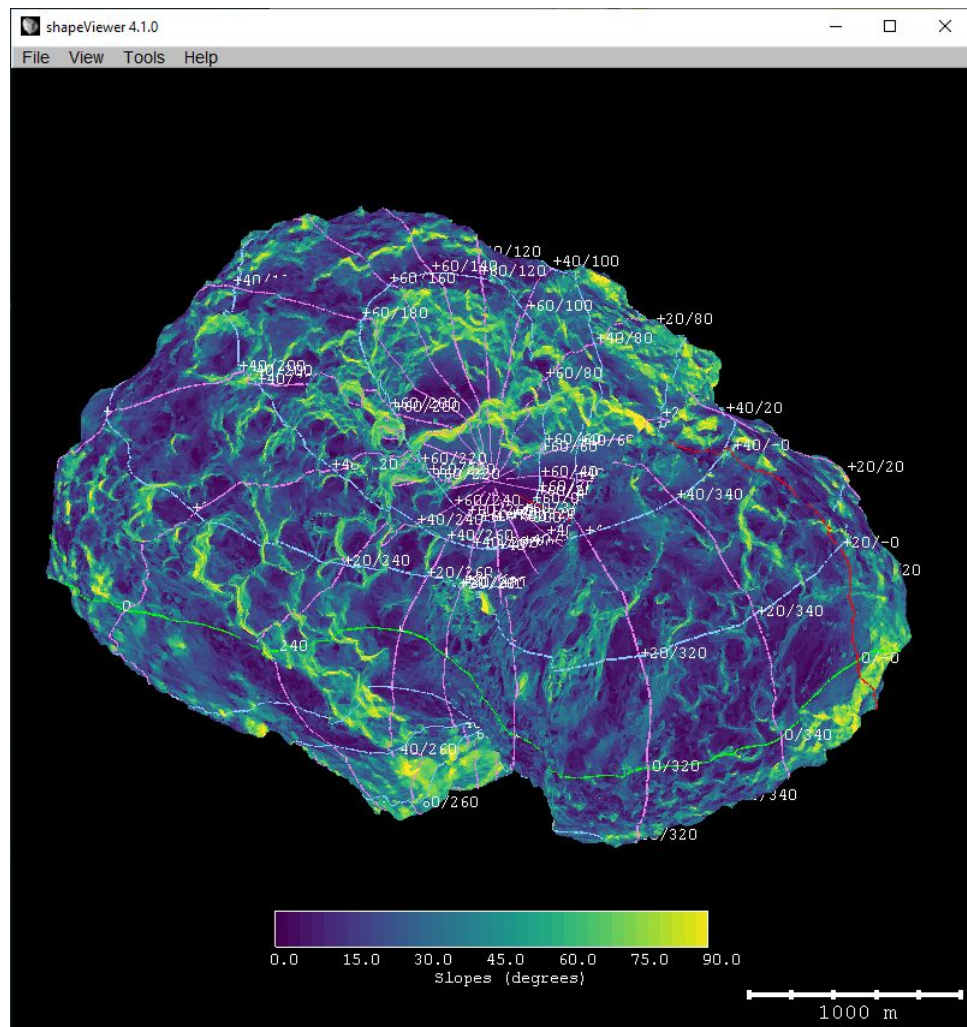


shapeViewer User Manual

version 4.1.1



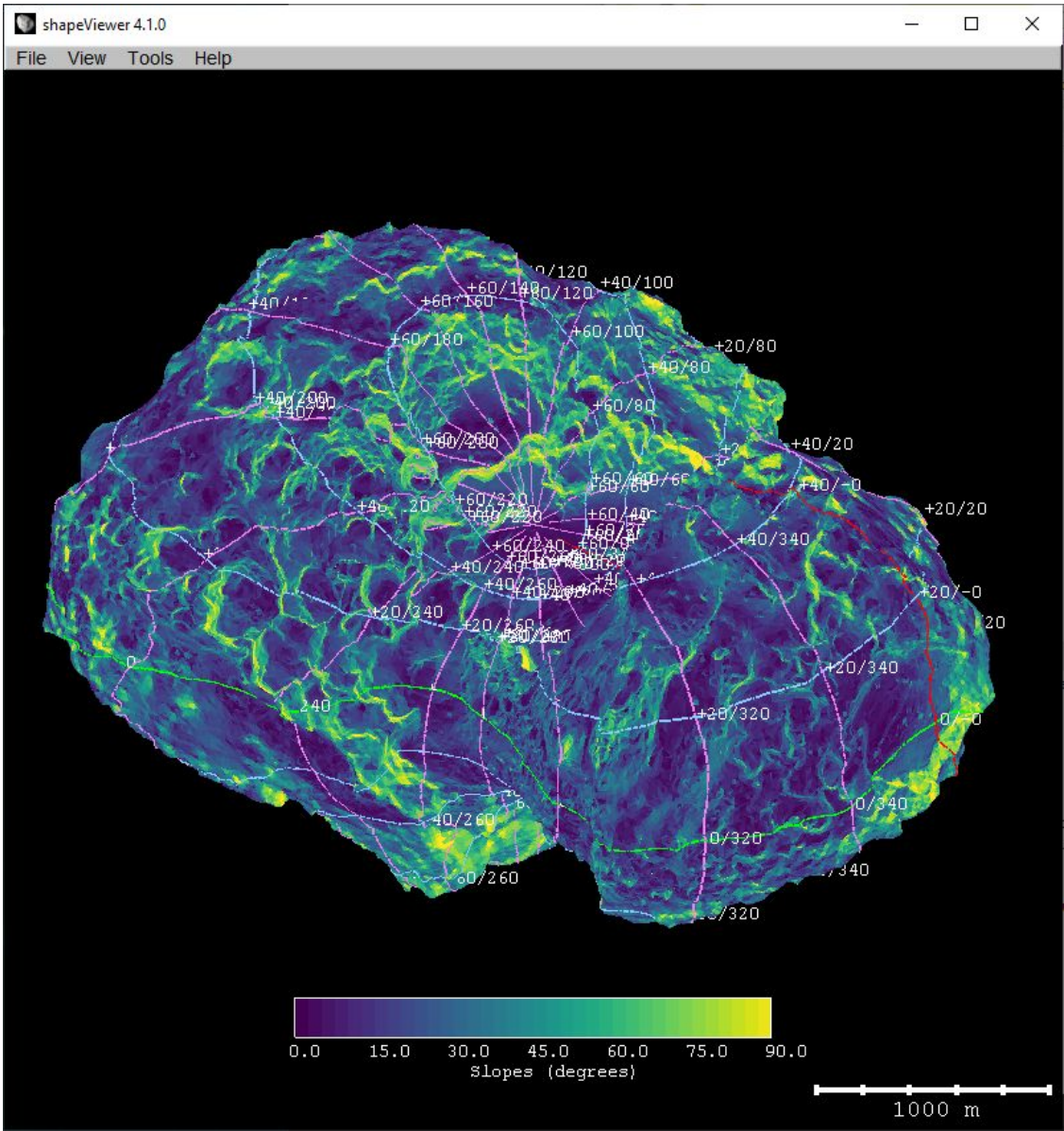
Jean-Baptiste Vincent

comet-toolbox.com

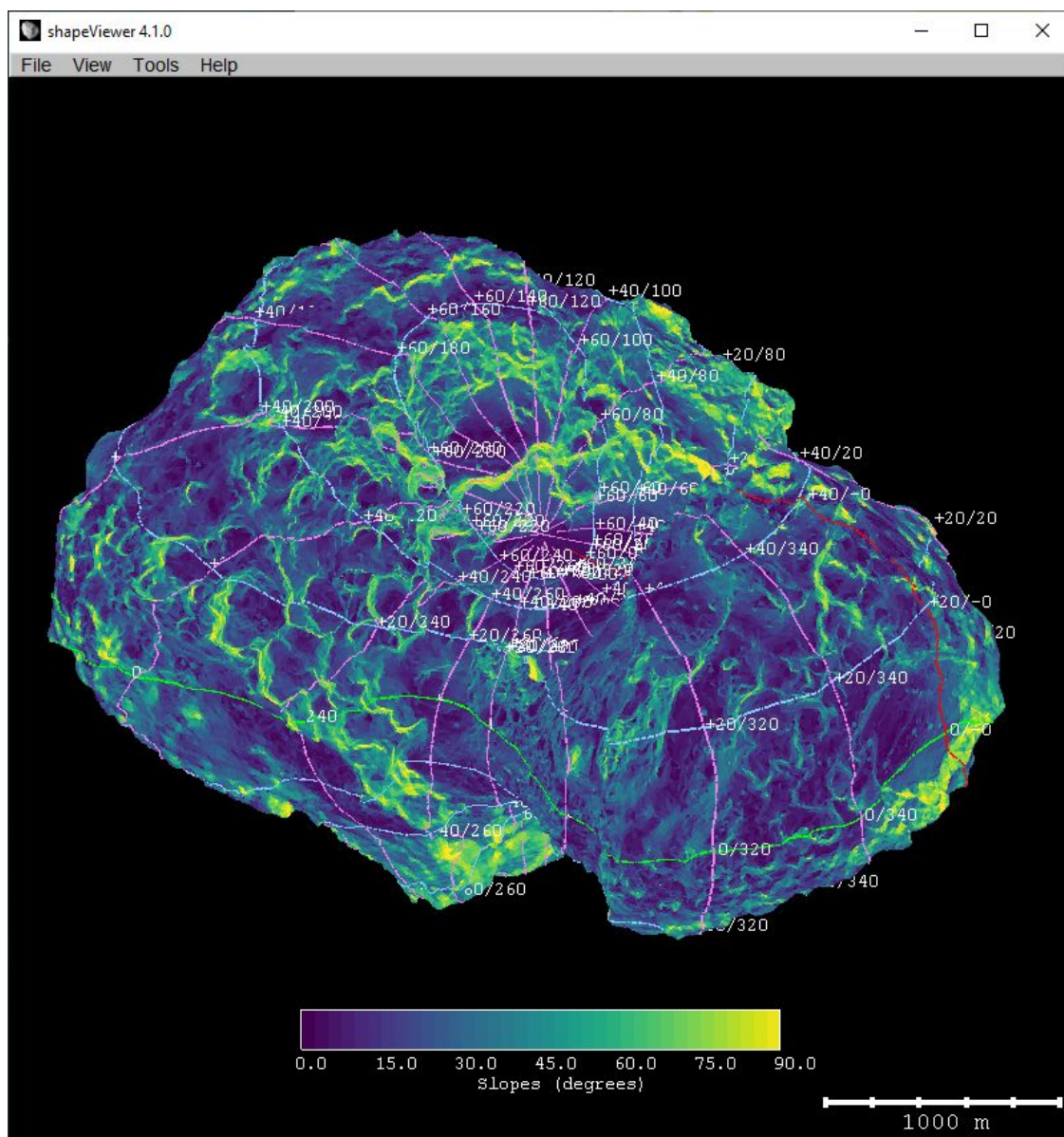
January 9, 2024

USER MANUAL

1	About	3
1.1	Main features	4
1.2	Download and Installation	4
1.3	Acknowledgement	5
1.4	License	5
2	Graphical Interface	7
2.1	Main window	7
2.2	File menu	8
2.3	View menu	14
2.4	Tools menu	19
2.5	Help menu	20
3	Integrated Console	21
4	Publications	25
5	CHANGELOG	27
6	VERSION 4 “Multiple bodies”	29
6.1	BRANCH 4.1 “JSON configuration”	29
6.2	BRANCH 4.0 “Multiple bodies”	30
7	VERSION 3 “Multiple missions support”	33
7.1	BRANCH 3.0 “Mission packages”	33
8	VERSION 2 “GUI”	35
8.1	BRANCH 2.9 “Embedded console and usability improvements”	35
8.2	BRANCH 2.8 “Image projection/texturing”	36
8.3	BRANCH 2.7 “Public release”	37
8.4	BRANCH 2.6 “CLI interface”	37
8.5	BRANCH 2.5 “Sequence mode, and usability improvements”	38
8.6	BRANCH 2.4 “COSSIM”	40
8.7	BRANCH 2.3 “Movie Mode”	40
8.8	BRANCH 2.2 “Photometric models”	40
8.9	BRANCH 2.1 “SPICE”	41
8.10	BRANCH 2.0 “GUI”	41
9	VERSION 1	43



ABOUT



shapeViewer is a scientific software designed to display a 3D model of a celestial object in the context of a space mission, simulate observations and operations scenarii, and project data on the shape.

It provides several tools specifically developed for mapping and geomorphological investigation. It is particularly suited for the study of small bodies (asteroids, comets, ...) as it can handle display and mapping of highly convex objects.

shapeViewer has been used for operations planning and scientific analysis by the Rosetta (ESA) and Dawn (NASA) missions since 2010. It is currently one of the planning tools for the Asteroid Framing Cameras on board Hera (ESA) and will be adapted to Comet Interceptor (ESA) in the near future.

As of 2022, the software has been referenced in [22 peer-reviewed articles](#) and many conference contributions.

shapeViewer is available for free at comet-toolbox.com/shapeViewer.

1.1 Main features

- interactive visualization of asteroid and comets shape models, with photometrically realistic rendering,
- high performance: displays shape models with 1+ million polygons, runs at 60 FPS on a standard laptop,
- real time rendering of operational scenarios, spacecraft orbit and attitude pre-configured, using the standard SPICE kernels for each mission,
- retrieval of observation geometry and solar angles (incidence, emission, phase, elevation),
- calculation and display of the gravity field and effective slopes on the surface,
- calculation of topographic roughness
- accurate projection of images and other datasets on the shape, individually or in a user-defined sequence (on the fly, no need for any preprocessing of the data),
- view and export cylindrical, orthographic maps,
- ... and many other features

The program consists of a small executable core engine (<4Mb) and pre-configured mission packages that contain all necessary information. Mission packages are available for many projects and new ones can be easily added by the users.

We try to streamline the user experience as much as possible. The most commonly used tools are accessible from the graphical interface and should be sufficient for the majority of users. The program can, however, do much more than that and provides an integrated console to access to advanced functions.

Note: *shapeViewer* is a fully offline software. Apart from checking for updates (manually activated), the software does not interact with the world outside your computer.

1.2 Download and Installation

shapeViewer can be found at comet-toolbox.com/shapeViewer.

The program is distributed as a ZIP file archive. Simply decompress it to the location of your choice, no installation required. The archive contains two executables:

- *shapeViewer.exe* is the Windows 64bits version. It is developed and tested on Windows 10, but should work on Windows 7 as well. Linux and macOS users can run this version via the [Wine](#) interface.
- *shapeViewer_linux* is a native Linux 64bits version compiled on Debian 11. It should run on all distributions based on the same set of kernels/libraries (e.g. Ubuntu).

1.3 Acknowledgement

We kindly ask all users to include the following acknowledgement: *This research has made use of the scientific software shapeViewer (www.comet-toolbox.com).*

Reference: Vincent et al. *shapeViewer, a software for the scientific mapping and morphological analysis of small bodies*, LPSC, (2018).

1.4 License

shapeViewer is written in C and makes use of several Free libraries:

- [FLTK](#) for the graphical interface,
- [OpenGL](#) for 3D rendering,
- [cJSON](#) for reading JSON files,
- [SPICE](#) for orbital calculations,
- [stb_image_write](#) for saving screenshots.

shapeViewer is distributed under a Simplified BSD license:

Copyright (c) 2010-2023, Jean-Baptiste Vincent

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GRAPHICAL INTERFACE

shapeViewer aims to be intuitive and ready to use out of the box. The most common functions are accessible directly from the graphical user interface, and via keyboard shortcuts. For instance to simulate observations at a given time: simply click on “View/Set time”. Most menu items have an associated keyboard shortcut, explicitly indicated in the menu entry. For instance, “View/Set Time” uses the letter [T].

When starting the software, a mission is already preconfigured and you can immediately visualize the field of view of different instruments for any epoch covered by the mission scenario.

2.1 Main window

The main window displays a 3D view of the shape(s) currently loaded. **Left-click and drag** to orient the view, **right click** to use the current tool. By default, this will give you the coordinates of any point on the surface.

Note:

At start, objects are displayed in the following orientation:

Main object:

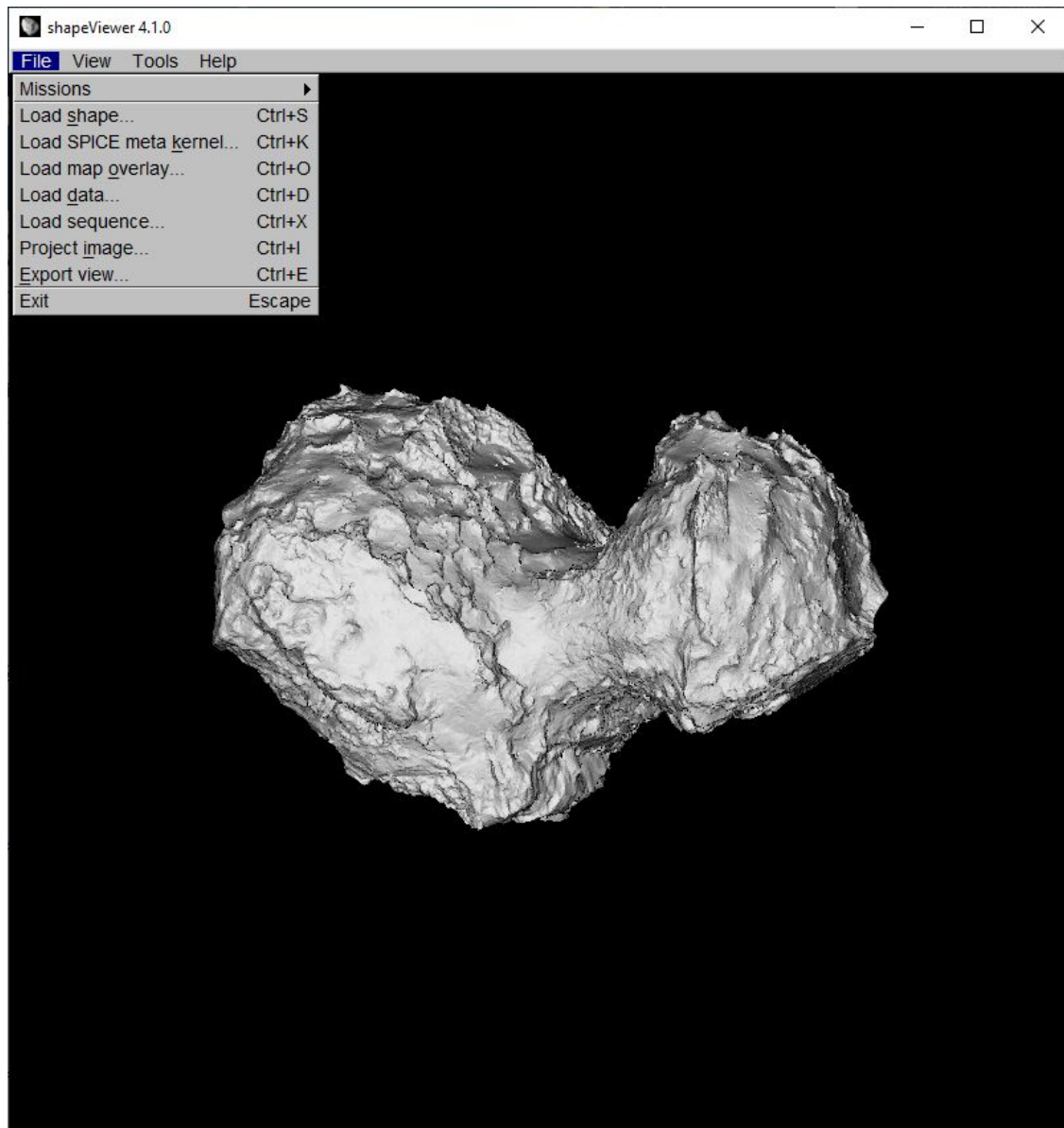
- Z axis (positive, counter-clockwise spin) is pointing UP
- X axis (longest, often the prime meridian) is pointing towards the OBSERVER

Secondary objects:

- Z axis (positive, counter-clockwise spin) is pointing UP
- X axis (longest, often the prime meridian) is pointing towards the MAIN OBJECT

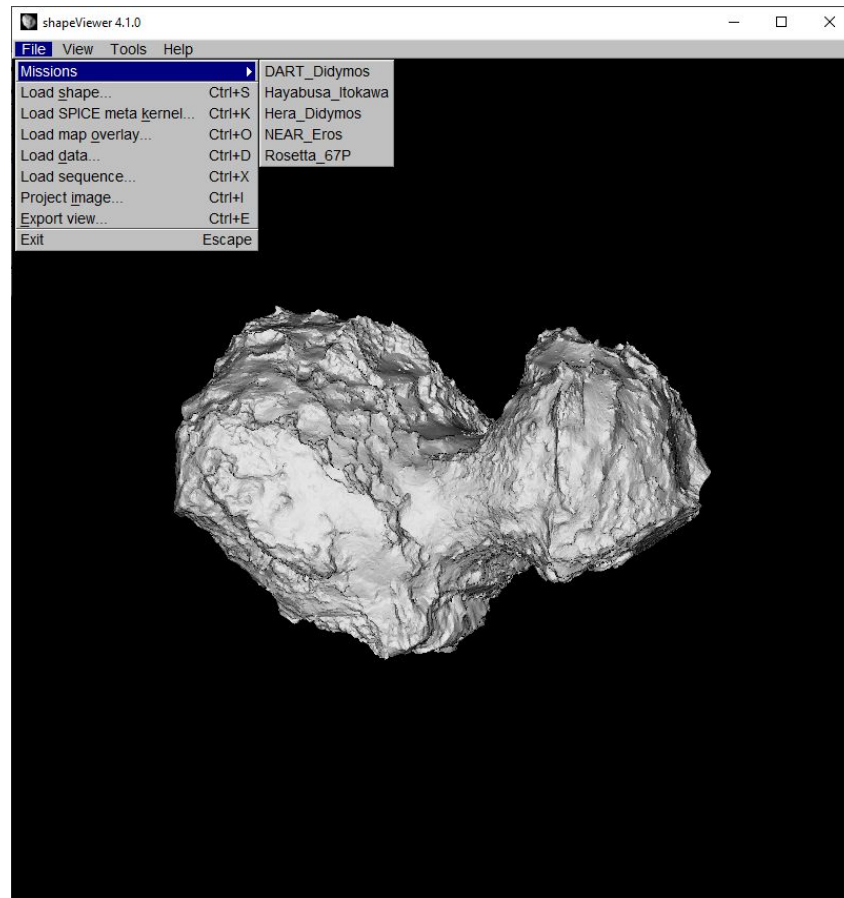
This mimics the most common case of a secondary rotationally locked to its primary (e.g. Earth/Moon), but **does not represent the real relative orientation of each object**. Use “View/Set Time” to see the real world orientation.

2.2 File menu



The “File” menu lets you load additional data in the software.

2.2.1 Missions



A mission package is a bundle of files that contains everything needed for visualizing a given scenario. This includes shape models, [SPICE kernels](#), instrument details, and so on.

Each package comes with a JSON file which summarizes the relevant information. When starting, *shapeViewer* will scan the “missions” sub-folder and detect available. When closing the software, it will remember the last mission you were working with.

The structure of a mission package is self-explanatory. To create your own, simply copy an existing JSON file and update it to describe your project.

2.2.2 Shapes

shapeViewer was originally designed to visualize **SHAPE** files. Although not a common file format in the 3D graphics industry, this has become a standard description for asteroids and comets.

You can find more information on the SHAPE format in the [Database of Asteroid Models from Inversion Techniques](#). Here is a short description:

SHAPE format

3D models are represented as a collection of triangular surface facets. Shape files are a textual representation of those triangles with the following format:

- The first line of the file gives the number of vertices and facets,
- then follow the x, y, z coordinates of each vertex, one vertex per line

- then for each facet the order numbers of facet vertices (anticlockwise seen from outside the body), one facet per line

In addition, *shapeViewer* also support **OBJ** and **PLY** files, with some restrictions:

- OBJ files can be of any complexity but only vertices and naked facets are read. Textured facets will not be recognized. Other data is ignored.
- PLY format is very versatile, *shapeViewer* 4 supports headers of the two following types:

Header for PLY Type 1 (Geometry only):

```
ply
format binary_little_endian 1.0
element vertex ***
property float x
property float y
property float z
element face ***
property list uchar int vertex_indices
end_header
```

Header for PLY Type 2 (Geometry + vertex color):

```
ply
format binary_little_endian 1.0
element vertex ***
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
element face ***
property list uchar int vertex_indices
end_header
```

2.2.3 Load SPICE meta kernel

This function lets you replace the current set of SPICE kernels with a new one. In practice, it is recommended to change the default meta kernel in the mission package JSON file.

2.2.4 Load map overlay

Load an image (JPG, PNG) that will be interpreted as a 2D equirectangular map of the current shape model. The image data will be used as a texture for the whole shape.

2.2.5 Load data

User data files can be loaded and displayed. So far, *shapeViewer* only supports a simple ASCII format to describe the data, similar to the OBJ format for 3D files.

It is foreseen that a new JSON-based data interface will be added in future versions of the software.

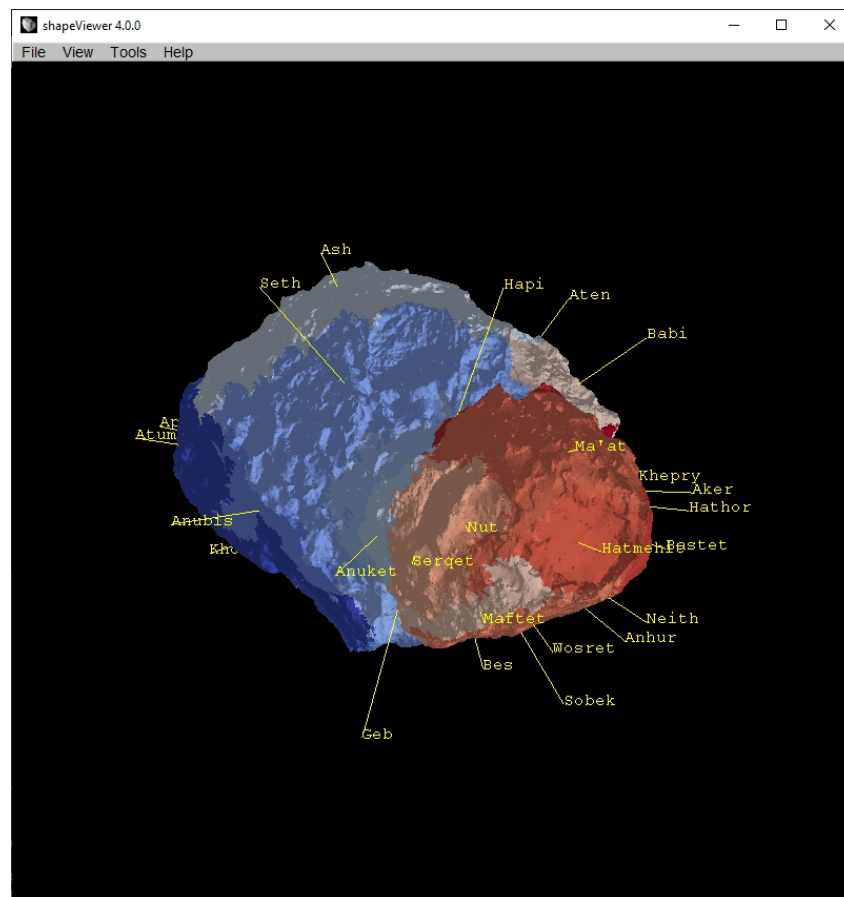


Fig. 1: Example of a data file showing region names and borders on comet 67P.

Data is provided as a text file, with one entry per line. Each line starts with the type of data, followed by the relevant information.

Version 4.1.0 supports 4 basic types:

- “ms” or “mc” for markers,
- “f” for facet data,
- “g” for geographic data.

A line starting with any other character will be ignored.

Note: It is possible to mix several types of data in the same file, and/or load multiple files. One could for instance load the thermal map of a given regions, and add coordinate markers to point at areas of interest.

“markers” data:

Displays a line marker and a text label at the given coordinates. Vector information can be provided in spherical (“ms”) or cartesian (“mc”) form.

```
ms latitude longitude altitude data
```

```
| "latitude" and "longitude" in degrees, altitude in km
| "data" is a string, the first space-separated token may be a float number
```

```
mc x y z data
```

```
| x y z are 3D cartesian coordinates, in km
| "data" is a string, the first space-separated token may be a float number
```

“facet” data:

Paint the given facets with the given color.

```
f facet_id R G B data
```

```
| "facet_id" is the facet index in the shape model, starting at zero
| "R, G, B" are red-green-blue floats, between 0.0 and 1.0
| "data" is a string, the first space-separated token may be a float number
```

Line of “geographic” data:

Assign data to the shape model, in areas defined by their central coordinate and angular size. The display will use the current color scheme, which can be changed afterward using the *Integrated Console*.

```
g latitude longitude angular_area data
```

```
| "latitude, longitude", in degrees, define a point of interest on the shape model
| "angular _area" is the radius of the area that should be considered, in degrees
| "data" is a string, the first space-separated token must be a float number
```

Note: Loading this type of data can be slow, depending on the number of points and angular size of the regions of interest. Once loaded, it is recommended to export the data to a “facet” file which can be loaded almost instantly in future sessions. See *Integrated Console* for further information.

2.2.6 Load sequence

Sequences of images can be defined in an JSON file. Once loaded, it will be used to automatically generate simulated images for the given sequence. You must set up what to display (labels, gravity, axes, camera footprints, ...) before loading the sequence file.

Each observation in the sequence must define an instrument and a time of observation. An optional texture can be specified and will be projected on the shape.

Example of a sequence.json file:

```
{
  "sequence": [
    { "camera": "WAC", "time": "2015-01-03T05:35:00", "texture": "WAC_image.jpg" },
    { "camera": "NAC", "time": "2015-01-03T05:35:00" }
  ]
}
```


2.2.7 Project Image

Load an image corresponding to the current field of view, and project it on the shape. This requires that you have set a time and a camera, the program will remind you to do so if you didn't.

Once projected, the image can be visualized in all views provided by *shapeViewer*. This means in 3D, but also in the various map projections available in the “View/Projection” menu.

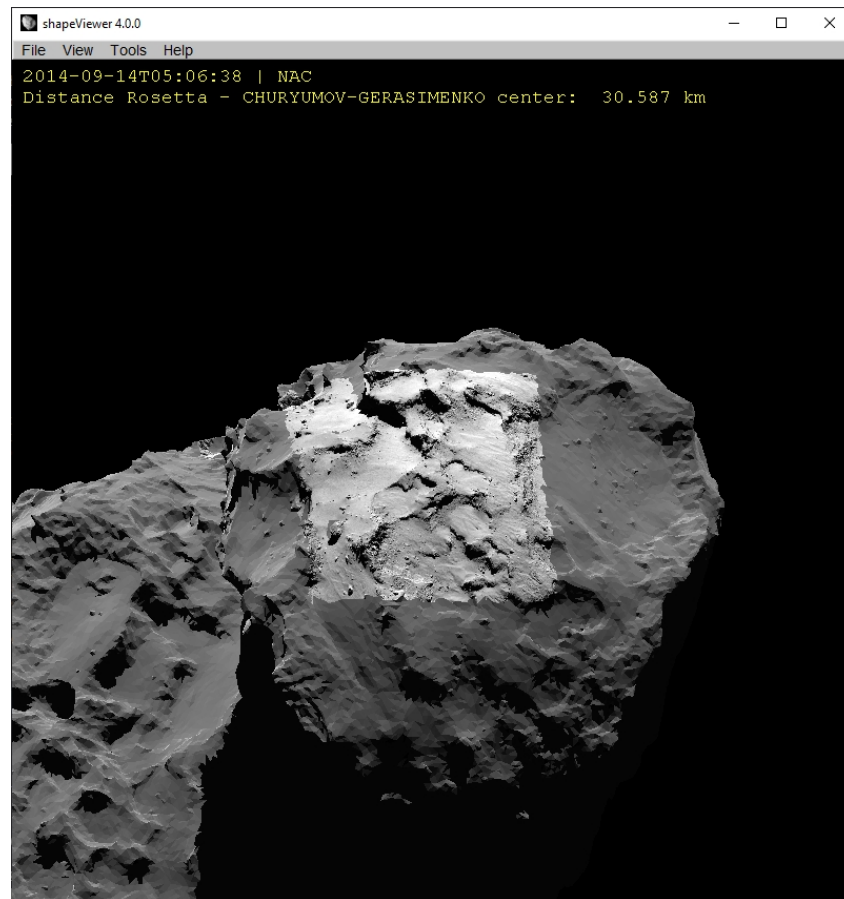


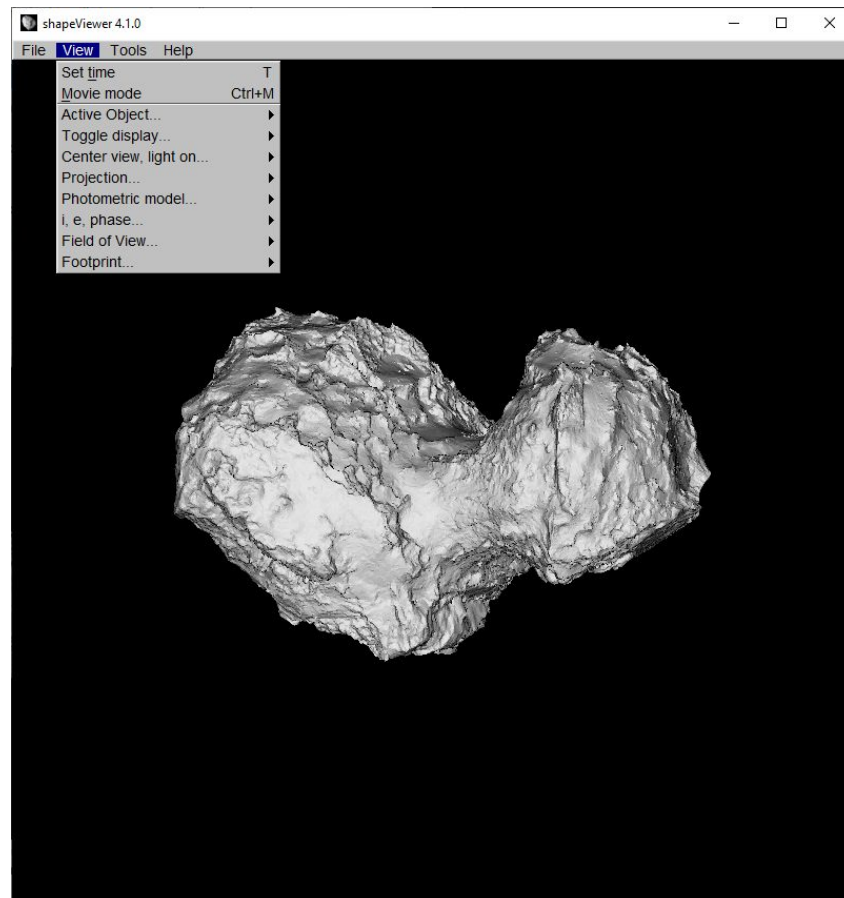
Fig. 2: Note: this shape model is shaded with a darker photometric model, to illustrate where the image has been projected. In general, you should use a photometric model that matches the data.

2.2.8 Export View

Export the current view to a PNG file in the *shapeViewer* folder.

This exported view is a PNG using the RGBA color encoding, i.e. the data resolution is limited to 32 bit colors (65536 values) or 8-bit (256) shades of gray. High depth photometric rendering is planned in the near future.

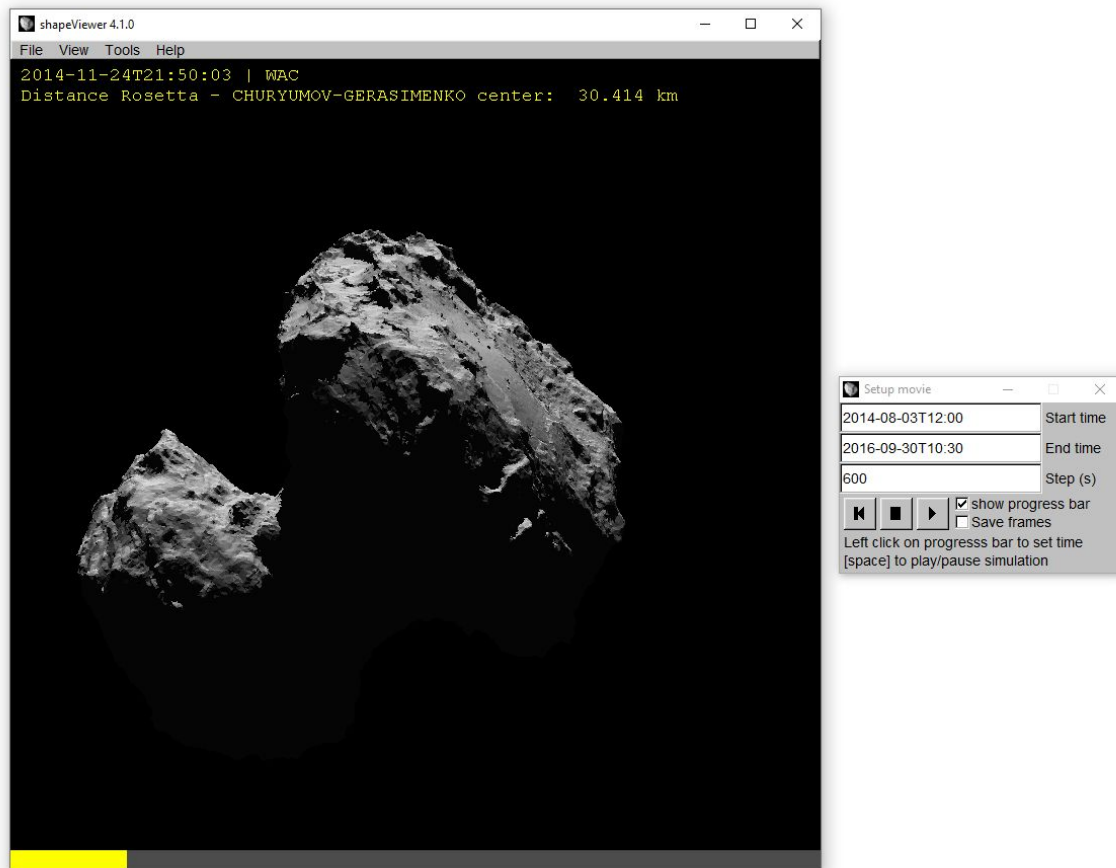
2.3 View menu



2.3.1 *Set time* and *Movie mode*

Probably the most used features. “Set time” let you define an epoch for which you want to simulate what instruments are going to observe. “Movie mode” does the same for the whole mission, and let you select the time displayed simply by clicking on a progress bar at the bottom of the screen (see figure below).

Note that all views are interactive. You can always pause the movie to visualize the shape from another angle than the one displayed, or activate different viewing options interactively.



By default the movie mode is configured to display the full mission. You can update the boundaries of the movie in the option windows that appears automatically. Selecting the option “save frames” will export each frame as a PNG file in the “screenshots” folder.

2.3.2 Active Object

The 3D mode displays all objects at once but that is not always possible in other view modes, e.g. map views. In that case, this menu option let you select the object that should be displayed in priority.

Right-clicking an object in 3D view will automatically set that object as active.

2.3.3 Toggle display

shapeViewer can display a lot more information than just a shape model:

- axes, coordinate system
- elevation
- gravity, slopes
- surface roughness
- direction of the Sun, Observer
- incidence, emission, phase

All these options are activated via the “Toggle” menu.

Note: Some information, e.g elevation, is displayed using a color scale. For multiple bodies with very different dimensions (e.g. Didymos and Dimorphos), it is difficult to find a color scale that would fit both objects so *shapeViewer* will adapt the scale to the **current active object**. Simply right-click on any object to st it to active

Color scale and limits can be changed using the *command line interface*.

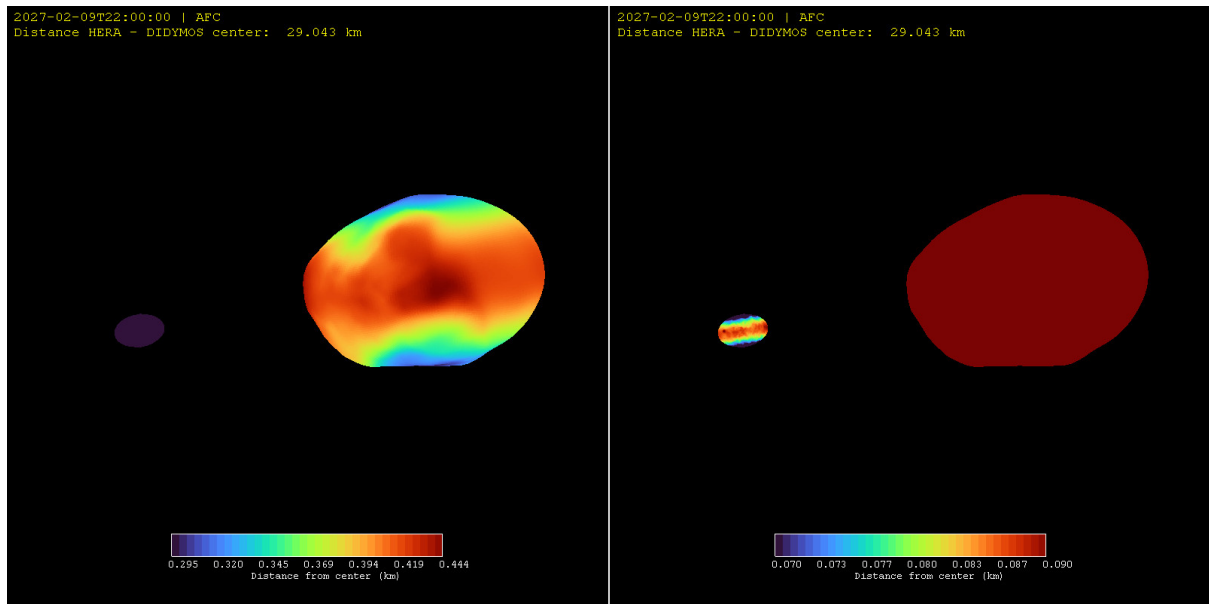


Fig. 3: Distance from center of shape in a binary asteroid system. Left: color scale adapted to dimensions of the main object. Right: color scale adapted to dimensions of the secondary.

A note on gravity

shapeViewer provides a model of **effective gravitational acceleration** (gravity corrected with centrifugal force). The acceleration is calculated for each facet using the model by Cheng+2012, equivalent to the classical Werner&Scheeres+1997 but faster to calculate. *shapeViewer* assumes a constant density throughout the object.

Results are stored in the “gravity” subfolder. Deleting files in that folder will trigger a prompt for input parameters when trying to display gravity again. Effectively, this allows you to recalculate the acceleration. Note that this is a slow process !

A note on roughness

Surface roughness is defined as the mean angle between a facet normal vector and its neighbours. It is calculated when the shape is loaded. Details on the algorithm and examples for several small bodies are available in Vincent+2023.

2.3.4 Center view

The items under this submenu will reset the view to preset configurations, by forcing the Sun and Observer position along one of the shape axes (X,Y,Z, -X,-Y,-Z).

2.3.5 Projection

This menu gives you access to several map projections of the active shape.

- Equirectangular projection
- Kavrayskiy VII projection
- Orthographic (North/South) projection

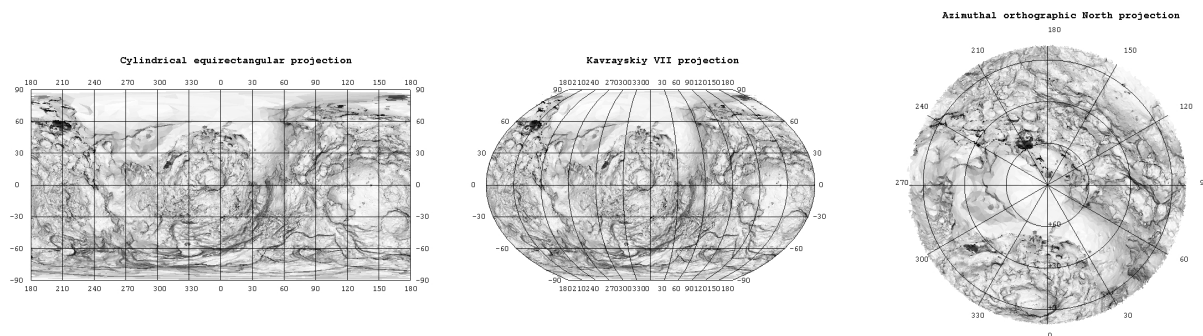


Fig. 4: Examples of map projections rendered with *shapeViewer*

2.3.6 Photometric model

This menu gives you some control on how the shape is being rendered. A few simple photometric models are available in v4 to calculate the reflectance F of the surface:

```
LAMBERT:          F = u0
LOMMEL_SEELIGER: F = u0/(u0+u)
LUNAR-LAMBERT:    F = u0(1 - L) + 2L * u0/(u0 + u), L in [0,1]
MINNAERT:         F = u0^k * u^(k - 1), k in [0,1]
```

with $u0 = \cos(\text{incident angle})$ and $u = \cos(\text{emergent angle})$.

Better model will be included in future versions of the software.

2.3.7 *i, e, phase*

Display a representation of the incidence, emission, and phase angle at the time of observation. Each angle can be shown individually, or together. In the latter case, angles are encoded in the RGB values of the image (incidence => Red, emission => Green, phase => Blue) with a resolution of 256 values per color.

2.3.8 Field of View

Select the field of view (instrument) to be displayed. The current view will be adjusted to fit this FoV in the frame. This menu is automatically populated from data defined in the mission package.

2.3.9 Footprint

This setting will overlay a box representing an instrument field of view on top of the current scene. It is possible to project the field of view boundaries onto the 3D shape using the *command line interface*.

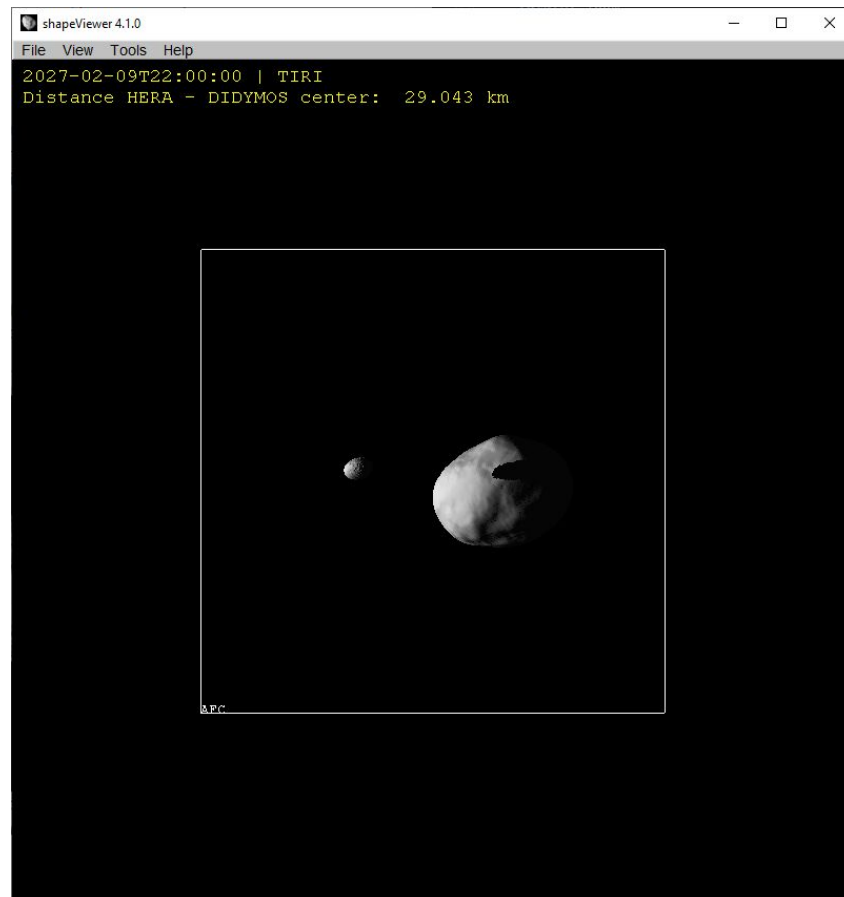
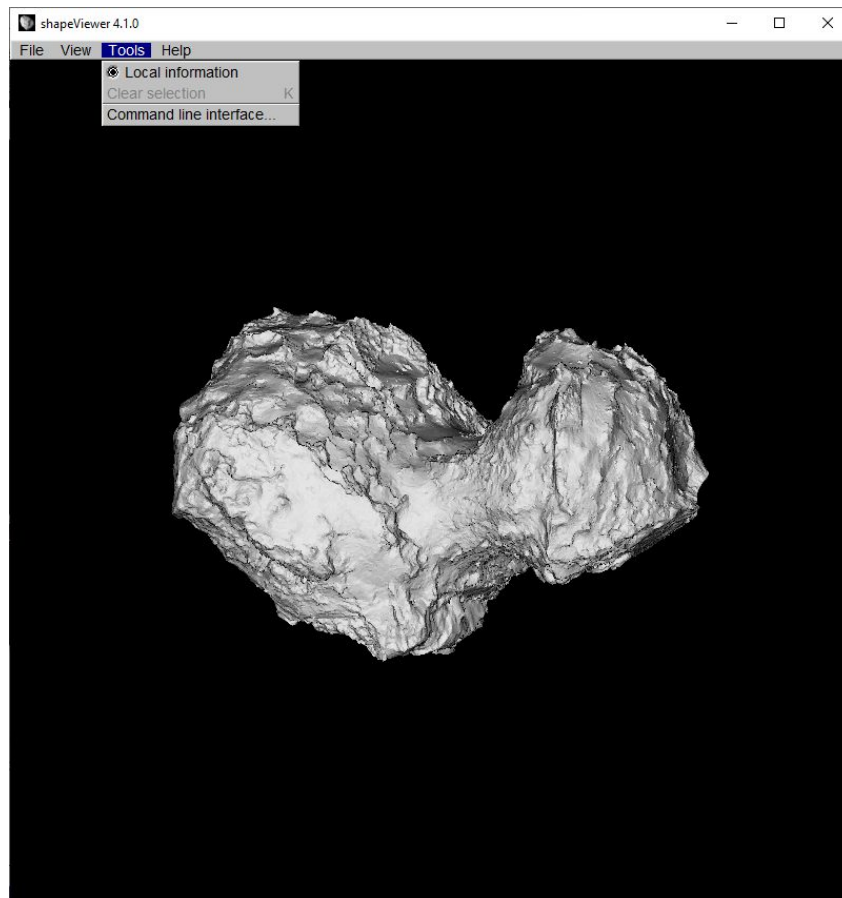


Fig. 5: Example of display showing the **field of view** of Hera's TIRI instrument as full frame, with the additional **footprint** of Hera's AFC instrument.

2.4 Tools menu



This menu controls what happens when you *right-click* an object. Most tools are being rewritten and are currently disabled.

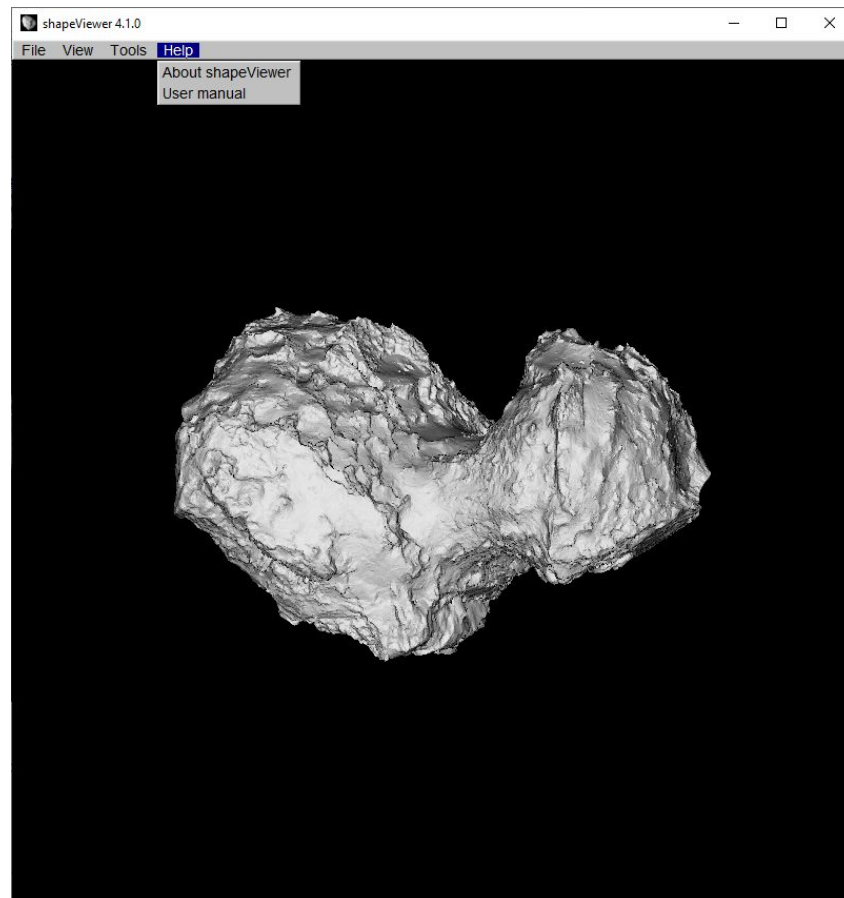
2.4.1 Local information

Right-clicking on a shape model gives you access to local information. The amount of information depends on the current selection in the “View menu” and the data available. You can remove this selection by using the menu item “Tools/Clear selection” or the keyboard shortuct [k].

2.4.2 Command line tools

Open a *command line interface* for advanced options. The command line can also be opened/closed with either the keys [F12] or [=].

2.5 Help menu



2.5.1 *About shapeViewer*

Some information about the current version.

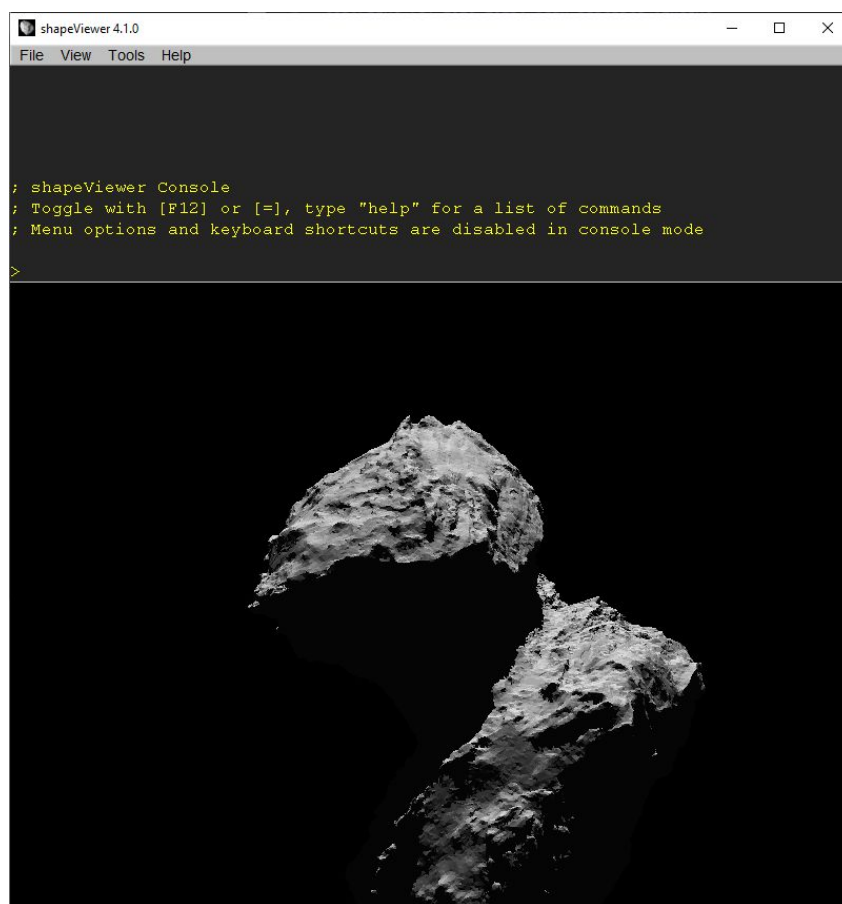
2.5.2 *User Manual*

Open a PDF version of this user manual.

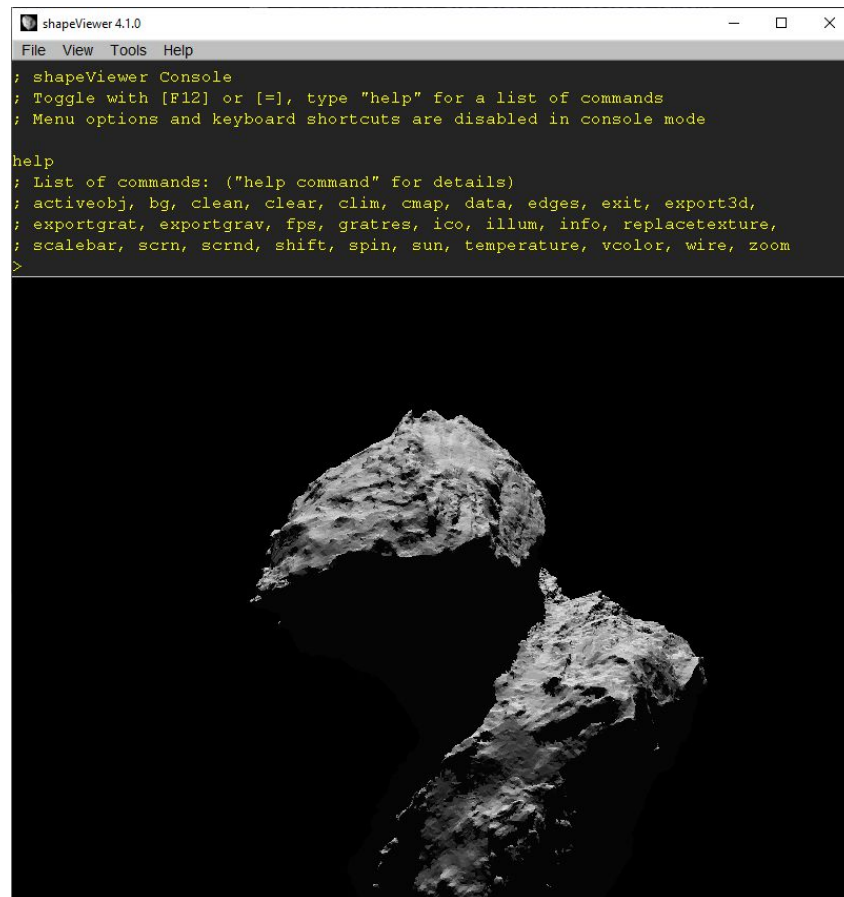
INTEGRATED CONSOLE

shapeViewer can do much more than simply displaying a shape model. Many functions that may not be relevant to all users are available through a command line interface (CLI) embedded in the program.

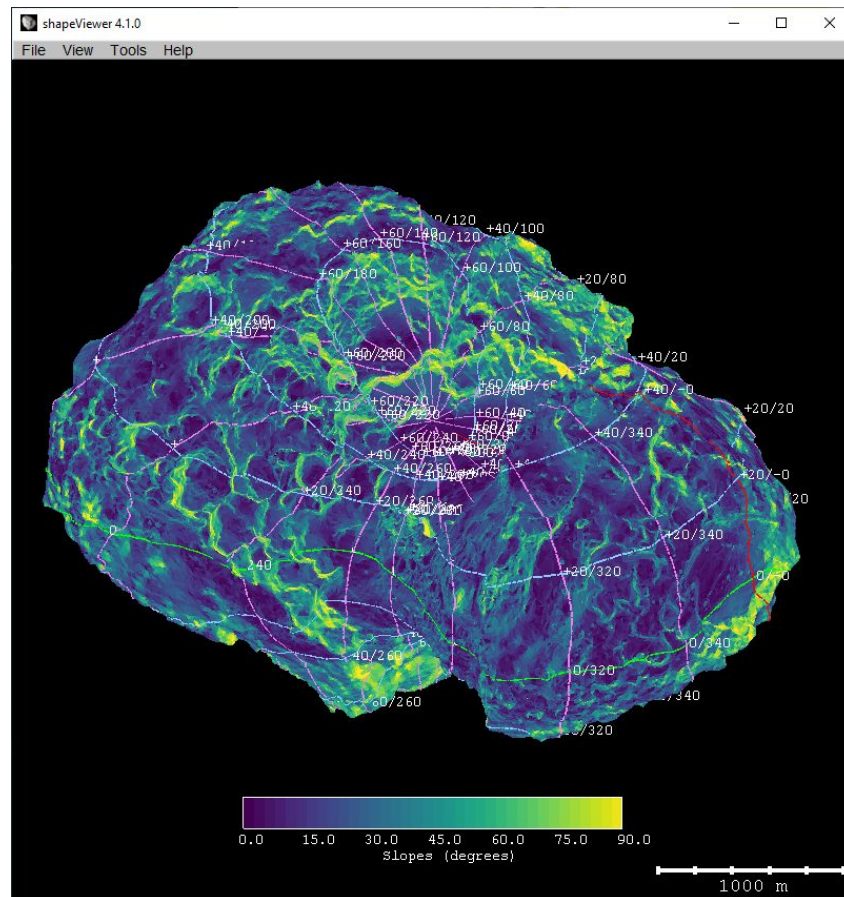
This CLI can be activated from the “Tools” menu or with either the [=] and [F12] keys.



Typing “help” in the command line will give you a list of all available commands, and how to use them.



The following image, for instance, displays the shape model of comet 67P. A graticule with a resolution of 20 degrees in latitude/longitude has been added, as well as a scale bar with 1000m reference distance. Colors describe the gravitational slopes, but the default color map (*turbo*) has been changed to *viridis* and the limits (0-180) have been reduced to 0-90.



All this can be achieved with the following commands:

```
[c] or "View/Toggle/Coordinates"
[h] or "View/Toggle/Slopes"
[=] or "Tools/Console"
```

then, in the console:

```
gratres 20      (change the graticule resolution)
cmap viridis    (change colormap)
clim 0 90       (change colormap limits)
scalebar 1000   (add a scale bar with 1000m reference distance)
```


PUBLICATIONS

Since 2010, shapeViewer has been referenced in many papers and conference talks. Here is a selected list of peer-reviewed publications which benefited from the software.

2023

- Korda et al, (433) Eros and (25143) Itokawa surface properties from reflectance spectra, A&A (2023)
-

2022

- Li et al, Numerical approach to synthesizing realistic asteroid surfaces from morphological parameters, A&A (2022).
 - Jindal et al, Topographically Influenced Evolution of Large-scale Changes in Comet 67P/Churyumov-Gerasimenko's Imhotep Region, PSJ (2022).
-

2020

- O'Rourke et al. The Philae lander reveals low-strength primitive ice inside cometary boulders, Nature (2020).
-

2019

- Birch et al, Migrating Scarps as a Significant Driver for Cometary Surface Evolution, GRL (2019).
 - Vincent et al, Bouncing boulders on comet 67P, EPSC (2019).
-

2018

- Vincent et al, shapeViewer, a software for the scientific mapping and morphological analysis of small bodies, LPSC (2018).
-

2017

- Birch et al, Geomorphology of comet 67P/Churyumov-Gerasimenko, MNRAS, 469:S50-S67 (2017).
 - Gicquel et al, Modeling of the outburst on 2015 July 29 observed with OSIRIS cameras in the Southern hemisphere of comet 67P/Churyumov-Gerasimenko, MNRAS 469:S178-S185 (2017).
 - Masoumzadeh et al, Opposition effect on comet 67P/Churyumov-Gerasimenko using Rosetta-OSIRIS images, A&A 599:A11 (2017).
 - Oklay et al, Long-term survival of surface water ice on comet 67P, MNRAS 469:S582-S597 (2017).
-

- Pajola et al, The pristine interior of comet 67P revealed by the combined Aswan outburst and cliff collapse, *Nature Astronomy* 1:0092 (2017).
 - Vincent et al, Constraints on cometary surface evolution derived from a statistical analysis of 67P's topography, *MNRAS* (2017).
-

2016

- Barucci et al, Detection of exposed H₂O ice on the nucleus of comet 67P as observed by Rosetta OSIRIS and VIRTIS instruments, *A&A*, 595:A102 (2016).
 - Oklay et al, Variegation of comet 67P/Churyumov-Gerasimenko in regions showing activity, *A&A* 586:A80 (2016).
 - Oklay et al, Comparative study of water ice exposures on cometary nuclei using multi spectral imaging data, *MNRAS* 462:S394-S414 (2016).
 - Vincent et al, Are fractured cliffs the source of cometary dust jets? Insights from OSIRIS/Rosetta at 67P, *A&A* 587:A14 (2016).
 - Vincent et al, Summer fireworks on comet 67P, *MNRAS*, 462:S184-S194 (2016).
-

2015

- Masoumzadeh et al, Photometric analysis of Asteroid (21) Lutetia from Rosetta-OSIRIS images, *Icarus* 257:239-250 (2015).
 - Vincent et al, Large heterogeneities in comet 67P as revealed by active pits from sinkhole collapse, *Nature* 523:63-66 (2015).
-

2014

- Vincent et al, Crater depth-to-diameter distribution and surface properties of (4) Vesta, *PSS* 103:57-65 (2014).
-

2012

- Vincent et al, Physical properties of craters on asteroid (21) Lutetia, *PSS* 66:79-86 (2012).
-

CHANGELOG

Each version of *shapeViewer* is numbered as **major.branch.minor**, following a semantic versioning convention:

- *major* versions introduce significant code and interface changes, breaking compatibility
- *branches* add major functionality, while aiming to remain compatible with previous versions
- *minor* versions indicate small updates in functions, data, performance, and bug fixes

Short history:

- 2010 v1: first release, with only a command line interface.
 - 2011 v2: added a full graphical interface to be more user-friendly. This version evolved with the Rosetta mission, adding features as needed by the science team.
 - 2018 v3: *shapeViewer* became a more general tool, taking its input data from “mission packages” provided with the software or user-generated.
 - 2021 v4: new rendering engine to allow multiple objects to be displayed (e.g. for missions to binary asteroids)
-

VERSION 4 “MULTIPLE BODIES”

6.1 BRANCH 4.1 “JSON configuration”

Warning: although this branch is not a major version, it does break compatibility with previous releases as the format of configuration files have changed from XML to JSON.

shapeViewer 4.1.1 - 10/01/2024

modified:

- reverted to old image projection method. New method was faster but had issues with multiple bodies in the field of view. Will be fixed in version 4.2.
- improved rendering of shadows at large distances

shapeViewer 4.1.0 - 14/07/2023

added:

- new user data format (lat, lon, angular coverage)
- console tool to export this projected data for further analysis
- roughness model, calculated when loading a shape
- support both square and rectangular field-of-view
- mission packages for Hayabusa, Hayabusa2, NEAR, DART, Hera
- autosave configuration

modified:

- debug mode is now started by calling "shapeViewer.exe debug" instead of `↪` modifying a config file
- converted all configuration files from XML to JSON
- shapeViewer automatically detects mission packages and populates menus `↪` accordingly
- switch to a new mission in one click, shapeViewer will remember your last choice
- updated existing mission packages to the new format
- secondary objects oriented as if rotationally locked to primary at startup
- pre-calculated gravity/slopes included in mission packages
- removed the check-for-update function, current code not compatible with HTTPS `↪` server
- removed user-defined screenshot size. Will be reintroduced with future FITS `↪` renderer

fixed:

- shadows sometimes not rendering at startup
- image resolution not showing
- front plane clipping issues at close distances

6.2 BRANCH 4.0 “Multiple bodies”

shapeViewer 4.0.3 - 05/07/2022

modified:

- Hera/Didymos/Dimorphos kernels updated to latest ESA planning kernels (v1.3.0)
- Updated documentation

fixed:

- Cleaned up typos in the command line tools

shapeViewer 4.0.2 - 21/06/2022

modified:

- Hera/Didymos/Dimorphos kernels updated to latest ESA planning kernels (v1.2.0)
- Tools that are currently being upgraded (topo profile, painting brush) are not visible anymore in the menu
- Screenshot generation is now much faster
- Screenshot resolution can be modified with the command line tools
- Cleaned up typos and outdated functions in the command line tools

fixed:

- gravity vectors using the wrong color scale
- data file crashing the program when trying to load an out-of-bound facet id

shapeViewer 4.0.1 - 16/02/2022

modified:

- Hera/Didymos/Dimorphos kernels manually updated to correct spin axis orientation

shapeViewer 4.0.0 - 30/05/2021

added:

- support for multiple objects, in preparation for the DART/Hera missions
- Hera is now the default mission scenario
- objects can be selected with either right-click, the "View" menu, or with the command "activeobj"
- individual objects can be replaced with other shapes while program is running
- colormaps automatically adjust to the active object
- rewrote all documentation

modified:

- New mission config files. This breaks compatibility with earlier shapeViewer versions
- removed the dependency in the OpenIL/DevIL library. No change for the user, faster screenshots
- shadow calculations now 100x faster, activated by default but can be disabled from "View" menu
- image projection also 100x faster than before
- reorganized items in menu bar
- rewrote the movie mode and added an interactive progress bar
- movie mode set to the full mission epoch by default
- console can now be started from the graphical interface (and still with keys [=] or [F12])
- "info" command will display information on the currently selected object

(continues on next page)

(continued from previous page)

- harmonized syntax of many commands
- upgraded SPICE to version N0066
- "spin" function removed from GUI (available in console, see user manual)
- performance improvement on map projection and graticule functions

fixed:

- text labels sometimes not checking properly against depth buffer.
This is an issue with FLTK 1.4, fixed by reverting to latest FTLK stable (v1.3.

↪ 6)

- viewing geometry not reset properly when loading new mission package
- error in display of surface area

VERSION 3 “MULTIPLE MISSIONS SUPPORT”

7.1 BRANCH 3.0 “Mission packages”

shapeViewer 3.0.4 - 22/02/2021

added:

- "exportgrat" command to write a grid of the current graticule to a file
- "exportgrav" command to export the surface gravity in spherical coordinates
- a progress bar is now displayed for the more expensive calculations (shadows, image projection, gravity)
- if gravity/slopes data exists, selecting a point with right-click in ↪ "Coordinates" mode will display the local gravity/slopes value

modified:

- upgraded FLTK to version 1.4, no change for the user

fixed:

- no more loss of precision when loading gravity data from a file

shapeViewer 3.0.3 - 01/09/2019

added:

- "turbo" colormap is the new default (see Google AI blog)
<https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html>

modified:

- wireframe set to black when background color is white
- UP and DOWN keys in console navigate history of commands (instead of both ↪ commands+results)

fixed:

- incorrect position of a camera footprint in the field of view of another camera. Non-aligned boresights are now properly modeled.

shapeViewer 3.0.2 - 27/07/2018

added:

- support for Hayabusa 2 mission (config files not publicly available for now)
- "activecolor" command to define the color of selected lines/facets
- "paintfov" command to paint the facets currently in the FOV with the active ↪ color
- "outlinefov" command to mark the projected contour of the current FOV ↪
- ↪ (EXPERIMENTAL)
- "exportfov" command to write facets currently in FOV to a data file

(continues on next page)

(continued from previous page)

```
- "clean"          command to reset the colors
modified:
- "export" command renamed to "export3d"
- "clim" command now works also when displaying angles (i, e, phase)
fixed:
- corrected syntax for loading "facet" data in this help file
- wrong FOV definition in Lutetia package
- typos in user manual
```

shapeViewer 3.0.1 - 26/06/2018

```
-----

added:
- "sun" command to define the subsolar point
- "clim" command to define the boundaries of the current color map
modified:
- simplified output format of topographic profiles
- improved loading of large files, and startup time
fixed:
- sequence files with no textures are now rendered properly once again
```

shapeViewer 3.0.0 - 16/03/2018

```
-----

added:
- new concept of "mission packages", which let the user define and load various ↵
↵mission scenarios
- info command displays more information about the shape
- loading of multiple data types, to easily mark regions or landmarks
- "Help" menu now offers an option to check for shapeViewer updates on the server
- support of OBJ files
modified:
- Removed external CLI interface as it was quite broken and very rarely used.
- reorganized menu bar, moved "Set time" action to "View" menu
- console now accepts all characters you can type, upper and lower case
- [Esc] key disabled in console, so that you do not close shapeViewer when ↵
↵attempting
  to leave the console
fixed:
- no more out of memory when loading large models (up to ~1 million facets).
- proper freeing of memory upon exiting the software
```

VERSION 2 “GUI”

8.1 BRANCH 2.9 “Embedded console and usability improvements”

shapeViewer 2.9.6 - 30/11/2017

fixed: corrected the wrong projection of some WAC images

fixed: "real colors" options appeared toggled off in GUI

shapeViewer 2.9.5 - 15/11/2017

modified: updated SPICE kernels to the official ESA 1.1.0 release

modified: default 3d shape is now the [DLR](http://europlanet.dlr.de/Rosetta/)
→ SHAP7 model

(Preusker et al, A&A, 2017)

modified: time/distance view now displays distances spacecraft-target center and
→ spacecraft-surface

modified: the coordinates tool now displays the altitude of the selected point

shapeViewer 2.9.4 - 04/10/2017

fixed: some textured facets not showing in map mode

modified: updated SPICE kernels to the official ESA 1.0.0 release

added: graticule resolution can now be modified in map view as well as in 3D view

added: "info" command to get some information from the shape file

added: "export" command to export textured shape as an OBJ file

shapeViewer 2.9.3 - 10/07/2017

fixed: "set time" dialog now disabled in movie mode

fixed: "set time" dialog will remember time of last frame after exiting movie mode

fixed: scalebar length is correct even if target not intersecting boresight

modified: complete rewrite of the graticule function. It is now faster, and
→ customizable

modified: similar rewrite of the topographic profile tool, now guaranteed to catch
all facets between two points

modified: 'View Axes' now shows both positive (solid line) and negative (dashed line)
→ axes

modified: replaced gravity model with Cheng 2012. Slower but more accurate

modified: some operations parallelized with openmp, slight gain of performance

(continues on next page)

(continued from previous page)

modified: slopes displayed in colors by default, use CLI to change colormap
 added: inline help in command line interface
 added: various colormaps accessible from the console

shapeViewer 2.9.2 - 11/05/2017

 Release 42 ! 2500 days since version 1.0 :)
 fixed: improved and sped-up raytracing. Shadows are now very accurate
 fixed: shift camera with arrow keys when time is set
 fixed: improved panning and zooming in all map modes
 modified: shapeViewer remembers the last valid time set
 modified: improved display of graticule, some issues with concavity remain to be fixed
 added: implemented Kavrayskiy VII conformal map projection
 added: on screen display of sub-Sun and sub-S/C coordinates when Sun or S/C display
 ↪ is activated
 added: many new commands in console

shapeViewer 2.9.1 - 31/03/2017

 fixed: image projection sometimes repeating the texture
 fixed: error message appearing when displaying camera footprint in movie mode
 modified: map view can be zoomed in/out and panned with the mouse

shapeViewer 2.9.0 - 26/03/2017

 fixed: application closing when exiting movie mode
 fixed: improved rendering
 added: generic colorbar for all modes (including 3d view)
 added: read basic PLY files (binary, little endian, with or without vertex color
 ↪ information)
 added: console interface embedded in shapeViewer (toggled with [F12])
 console implements new displays and options (fps, scalebar, vertex color,
 ↪ edges, ...)
 console is undocumented in this version and highly experimental

8.2 BRANCH 2.8 “Image projection/texturing”

shapeViewer 2.8.0 - 25/01/2017

 added: Rosetta images can now be overlaid on the shape model, after time and camera
 ↪ have been set
 This release supports only .JPG, .PNG, and .BMP images.
 added: "Facet" files can now contain data which can be displayed in 3D
 modified: Rewrote part of the camera code to have more intuitive interaction in all
 ↪ views
 modified: If selected, camera footprint now showing at all levels of zoom
 modified: shapeViewer application is now resizable
 modified: Time can be input with hours/min/sec separated by ':' or '.'
 modified: updated compiler to gcc 5.3

(continues on next page)

(continued from previous page)

```
modified: updated SPICE library to allow processing of fast kernels (NAIF Types 6 and ↵
↵19)
```

8.3 BRANCH 2.7 “Public release”

```
shapeViewer 2.7.1 - 12/10/2016
```

```
-----
```

```
modified: Harmonized console output
```

```
modified: "Surface Area" tool now exports also facet normal vectors.
```

```
shapeViewer 2.7.0 - 04/10/2016
```

```
-----
```

```
First public release
```

```
Restricted time set to 2014-07-01T12:00 to 2016-09-30T12:00
```

```
Shape provided by default is latest NavCam model
```

```
modified: CLI now uses the right FOV
```

8.4 BRANCH 2.6 “CLI interface”

```
shapeViewer 2.6.5 - 20/09/2016
```

```
-----
```

```
added: Rosetta NavCam FOV [Ctrl+r], and footprint [Alt+r]
```

```
added: label to footprint drawing
```

```
modified: frustum culling between distance/100 and distance*10 instead of [1km, ↵
↵1000km].
```

```
shapeViewer 2.6.4 - 11/09/2016
```

```
-----
```

```
modified: Viewing geometry/time preserved when loading new shape.
```

```
shapeViewer 2.6.3 - 02/09/2016
```

```
-----
```

```
added: FAC "Far away camera": 90deg FOV camera for full scene view. Activate with ↵
↵[Ctrl+f]
```

```
modified: CLI interface now always render in FAC mode.
```

```
shapeViewer 2.6.2 - 15/06/2016
```

```
-----
```

```
added: NAC and WAC footprints can be toggled with shortcuts [Alt+n], [Alt+w].
```

```
fixed: Rendering used ortho projection instead of perspective, which created a lens ↵
↵distortion artifact.
```

```
modified: The perspective view may distort the rendering when resizing the window.
          As resizing was rarely used, it is now disabled.
```

shapeViewer 2.6.1 - 07/06/2016

added: use keys {shift+ X,Y,Z] to view and illuminate shape from axes -X, -Y, -Z.
modified: updated SPICE library to use Type 19 kernels (much faster)
modified: drawing pipeline. Colors/shading calculated only when needed, and not each
→ frame.
modified: better raytracing, not 100% accurate yet, but good and fast enough to be
→ used.

shapeViewer 2.6.0 - 23/03/2016 - "CLI"

added: first implementation of a command line interface
added: option to hide labels when exporting view
fixed: camera footprints not correct when zooming
modified: increased line thickness for camera footprints

8.5 BRANCH 2.5 “Sequence mode, and usability improvements”

shapeViewer 2.5.10 - 17/08/2015 - "Perihelion 2"

fixed: map overlay not aligned with shape when setting time
fixed: movie not looping back to beginning
fixed: shape resetting orientation when exiting movie

shapeViewer 2.5.9 - 13/08/2015 - "Perihelion"

fixed: distance not correct in camera class

shapeViewer 2.5.8 - 17/06/2015 - "Better slopes"

improved: slope calculations
added: display of slopes values in black and white
fixed: display of slopes values in legend

shapeViewer 2.5.7 - 08/05/2015 - "Improved Pointing"

fixed: rendering implementation introduced an offset in apparent pointing
added: simplified gravity model (2 central masses) to speed up calculation,
future versions will let you chose between different models

shapeViewer 2.5.6 - 04/03/2015 - "Angles and facets"

added: display of incident/emergent/phase angles with [F2, F3, F4]
added: export of angles as one RGB image (R=i, G=e, B=phase)
added: export area now saves all selected facets in a file

(continues on next page)

(continued from previous page)

added: such list of facets can be reloaded
improved: facet selection

shapeViewer 2.5.5 - 14/11/2014 - "We have landed !"

fixed: kernel keyword being ignored in config file
fixed: when exporting view, filename gets written on the image
added: display of incident, emergent, phase angles

shapeViewer 2.5.4 - 31/10/2014 - "Halloween public release"

Unified interactive display and movie mode
Both modes will now show the correct pointing and illumination
Unified rendering loop, merged all SPICE calls in one function
Interactive mode displays distance and camera Id when time is set
Added configuration file to change startup options
Debug console can now be activated from this config file
Added display of incident, emergent, and phase angles
Harmonized keyboard shortcuts

shapeViewer 2.5.3 - 28/07/2014 - "SHAP1"

Added graticule

shapeViewer 2.5.2 - 15/07/2014 - "RPMOD7"

Added new shape model
Change rotation rate with [and]
Fixed bug in rotation

shapeViewer 2.5.1 - 26/02/2014 - "MPS release"

Added graticules to maps
Harmonized keyboard shortcuts
3D view can now be shifted with the arrow keys

shapeViewer 2.5.0 - 23/09/2013 - "Sequences"

Modified movie mode. Possibility to save frames, by default in a directory called
↪ "screenshots",
automatically created.
Generates screenshots from a given sequence, provided as an external xml file.
Reference frames are now defined in SPICE meta-kernel.
Possibility to load a predefined sequence from an XML file. In that case screenshots
↪ are
automatically saved.

8.6 BRANCH 2.4 “COSSIM”

shapeViewer 2.4.1 - 20/12/2012

Added raytracing rendering to the photometric model selection menu.
This feature is very demanding for the CPU and can take a long time for a model with more than a few thousand facets.
The rendering uses only the LAMBERT photometric model for the raytracing, and does not calculate reflections.
It does, however, provide the right feel for illumination conditions.

shapeViewer 2.4.0 - 02/11/2012 - "COSSIM"

Added COSSIM (COma Structures SIMulator) to shapeViewer.
The user can now define active regions in an external XML file, and visualize the structures created by the activity in real time (movie mode).
The example XML file (*COSSIM.xml*) contains a description of the format.

8.7 BRANCH 2.3 “Movie Mode”

shapeViewer 2.3.0 - 09/07/2012

Added a "Movie mode" which animates a trajectory based on SPICE kernels.

8.8 BRANCH 2.2 “Photometric models”

shapeViewer 2.2.4 - 26/06/2012

Projection bug corrected in version 2.2.1 had not been propagated in subsequent versions...
Added application icon.

shapeViewer 2.2.3 - 25/06/2012 - "Sliders"

Rewrote definition of photometric models, parameters for Lunar-Lambert and Minnaert functions can be set interactively.

shapeViewer 2.2.2 - 22/06/2012 - "NAC & WAC"

Added more photometric functions
Added display of NAC and WAC FoV.

shapeViewer 2.2.1 - 23/04/2012

(continues on next page)

(continued from previous page)

Corrected bug in equatorial projection

shapeViewer 2.2.0 - 12/04/2012 - "Photometry"

Added Lommel-Seeliger photometric model, and options to select different models

8.9 BRANCH 2.1 "SPICE"

shapeViewer 2.1.0 - 05/04/2012

SPICE library is now integrated in shapeViewer, no need for spiceData.exe anymore.
Code migrated to the CVS server.

8.10 BRANCH 2.0 "GUI"

shapeViewer 2.0.2 - 22/02/2012

Added distances and areas measurements.

shapeViewer 2.0.1 - 23/01/2012

Finished to implement extraction of topographic profile.

shapeViewer 2.0.0 - 29/09/2011

Rewriting of the interface. shapeViewer has now a GUI written with FLTK.

VERSION 1

shapeViewer 1.6 - 22/03/2011

implemented gravity model

shapeViewer 1.5 - 22/02/2011

performance optimization

shapeViewer 1.4 - 21/11/2010

added calculation of solar elevation,
selection of facets, areas measurements

shapeViewer 1.3 - 09/09/2010

added conversion to cylindrical maps

shapeViewer 1.2 - 20/08/2010

added SPICE support, and texture mapping

shapeViewer 1.1 - 20/07/2010

can read both ".shape" and ".ver" file formats

shapeViewer 1.0 - 07/07/2010

1st version.