Caviar

Installation Instructions & User Guide

Version r1.0 June 2017

Queen Mary University of London

l'Observatoire de Paris

Institut de Mécanique Céleste et de Calcul des Éphémérides

Contents

1 Introduction

Caviar is an IDL-based software package for the display and astrometric reduction of images taken by spacecraft. The current version supports images recorded using the Cassini Imaging Science Subsystem (ISS). The package includes options to estimate and correct errors in the camera pointing direction, and for measuring the astrometric positions of natural satellites. The NAIF SPICE package (Acton, 1996) is used internally to access spacecraft, planet and satellite ephemerides, as well as camera pointing information and satellite limb models. The Vizier cdsclient package is used internally to access reference star catalogues via the Vizier web interface.

Caviar has two main functions:

- Correcting possible errors in the camera pointing direction
- Measuring the astrometric position of natural satellites

The original version of Caviar was developed at Queen Mary University of London (QMUL) by M.W. Evans & K. Beurle under the direction of C.D. Murray in 2004. The current version, developed by Louis-Etienne Meunier at the Institut de Mécanique Céleste et de Calcul des Éphémérides (IMCCE), under the direction of Valery Lainey (IMCCE), Nick Cooper (QMUL/IMCCE) and William Thuillot (IMCCE), represents an extensive re-structuring of the software, including the addition of a graphical-user interface (GUI) and additional options for astrometric measurement.

The software has so far been tested using IDL version 8.6 under MAC OS Sierra version 10.12.4 and Linux.

Acknowledgements:

Work undertaken at QMUL was supported by the UK Science and Technology Facilities Council while work at IMCCE was funded through the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement 263466 for the FP7-ESPaCE project. In addition to those mentioned in the introduction, the work has benefited greatly from discussions and suggestions from Zhang Qing-Feng, Kevin Baillie and Alain Vienne.

2 References and Further Reading

Acton, C. H. 1996. Ancillary data services of NASA's navigation and ancillary information facility, Planet. Sp. Science, 44, 65-70.

Canny, J., 1986, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698.

Cooper, N.J., V. Lainey, Meunier,L-E., Murray, C.D., Zhang, Q-F, Baillie, K., Evans, M.W., Thuillot, W., Vienne, A., 2017, The Caviar Software Package for the Astrometric reduction of Cassini images: description and examples, Astron. & Astroph, In preparation

Owen, W.M. Jr., 2003, Cassini ISS Geometric Calibration of April 2003, JPL IOM, 312.E-2003.

Stetson, P.B., 1987, DAOPHOT - A composite program for crowded-field stellar photometry, Pub. Astron. Soc. of the Pacific, 191-222.

3 Installation

3.1 Obtaining Caviar

The Caviar code can be obtained via the NASA PDS website at:

https://pds-rings.seti.org/cassini/iss/software.html

3.2 Requirements

- A PC running Linux (kernel 2.4.xx or greater) or a Mac running OS X with X11 (XQuartz) support.
- A license for IDL version 8.2 or greater. Currently a full IDL license is required as Caviar does not run using the free IDL Virtual Machine. Both 32 and 64 bit architectures are supported.
- The NAIF ICY interface for the NAIF SPICE package, compiled for the appropriate architecture and the installed version of IDL. At the time of writing, the latest version is N0064. This is available from:

  http://naif.jpl.nasa.gov:/pub/naif/toolkit/IDL/

- The Vizier 'cdsclient' package (see 3.4 for more details). At the time of writing, the latest version is cdsclient-3.84. This is available from:

  http://cdsarc.u-strasbg.fr/ftp/pub/sw/cdsclient.tar.gz

- An active internet connection

Note: Since Caviar is compiled at runtime, Caviar is itself architecture-independent.

3.3 Cassini Images

Images taken using the Imaging Science Subsystem of the Cassini Spacecraft may be downloaded from the NASA PDS Imaging Atlas:

https://pds-imaging.jpl.nasa.gov/search/

3.4 SPICE Kernels

SPICE kernels are used to retrieve camera information as well as planetary, satellite and spacecraft trajectories, and also to perform various data transformations.

For convenience, a full set of Cassini SPICE kernels valid from the year 2000 up to March 2015 is bundled with the Caviar source code. Please be aware that although the source code requires only 2.9 Megabytes of space, the SPICE kernels require 6.7 Gigabytes. The kernels may be found in the subdirectory /kernels in the main source directory, and are listed in the file meta-kernel file, kernels.ker. Additional kernels may simply be added to the /kernels subdirectory and their names appended to the kernels.ker file, with the most recent kernels positioned last.

Additional kernels may be found at:

http://naif.jpl.nasa.gov/naif/data.html

Please consult the NAIF website for further information:

http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/kernel.html

3.5 Star catalogues

Caviar uses the Vizier web-interface to obtain reference star positions. Currently the Tycho2 and UCAC5 catalogues are used. The UCAC5 catalogue has been chosen in preference to GAIA-DR1 because of the lack of proper motion information in GAIA-DR1. Later versions of Caviar will take advantage of the better coverage in future releases of GAIA, in addition to the GAIA proper motions.

In order to access the star catalogues, an active internet connection is needed and the Vizier ‘cdsclient’ package must be installed. To check if this package is already installed on your system, type the following command in a terminal window (i.e. not in IDL):

```
vizquery -mime=text -source=tycho2 -out.form=mini -c=83.55-20.17 -c.rd=0.497 f.mag="<13.0"
```

This should return a list of TYCHO2 stars. If the vizquery command is not found, the cdsclient software will need to be downloaded and installed. It can be obtained from:

http://cdsarc.u-strasbg.fr/ftp/pub/sw/cdsclient.tar.gz

(Notes: if installing cdsclient under MAC OS X, the Xcode command-line tools must be installed, otherwise the installation will fail with stdio.h not found etc. If the message ‘command wget not found’ occurs, the ‘wget’ routine must be downloaded. This is available freely from various locations on the web).

If the vizquery command is found, but the command hangs, and no stars are returned, then the cdsclient software is already installed on your system, but either you have no active internet connection, or the Vizier web-server is down. Either connect to the internet or await the return of the Vizier web-server.

3.6 Demonstration Images

A selection of Cassini images has been included with the source code, for demonstration purposes and for testing the installation. They can be found in the subdirectory demo_images in the main source directory. These include images from the Jupiter flyby as well as from the prime Saturn mission.

3.7 Caviar Configuration

1. Choose a suitable location for the Caviar software and unpack the tar file. For example:

```
cd /Users/nc/

tar -xzvf caviar_r1.0.tar.gz
```

2. Add the chosen location of the Caviar source directory to the IDL_PATH environment variable in the .bashrc or .cshrc file in your home directory, as shown below (create a file with this name if it does not already exist and add the text as shown). The syntax will depend on the type of shell being used.

For a C-shell (*.tcshrc or .cshrc* file):

```
setenv IDL_PATH '/Users/nc/caviar_r1.0:+/Applications/exelis/idl84'
```

For a Bourne shell (*.bashrc* file):

```
declare -x IDL_PATH='/Users/nc/caviar_r1.0:+/Applications/exelis/idl84'
```

3. Edit the file kernels.ker, located in the caviar source directory and add the chosen location of the Caviar source directory to the PATH_VALUES name. Note that /kernels/ must be appended to the path name. For example:

```
PATH_VALUES =( '/Users/nc/caviar_r1.0/kernels/' )
```

4. Edit the file caviar.pro, located in the caviar source directory, inserting the correct paths for the environment variables listed below.

Your chosen location of Caviar source directory. For example:

```
caviarDir = '/Users/nc/caviar_r1.0/'
```

The location of the SPICE Toolkit on your system (specifically the directory containing file icy.dlm). For example:

```
icyDir = '/Users/nc/spice/N0065_64/icy/lib/'
```

The default location of the Cassini images can also optionally be set here. Leaving the existing command unchanged will default to the location of the demo_images directory in the caviar source code directory (see 3.5). Alternatively, to set the location to the local working directory, use:

```
SETENV, 'CAVIAR_IMAGES_DIRECTORY=.'
```

No other installation steps should be necessary.

4 Launching Caviar

- Ensure that you have a live internet connection.
- Open a terminal window or X window and optionally go to the directory containing the images (alternatively the directory location may also be changed in the GUI) . At the command line prompt, type idl .
- At the IDL prompt, type @caviar .

Caviar routines will be compiled and the Graphical User Interface (GUI) will start ...

5 Loading Images

Image

After launching Caviar, or after clicking on "Image" from the menu "Load" in the Caviar main window, the window shown in Figure 1 will appear.

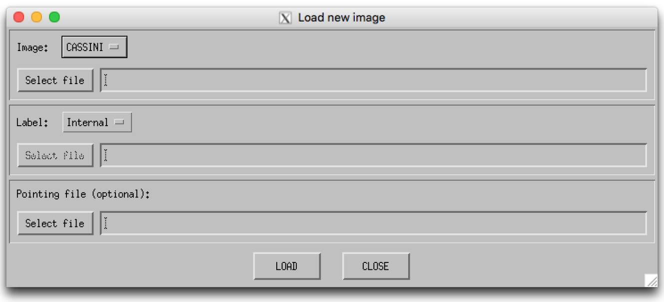

Figure 1: *"Load new image"* GUI

Select the image you want to open by clicking on "Select File". A drop down menu will allow you to locate the image. Cassini ISS images can be identified by the suffix .IMG. Image filenames starting with N denote narrow angle camera (NAC) images, while those starting with W are WAC images.

Label File

Optionally supply an image label file. In most situations, the label field can be left to default to “internal”, with the filename left blank. This will instruct Caviar to use the binary internal header attached to the image. Optionally an external label file (with suffix .LBL) can be used, if available. The information contained in the internal and external labels should be the same but the ability to supply an external (LBL) file is provided as an option. If loading the external (LBL) file, take care to load the correct file i.e. the one corresponding to the image to be loaded.

Pointing File

Optionally supply a Caviar pointing file from a previous Caviar session. If no Caviar pointing file from a previous session of Caviar is supplied here (see Section 8.5), Caviar will use the pointing information extracted from the C-kernels. Note that, if the raw, uncorrected C-kernels are being used, the image will be loaded without any correction to the nominal camera pointing.

Alternatively, if available, a pointing file generated by previous astrometric reduction by Caviar may be supplied, in QMPF (Queen Mary Pointing File) format. The software will then apply the pointing correction contained in the file when the image is loaded. This is useful when, for example, experimenting with satellite measurement options: the image can be loaded and options tested without having to re-estimate the pointing correction.

Having made your selections, press the “Load” button. The main Caviar GUI will now load.

A few seconds may elapse before the main GUI appears, while the software attempts to load the reference stars from the Vizier online database. An active internet connection is required otherwise the stars will not be loaded. If the internet connection is live, but the stars still do not load, the Vizier online server may be temporarily unavailable.

Note that for the first image to be loaded, Caviar will automatically load the reference stars and all satellites in the field-of-view, together with the primary planet, if visible (see section 7.1 for more information). However, when loading any subsequent image in the same Caviar session, the stars and satellites are not loaded automatically and must be loaded manually using the menus in the main GUI, described in the next section.

6 Caviar Main Interface

The main Caviar window is divided into two parts: The image drawing area on the left and the "lateral Pane" on the right containing the menus used to access each of the various tools available within the package (Figure 2).

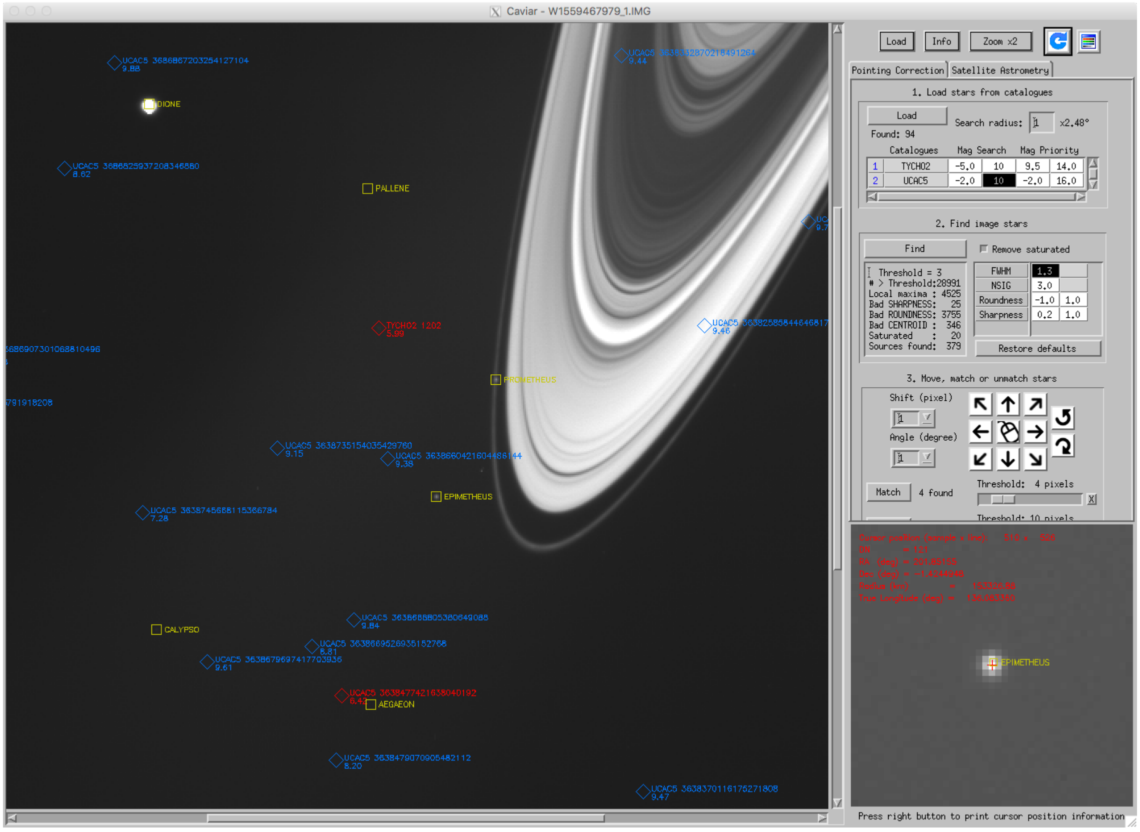

Figure 2: Caviar main interface

6.1 Image Drawing Area

The image drawing area displays the raw image, together with various objects such as stars, satellites or rings, superimposed as overlays. The positions of objects within an image are computed using information from SPICE kernels and star catalogues. Camera distortion effects are implicitly taken into account in the transformation from sky coordinates to image coordinates via an empirically-derived model for the Cassini ISS cameras Owen (2003).

The image drawing area automatically resizes if the user resizes the "Main window". Invisible parts can be accessed by clicking and dragging or with the sliders. The zoom factor can be changed with the mouse wheel or by using the zoom button towards the top right of the lateral pane. Also, moving the pointer over the image displays a secondary zoom window at the bottom of the *"lateral pane"* with the part of the image surrounding the cursor magnified by a factor of four. The zoom window also shows various information related to the position of the cursor, such as RA and DEC, and radius and true longitude relative to the planet's centre. This information can be printed to the console by pressing the right button. Finally, drawing options are available at the top-right of the *"lateral pane"*, including the REFRESH button. Note that the zoom window partially covers some of the buttons in the lower right of the lateral pane. To remove the zoom window, press the REFRESH button. To restore the zoom window, move the cursor back into the image area.

6.2 Lateral pane

The lateral pane gives access to each Caviar tool:

- *"Load"* menu... for loading images, satellites, rings, and stars from catalogues.
- *"Info"* menu... to display the image label in a pop-up window.
- *"Zoom"* menu... to change the zoom factor for the image drawing area.
- *"Refresh"* button... to re-draw the image and objects in the drawing area and to remove the Zoom window at the bottom right.
- *"Colour table & levels"* button... to display an interface allowing the colour table and levels to be changed or adjusted. The image drawing area is automatically updated after each modification.

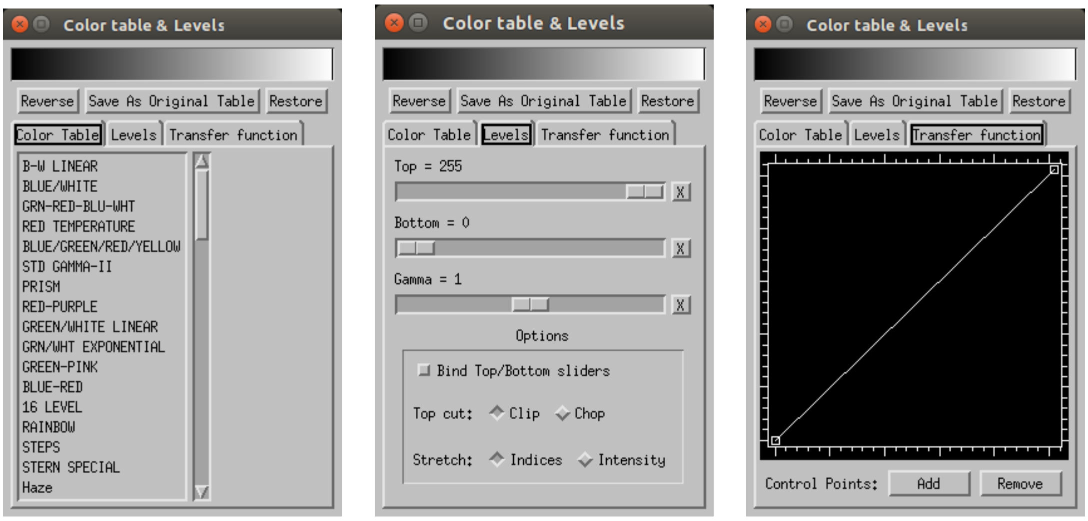

Figure 3: *"Colour table and levels"* GUI

- *"Image Pointing"* tab... tools to perform astrometric reduction (camera pointing correction). See section 8
- *"Satellite Astrometry"* tab... tools to compute the position of the centres-of-figure of satellites. See section 9

7 Loading Objects

When launching a new Caviar session, an overlay of the predicted positions of any satellites and stars within the field-of-view will automatically be loaded for the first image. However, for any subsequent images, these overlays must be loaded manually by pressing the "Load" button at the top of the lateral plane, then selecting the type of object to be loaded from the drop-down menu.

7.1 Satellites

Even if Caviar has already automatically pre-loaded all the satellites corresponding to the parent planet, they can also be loaded at any time manually by pressing the "satellites" button in the "Load" menu of the lateral pane of the Caviar main window, as described above. The window presented in figure 4 appears.

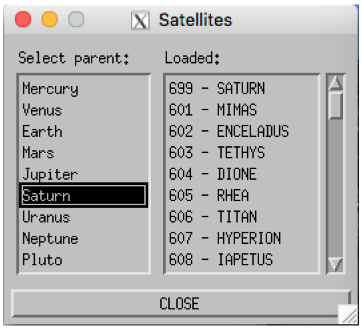

Figure 4: *"Load satellites"* GUI

The list on the left shows the primary body. On the right is a list of the satellites that will automatically be loaded for the selected parent planet, if they are in the field-of-view. Clicking on a primary body in the list on the left will remove previously loaded objects and load satellites for the selected parent planet.

The source routines for these options are grouped in the file: *'caviar_satellites_pkg.pro'*.

7.2 Rings

Caviar allows the creation and overlay of circular rings, as well as a Saturn F-ring model. Circular rings have a user-supplied radius in the equatorial plane of the parent object. The parent can be any loaded satellite or the parent planet. To launch the Rings GUI, go to "Load" menu in the Caviar main window and click on "Rings".

The "rings" GUI in the figure 5, shows a table listing selected circular rings to be overlaid on the image. Each ring is displayed relative to a parent represented by its identifier (PID). The available parent objects are listed on the right of the interface with the corresponding PID. Each ring is defined by a radius, the number of points to compute/display (3000 is the default), a start position and an end. These last two parameters allow a small arc to be displayed rather than an entire circle, thus speeding up the display. By default, the start and end parameters are 0° and 360°, respectively, and the entire circle will be computed.

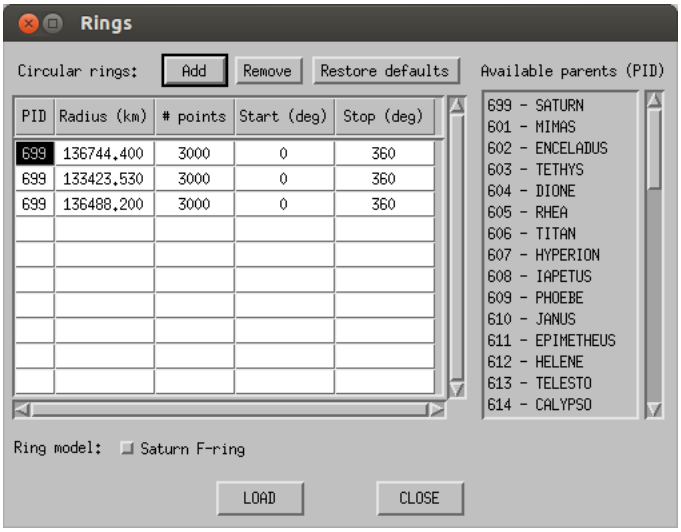

Figure 5: *"Load rings"* GUI

If the table is full, a new line may be added by pressing the "Add" button at the top of the GUI. A line/ring or multiple lines/rings may also be removed by selecting at least one cell on a line and then pressing the "Remove" button. By default, Caviar adds predefined rings for Jupiter and Saturn. If you change these settings and want to retrieve the defaults, just click on the 'Restore defaults' button.

An option to load a model of Saturn's F ring is also available (using fit 11 from Table 3 of Cooper et al. 2013). To display a ring overlay based on this model, check the box below the table. The model will only be visible if the F ring is within the field-of-view. To remove the F ring model, un-check the box and press the load button again.

Finally, click the "Load" button to display the rings defined in the table. The new rings are stored in the "rings" variable under the 'CAVIAR_DATA' common block (See 10 for the structure of this variable). At the end of the process, the Caviar image drawing area will update showing the new ring(s), if visible within the image field-of-view. When satisfied, click the “Close” button to close the rings’ GUI.

8 Image Pointing Correction Tools

Although the position of the spacecraft is known to a high degree of accuracy (100’s of metres), the attitude and therefore the pointing direction of the cameras, based on the star-trackers, is known only approximately. The approximate pointing information is encapsulated in SPICE C-kernels (kernels with suffix bc). The purpose of the image pointing tools in Caviar is to correct for possible errors in the raw C-kernel derived pointing direction. Errors in the raw C-kernel-derived pointing direction are generally of the order of 10’s of pixels, but in exceptional cases can be several hundred pixels. After correction, Caviar is typically able to correct the pointing to a precision of 0.1 pixel, depending on reference star coverage and star detectability. If using reconstructed (i.e. corrected) C-kernels, no pointing correction may be necessary. The image pointing tools are accessible via the Pointing Correction tab.
Note that Caviar does not output corrected C-kernels. Instead, corrected pointing information for a given image may optionally be output to a text file (by default, with suffix QMPF), in the form of the RA, DEC and TWIST of the camera pointing vector, and its equivalent quaternion. This “pointing file” (QMPF file) may optionally be read back in to Caviar in order to apply a previously derived pointing correction 8.4.

If no previously-derived pointing file is selected when the image is loaded, the software will obtain the approximate camera pointing direction from a SPICE C-kernel. Internally, the pointing information may be defined either in terms of a so-called C-matrix, or in the form of the RA, DEC and TWIST angle of the centre of the image, or equivalently as a quaternion (see the SPICE documentation for more details).

To correct the approximate pointing direction to sub-pixel precision, the overlayed positions of reference stars from a catalogue (see section 8.1) must first be manually moved to align them approximately with their actual positions in the image (see section 8.2) before performing a statistical match (see section 8.3). Following this manual alignment and matching, an iterative algorithm is used to fine-tune the match of the catalogue stars to their imaged positions, in order to obtain the final camera pointing direction (see section 8.4).

8.1 Loading Stars From Catalogues

To correct the error in the camera pointing direction, the positions of background reference stars in the image are compared with their expected positions based on the star catalogues. Currently, the software uses the Tycho2 and UCAC5 catalogues. These are obtained via a web interface, so an active internet correction is required.

Caviar has a graphical interface to manage the star catalogue extraction parameters. The interface is centered on a table that shows each available catalogue and allows the setting of various parameters:

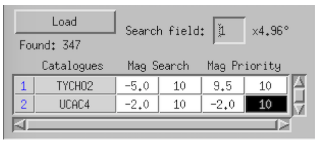

Figure 6: *“Load stars from catalogues”* GUI

When launching a new Caviar session, catalogue stars will automatically be loaded for the first image selected. For any subsequent images, the stars will not be loaded automatically and must be loaded from the gui, either via the “Image pointing” tab of the Caviar lateral pane, or from a pop-up window accessed by going to the “Load” menu and clicking the “Stars from catalogues” button.

Catalogue selection:

By default, all the catalogues are set as "selected" for star searching. Any catalogue can be unselected by clicking on the cell corresponding to its name or priority number. An unselected catalogue will be put at the end of the table and the corresponding cells will become non-editable with a grey background colour. Select that catalogue again and it will move to the bottom of the "selected" catalogues and the top of the "unselected" ones.

Catalogue priority:

As different catalogues may contain the same stars, duplicates must be removed. This is done according to the priority of each catalogue. This priority appears as a number on the first cell on the left and can be changed using the selection process.

Magnitude search range:

The two columns below "Mag search" correspond to the magnitude search range. The cells are editable and can be changed as required. Be aware that for "recent" catalogues and for images with a wide field of view ($> 1°$), defining too large a maximum value can lead to a very large number of stars being displayed, possibly far more than will be realistically useful in the correction process. This may also greatly slow down the drawing of the image display as well as the computation of the pointing correction itself.

Magnitude priority range:

The two columns on the right, below "Mag Priority", allow the definition of a magnitude range within which a given catalogue will have priority over other catalogues having a higher priority number. In the example in the above figure, the UCAC5 catalogue has priority over TYCHO2 for stars of magnitude within [-5 : 9.5] only. Note that for values for the catalogue with highest priority, this range has no meaning.

Search field dimension:

The search field radius (or box dimension, depending on the catalogue access routine) can be modified here. By default, the star-field radius corresponds the image field-of-view at the diagonals (i.e. the radius of the circumscribed circle for the image). For NAC images, this corresponds to 0.24735 degrees, and for WAC images, to 2.4735 degrees

To load the stars from the catalogues and automatically remove duplicates, click the *"Load"* button at the top of the GUI. The number of unique stars found will update in the image drawing area. The stars are represented by blue squares with, on the right, the star identification number taken from the catalogue database and the magnitude.

8.2 Find Stars In The Image

Like the "catalogue stars" module, this tool is also used as part of the camera pointing process. Its purpose is to search for objects in the image that may correspond to stars. The routines is base on *DAOPHOT FIND* (Stetson, 1987) and searches for local maxima in the image, convolved by a Gaussian kernel.

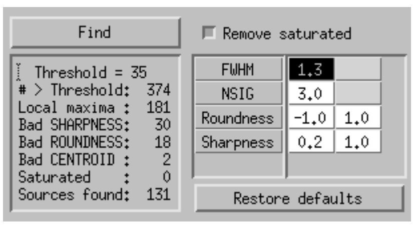

Figure 7: *"Find image stars"* GUI

The left part of the module's interface includes the *"Find"* button to execute the algorithm and a text box which will contain some statistics, such as the number of sources found and the number of those which have been discarded. On the right, some parameters can be modified:

Keep or remove saturated stars:

Saturated sources, with a maximum equal to 255 for 8-bit images can optionally be removed in order to prevent these from contaminating the results.

The full width at half maximum (FWHM) – in pixels:

This value varies according to the camera, but most spacecraft imaging cameras have broadly similar sampling characteristics, so the default value should work in most cases. The default value is 1.3 pixels and corresponds to the sampling of the Cassini ISS NAC when no binning is applied to the image. Binning consists of summing 4 pixels (2x2) or more to reduce noise, etc.. In this case, the FWHM value should be multiplied by a factor of 2.

The global maximum threshold (NSIG) – greater than unity:

After convolving the image with the Gaussian kernel, the algorithm computes the mean and the standard deviation (sigma) of the resulting image. This value is used as a threshold to extract maxima and future sources. The NSIG parameter allows the threshold level to be changed. Statistically, a value of 3 (the default) corresponds to a confidence of 99.7% in the quality of the maxima extraction ($NSIG = 2 \Rightarrow P = 95\%$; $NSIG = 1 \Rightarrow P = 68\%$). This means that, even if some good sources are removed from the process and bad ones remain, most bad sources will still be excluded. Generally, the detection of background noise comes with a value between 1 and 1.5. Avoid setting a value that results in background detection even if you find only one star that matches the catalogues. So, a good starting value for NSIG would be around 2.5 to 3, hopefully resulting in at least 3 or 4 stars that match the catalogues. After applying this threshold, the algorithm extracts local maxima by comparing values of neighbouring pixels. It then computes some statistics to reject bad sources.

The roundness parameter:

This parameter allows elongated sources like trails and other kinds of noise to be rejected. A perfectly circular source will have a roundness of 0 while an elongated source will have a greater value, positive or negative. By default, the range is between -1 and +1 and is quite conservative. Setting -0.3 and 0.3 will reject more sources.

The sharpness parameter:

This parameter may be used to reject sources based on their gradient. A null value corresponds to something flat while a value of one corresponds to an infinite gradient such as a pair of pixels with one equal to zero (black) and the other pure white. The default value is 0.2 and is, again, conservative. Generally, the rejection becomes stronger between 0.7 and 0.8, but be aware that there is a fine distinction between removing noise and removing relatively faint stars!

Finally, the default parameter settings can be restored at any time by clicking the "Restore defaults" button.

8.3 Match Stars

With stars loaded from catalogues and star-like sources found in the image, the next stage in the pointing-correction process is to match them. To do so, you will have to do a rough manual alignment of the star template, to superimpose the symbols for the stars from the catalogue(s) with the symbols for the sources found in the image, and then click on the "Match" button. The interface is available in the "Image pointing" tab and looks like the screen-shot below:

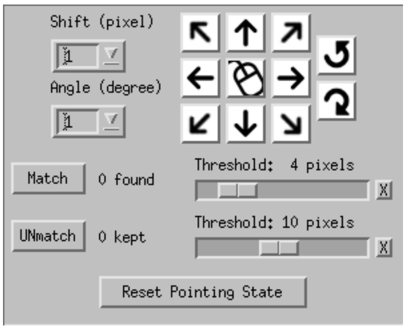

Figure 8: *"Move, match & un-match stars"* GUI

***Move*:** On the top, a block of buttons allows the catalogue star template to be moved manually according to the direction represented by the arrow icon and the distance (in pixels) set in the "Shift" field on the left. As the stars are "moved" the routine will modify the pointing matrix (C-matrix) of the camera in real time and update the positions of all the objects load in Caviar that have RA/dec coordinates. You can also rotate the field of view with the buttons on the right, according to the angle defined on the left. The icon at the centre allows the mouse to be used to change the pointing direction. After clicking on this button, move the pointer over the image drawing area. Then, hold the left button to shift the pointing direction or the right button to rotate. Leave the image drawing area to quit mouse mode. The *"Reset Pointing State"* button can be used to set the pointing direction back to the initial state.

***Match*:** Below, on the left, a button will execute the matching algorithm that computes the distance between predicted and imaged stars and matches them if their distance is lower than the threshold. By default this distance is 4 pixels. Its value may be changed with the slider. At the end of the computation, the number of matched stars will appear next to the "Match" button and the matched stars will be shown in red in the image overlay.

***Un-match*:** As for "Match", a button launches the un-matching mode. This mode is **manual**. Left-click with the mouse on any individual matched star to un-match it. Just leave the image drawing area to escape. The number of remaining matched stars appears on the right. Modify the selection distance threshold if needed.

8.4 Automatic Fine Image Pointing

This is the final stage in the estimation of the camera pointing correction.

Having now matched and approximately aligned the chosen catalogue stars with their imaged equivalents, in this step the camera pointing direction is adjusted iteratively to minimize the overall distance between the positions of the matched star-like sources and their counterparts from the catalogues.

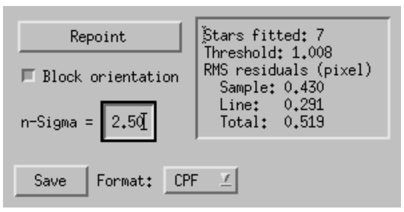

Figure 9: *"Automatic Fine Image Pointing"* GUI

The graphical interface consists of a button to launch the algorithm, a check box to indicate if the orientation of the camera (the TWIST angle) should be included as a free parameter, and a field to set the threshold factor for keeping (or not) stars with high residuals in the next iteration. For this last parameter, the default is 2.5 pixels. This value has been found to be the best compromise in most cases, but it can be changed if necessary. The text field on the right displays information about the results such as the number of stars actually fitted, the threshold values used for "sample", "line" and "total" residuals for each star, the final RMS residuals in sample and line directions and of the total, given by $\sqrt{sample^2 + line^2}$.

8.5 Saving Results

At the end of the computation, a window will ask if the pointing results are to be saved in a file. This is optional.

Caviar currently supports one output file format:

- QMPF (Queen Mary Pointing File). A format used as a standard since 2004 by the Cassini imaging group at Queen Mary University of London.

The file format is in plain text and the file contains the date & time, user-name, kernels used and relevant parameters, as well as measured positions and residuals for each fitted star or satellite, together with the camera pointing direction in the form of a quaternion. The saved "Pointing file" may be loaded back in to Caviar, when reloading the same image.

9 Satellite Astrometry Tools

Satellite astrometry tools are accessible via a tab from the main gui. These tools allow the position of the centre-of-figure of natural satellites within a given image to be estimated. Two methods are available. The choice of method will depend on whether or not the chosen satellite is resolved in the image, or not. If resolved, the limb-fitting option should be selected. If unresolved, the centroiding option should be used.

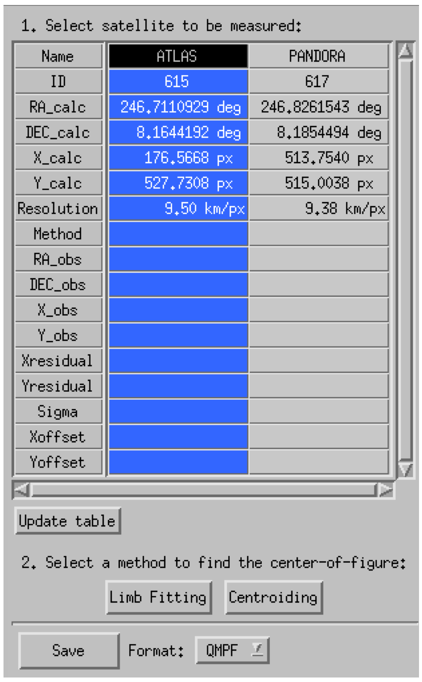

Figure 10: *"Satellite Astrometry"* GUI

A table (Figure 10) shows the satellites visible in the image, together with information about their position and approximate resolution in kilometres per pixel at the object's surface, based on the ephemeris extracted from the SPICE kernel. The column with blue highlighting corresponds to the satellite that is currently selected for detection. You can click on any cell in a column to change the selected object. The highlighting will change and a square symbol indicating the predicted satellite centre position based on the ephemeris extracted from the SPICE kernels will be shown on the image. A button allows the table contents to be updated after loading new satellites (it is not updated automatically).

Below are two options for the detection of the chosen satellite's centre of figure: **Centroiding** and **Limb Fitting**.

In addition, a button and a drop-down list are provided for saving the results and choosing the output format. Caviar supports *QMPF* and *CPF* file formats (see section 8.4). During the saving process the user will be also have the option to save measured satellite position results in a *CSV* file that can be read in a simple text editor or spreadsheet software.

9.1 Centroiding

This method is suitable for finding the centre-of-figure of unresolved satellites. It uses the same algorithm adopted for the *"find stars"* module. Based on DAOPHOT FIND, it searches for local maxima in a part of the image surrounding the satellite corresponding to twice its diameter, convolved by a Gaussian kernel. Currently there is no GUI to allow parameters to be changed, and the hardwired values are as follows:

- $FWHM = 2 * R_{satellite}$
- $NSIG = 1$
- $Roundness = [-1, 1]$
- $Sharpness = [0.2, 1]$

See section 8.2 for parameter definitions.

The raw satellite position generated by this method corresponds to the the centre-of-light of the object. If the spacecraft-object-Sun angle (the phase angle) is greater than zero, the measured position will not correspond to the desired physical centre of the object (the centre-of-figure). A phase-angle dependent correction is applied automatically to measured centre-of-light to arrive at the measured centre-of-figure. The centre-of-figure of the satellite in pixel coordinates is given by the $X/Y_{centroid}$ variable, while the phase correction (in pixels) is given by X and Y_{offset}.

9.2 Limb Fitting

This method is suitable for measuring the centre-of-figure of resolved satellites and involves three distinct steps:

- Detect the **limb edge** in the image
- Load a **shape model** for the satellite
- Fit the **shape model** to the **detected edge**

9.2.1 Limb detection

The first step is to detect the satellite limb edge in the image. The detected limb points are shown as red crosses overlaid on the image (Figure 11). Caviar provides two methods: **Gradient** and **Canny**. Select a method in the drop-list then click the *"Detect"* button to execute the algorithm with the parameters set below. The user can opt for the limb detection to be updated automatically each time a parameter has been changed, or only on pressing the *"Detect"* button. These parameter options can be displayed by clicking the + button. Some basic image processing tools to improve edge detection are also optionally available, in addition to parameters specific to each method. All are described below.

***Image pre-processing tools*:**

- **Band-pass filter:** Increase edge sharpness.
- **Morphological operator "close":** Close/fill craters with a mask of radius "r" pixels.
- **Interpolation:** Improve edge detection precision and compensate blurring of median filter by allowing fine tuning of the the effects of its parameter, "width".
- **Median filter:** Remove noise but blur edges.

***Limb detection methods*:**

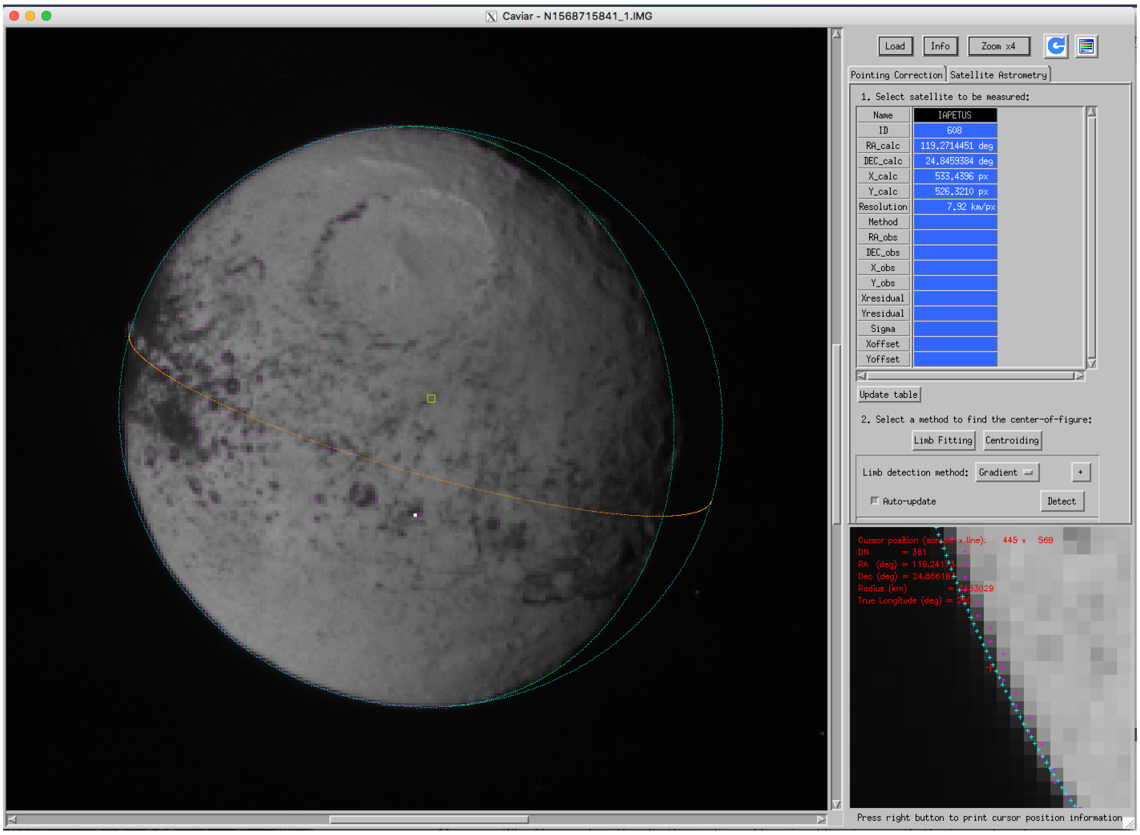

Figure 11: Limb Detection

***Gradient*:** This method computes the gradient of a sub-image including the satellite, along sample (x), line (y) and/or diagonal directions according to the user's choice and combines them to form the *gradient image*. Because the limb of a satellite is more or less semicircular it is generally recommended to compute the gradient in all three directions.

The algorithm then searches for the first and the second maximum in each row (i.e. sample direction) and/or column (i.e. line direction) of that image, with respect to the *activity threshold* and the *separation threshold*. Again, it is preferable to search for maxima in both directions, to avoid problems on the top/bottom edge or right/left edge of the satellite. The *activity threshold* defines the limit of the gradient magnitude below which no maxima will be found while *separation threshold* defines the minimal distance between the first and second maxima. It is important to compute the first two maxima because the terminator can induce a high gradient magnitude.

***Canny*:** The Canny algorithm firstly applies a Gaussian filter to smooth the image based on the *sigma* parameter. The gradient is then computed in sample (G_x) and line (G_y) directions using a pair of convolution masks, and the magnitude approximated with $\sqrt{(G_x)^2 + (G_y)^2}$. Edge direction is calculated using $\arctan(G_y/G_x)$. An edge point is defined to be a point whose gradient magnitude is locally maximum in the direction of the gradient.

Finally, hysteresis is applied to eliminate gaps and assure continuity of edges. A pixel not considered as a maximum but with a value greater than the *high threshold* is presumed to be an edge pixel and pixels connected to an edge pixel with a value greater than the *low threshold* are also included as edge pixels. *High* and *low* threshold values have a range of $[0, 1]$, corresponding to a factor of the histogram of the magnitude array.

This option uses the IDL Canny routine. More details may be obtained from the IDL documentation.

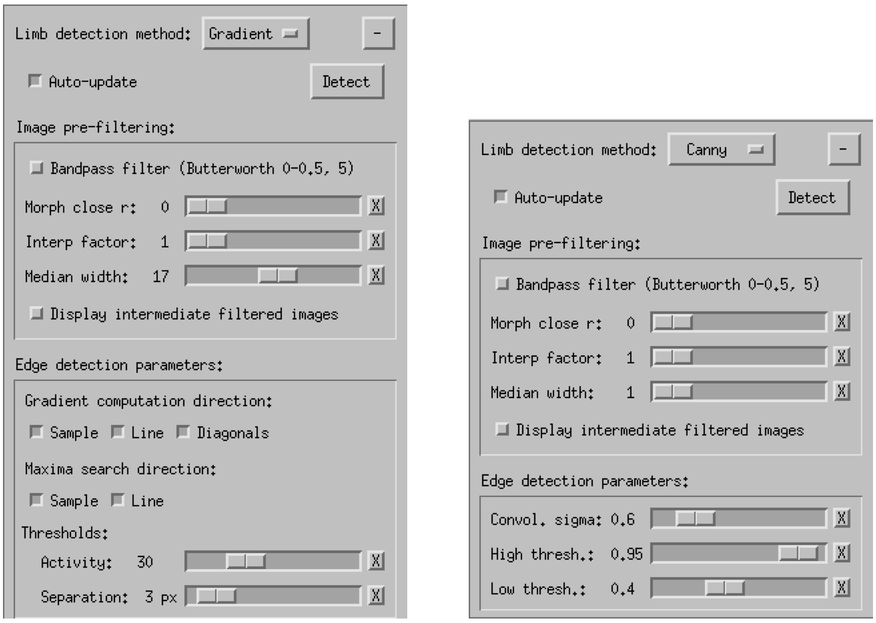

Figure 12: *"Limb detection"* GUI for both *Gradient* and *Canny* methods

9.2.2 Satellite shape model:

Having detected the limb, using either the gradient or Canny method, the next step is to load a shape model for the satellite. Currently, Caviar uses SPICE tool-kit routines to create a 3D ellipsoid based on the body radii obtained from the kernels. Click on the *Load* button to launch the routine. The algorithm will extract the contour of the satellite as seen from the spacecraft and project it onto the image drawing area with a cyan colour. It will also extract and display the terminator (in light green) and the equator (in orange). See Figure 11.

The graphical interface allows users to change the number of points used in the edge model and, by clicking on the *Move* button, to move the model by hand. This will be necessary before attempting the next step (fitting) if the offset between the model and the object is larger than the fitting threshold value (see below). In these circumstances fitting algorithm may not converge.

Figure 13: *"Satellite shape"* GUI

9.2.3 Limb fitting

Having detected the limb of the satellite in the image and loaded its shape model, the final step is to fit the model to the detected limb. The observed centre-of-figure of the object is then taken to be the centre of the fitted model. Caviar currently offers two algorithms for limb fitting: iterative offset (the recommended default) and least-squares.

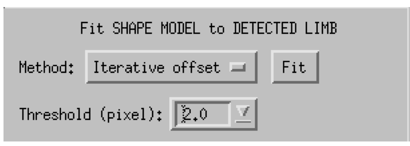

Figure 14: *"Fit SHAPE MODEL to DETECTED LIMB"* GUI

Of the two methods available, the iterative offset is recommended for most situations. Its advantage over the least-squares method is its robustness in the presence of noise, craters, or near the terminator. However, uncertainties may be correspondingly larger. Thus, it may be beneficial in particularly difficult cases to perform a second pass of fitting using the least-squares method (below), to reduce uncertainties further.

***Iterative offset*:** This algorithm computes the minimum distance between each detected edge point and the corresponding point on the model. Then, after removing edge points more than the "threshold" pixels away from the model (where "threshold" is a parameter), the mean distance between all edge points and model points is computed. The mean distance is then used as a correction, to update the model position, including its centre. The algorithm iterates until the offset becomes negligible ($< 10^{-6} pixel$). The measured centre-of-figure then corresponds to the final corrected position of the centre of the model.

***Least squares*:** This algorithm uses a standard least-squares technique to fit the centre coordinates of a triaxial ellipsoid model to the detected edge points. Fitted parameters are the centre coordinates and orientation angle of the ellipsoid.

The least-squares method can result in a greater precision, but is generally less robust than the iterative offset method. For this reason, the iterative offset method is the preferred option. As described above, it may be beneficial to perform the fitting in two passes, firstly using the more robust *"Iterative offset"* method, before using a second pass of fitting using the *"least squares"* method.

9.3 Saving Results

Finally, as for image pointing, the results of the astrometric measurement of the satellite's centre-of-figure may be saved, in QMPF file format:

- QMPF (Queen Mary Pointing File). A format used as a standard since 2004 by the Cassini imaging group at Queen Mary University of London.

This is a plain text format containing the date & time, user-name, kernels used and relevant parameters, as well as measured positions and residuals for each fitted star or satellite, together with the camera pointing direction in the form of a quaternion. This "Pointing file" may be loaded back in to Caviar, when reloading the same image.

In addition, a dialogue will ask if the astrometry results are to be saved in a CSV file, recognized by spreadsheet software, such as *Excel*. This file contains the following information (column format):

- (1) Image name | (2) Satellite name
- (3) Centre-of-figure calculation method (*"CENTROIDING"*, *"LIMB FITTING"*)
- (4) RA | (5) DEC coordinates of the satellite's centre-of-figure (degrees)
- (6) Sample | (7) Line coordinates of the satellite's centre-of-figure (pixel)
- (8) RMS residual (1-sigmal) in pixel
- (9) Sample | (10) Line offsets applied to correct for the phase angle in the *"CENTROIDING"* method (pixel)
- (11) RA | (12) DEC coordinates of the satellite's predicted position, based on the SPICE ephemeris (degrees)
- (13) Sample | (14) Line coordinates of the satellite's predicted position, based on the SPICE ephemeris (pixel)
- (15) Sample | (16) Line residuals (in pixel) between observed and predicted positions (O-C)

10 COMMON blocks and global variables

COMMON blocks:

```
- CAVIAR_GUI: wCAVIARtlb, wFITSATLIMBbase, wRINGSLOADbase, wCATSTARSLOADbase
- CAVIAR_DATA: image, catstars, imgstars, planets, rings
- CAVIAR_PARAMS: dispParams, imgDraw, dispProName, starsCatalogs, find_imgstars_params
- CAVIAR_SATLIMBFIT: selSatIndex, wSatTblID, wSaveLbl, WSFLbaseYsize
```

Global variables:

- `wCAVIARtlb`, `wFITSATLIMBbase`, `wRINGSLOADbase`, `wCATSTARSLOADbase` are widget base id's.
- `image` variable is of type *structure* and contains the following elements:

```
'PATH' = Full path of the image file.
'NAME' = Name of the image file with extension.
'WINDOW' = Caviar GUI window containing the image, initialized to -1.
'HEADER' = 1 or 2-Column(s) array containing the header of the image.
'HDRFRMT' = Scalar string representing the header format.
            Either 'INTVIC', 'EXTVIC'
'RAWIMG' = 2-Dimensional float array containing the image.
'BYTEIMG' = 2-Dimensional byte array containing the byte scaled image.
'NS', 'NL' = Scalar long integer. Number of samples/lines of the image.
'ET': Scalar double. Image mid-time in ephemeris seconds past J2000.
'EXPOSURE' = Exposure time in seconds.
'BINNING' = Scalar integer. Image binning.
'TARGET' = Structure variable with following elements:
    'NAME' = Scalar string. Image target name.
    'ID' = Scalar long integer. Image target ID.
'CAM' = Structure variable with following elements:
    'LNAME' = Camera long name with spacecraft + camera short name.
    'ID' = Camera NAIF integer code.
'SPC' = Structure variable with following tags:
    'NAME' = Spacecraft name.
    'ID' = Spacecraft NAIF integer code.
'METAKERNEL' = Scalar string. Spacecraft's SPICE meta-kernel file full path.
'FOVPIX' = Scalar double. Field-of-view of one pixel of the camera (radians).
'FOVIMG' = Scalar double. Field-of-view of the image along the diagonals (degrees).
'FOCAL' = Scalar double. Focal length of the camera.
'CENTER' = 2-Elements array of double. Bore-sight of the camera in pixel.
'KMAT': 3 by 2 array of doubles. Camera distortion coefficient matrix.
'EPSILON': 6-Elements array of doubles. Electromagnetic and optical
           distortion parameters vector.
'CMAT', 'CMAT_INI', 'CMAT_SAVED': 3 by 3 array of doubles. Rotation matrix
     corresponding to a specified unit quaternion (C-matrix).
'VOBS_STARS': Spacecraft velocity relative to the stars.
```

- `catstars` variable is a *list* of *structures* containing the following elements:

```
'NAME' = Name of star (type string).
'MAG' = Apparent magnitude (type double).
'RA' = Right-Ascension coordinate from the catalogue (type double).
'DEC' = Declination coordinate from the catalogue (type double).
'XCOORD' = Sample coordinate in the image (type double).
'YCOORD' = Line coordinate in the image (type double).
'MATCHED' = Flag indicating that the star has been matched and can
             be used to fit the camera pointing direction.
'FITTED' = Flag indicating that the star has been used to fit the
```

```
camera pointing direction.
```

- `imgstars` variable is a *list* of *structures* containing the following elements:

```
'XCENT' = Sample coordinate in the image (type double).
'YCENT' = Line coordinate in the image (type double).
'FLUX' = Flux of the star as identified by the Gaussian fit.
'SHARPNESS' = Sharpness statistic of the star.
'ROUNDNESS' = Roundness statistic of the star.
```

- `planets` variable is a *list* of *structures* containing the following elements:

```
'NAME' = Name of the satellite (or the main planet if it is the first element).
'ID' = UAI identification number of the body.
'RADII' = Radii of the body (km) in x,y,z directions given by spice kernel.
          Type: 3-Elements array of double
'POLERA' = Right-Ascension of the direction of the body's pole axis (type double).
'POLEDEC' = Declination of the direction of the body's pole axis (type double).
'XCOORD' = Sample coordinate in the image (type double).
'YCOORD' = Line coordinate in the image (type double).
'PIXSIZE' = Resolution at target i.e. size of a pixel on the body surface (type double -
'VISIBLE' = Flag indicating that the body must be in the image frame.
```

- `rings` variable is a *list* of *structures* that contains the following elements:

```
'PLANETID' = UAI identification number of the parent body.
'RADIUS' = Radius of the ring (km). Type: Scalar double
'NPOINTS' = Number of points computed for the ring as set by the user.
'STARTLON' = Longitude of the beginning of the ring as set by the user.
'STOPLON' = Longitude of the end of the ring as set by the user.
'RA' = Right-Ascension coordinates of the ring points.
       Type: n-Elements array of double.
'DEC' = Declination coordinates of the ring points.
        Type: n-Elements array of double.
'XCOORD' = Sample coordinates of the ring points in the image.
           Type: n-Elements array of double.
'YCOORD' = Line coordinates of the ring points in the image.
           Type: n-Elements array of double.
```