PlanetarySystemStacker for High-Resolution Imaging of Planetary System Objects through Turbulent Air

User Guide (Version 0.7.0, February 2020)

Rolf Hempel

Table of Contents

1. Introduction

PlanetarySystemStacker (PSS) processes a large number of video frames or still images of a planetary system object, taken in rapid succession, into a high-resolution image. It selects the sharpest sections of the best frames, removes the image warping caused by atmospheric turbulence, and combines the resulting snippets into a single image with maximum detail. In a second step, PSS offers the option to remove image blur by applying a multi-level sharpening filter.

With the introduction of digital image technology the photography of planetary system objects (the moon, sun and planets) has taken a great leap forward. The so-called "Lucky Imaging" technique today

allows amateur astronomers to record surface details which in the old days of analog photography were out of reach even to professional astronomers with access to the best earth-bound telescopes. For high-resolution work today the camera of choice is a video module which connects to the USB interface (preferably 3.0 or higher, allowing high data transfer speed) of a portable computer. With such a camera one can take many images in a short time and store them without compression artefacts in a RAW image sequence, using typically a SER or AVI video file container. On the negative side, however, the image sensors are quite small and exhibit relatively low pixel counts.

As an alternative, digital cameras (such as a DSLR) can be used to capture many images of the object in rapid succession. These images are stored as single files with identical pixel counts in a folder. Input to PSS can be either a video file or a directory containing still images of the same scene.

PlanetarySystemStacker is the first open-source software of its kind. The complete source code is published on Github, together with a detailed documentation of the mathematical algorithms used.

1.1 Project Authors and Contributors

In 2018 Rolf Hempel initiated the project. He developed the underlying algorithms and wrote the complete source code up to the first PSS release in May 2019. From the start it was his intention, however, to make PSS a community project, so the help by co-authors and contributors is encouraged. Beginning with Version 0.6.0, the following developers are contributing to the project:

Co-Authors:

Michal Powalko:

- Responsible for class "VideoReader" in module "frames.py". The goal is to provide support for all variants of ".avi", ".ser" (8 and 16bit) and ".mov" video formats.
- Handling of image I/O using .fits format.
- Python code style improvements and performance optimizations throughout the source code.

Contributors:

Jens Scheidtmann:

- Proposal of a different algorithm to perform frame stabilization in planetary mode (implemented in version 0.6.0).

John Duchek:

- Instructions for installation on Linux other than Ubuntu 16.04 LTS (Appendix B: Linux Installation Instructions)

2. Changelog

2.1 Changes in version 0.7.0 (February 2019)

- New "multi-level correlation" algorithm for image stabilization and de-warping.
- New algorithm for blending the stacked AP patches and background into a single image.

- Bug in handling large ".ser" files fixed.
- Manual selection of debayering pattern added.
- Support for video formats ".mp4" and ".mov"and image format ".png" added.
- User notification of job aborting via a pop-up window added.

2.2 Changes in version 0.6.0 (August 2019)

- Version for Linux added.
- Support for dark and flat frame calibration added.
- The required RAM of a job is estimated. If it exceeds the available RAM, an error is issued and PSS continues with the next job.
- Support for video format ".ser" and image format ".fits" added.
- Improvement of de-warping algorithm for videos with varying image brightness.
- New implementation of frame stabilization for "planetary" mode.
- Internal refactoring, most important: video input is encapsulated in class VideoReader.
- Several bug fixes.

2.3 Initial release 0.5.0 (May 2019)

- The Windows installer was built on a Windows 10 system. The software and the installation process were tested on Windows 10 only.

3. System Requirements and Software Installation

PlanetarySystemStacker in principle runs on any computer where a Python 3.5 environment and the required Python libraries are available. Starting with Version 0.6.0, automatic installers are available for both Windows and Linux. They can be used to install the software without installing Python first.

3.1 Windows (7 / 8 / 10)

The whole software was tested on an Acer laptop computer (type "Acer Aspire V5-573G", Intel Core i5-4200U, 12 GBytes RAM), and on a PC (Intel Core i7-7700K, 64GBytes RAM), both running Windows 10 Professional, version 1903. Video stacking very much benefits from large RAM. A minimum of 16Gbytes of RAM is recommended. For large video files processed on a system with less RAM it is unlikely that all data can be held in memory, so re-reading and re-computing data will slow down the execution.

The PlanetarySystemStacker software is distributed as a single file: the Windows installer "PlanetarySystemStacker_V0.7.0_Windows-Installer.exe". Before installation, any earlier PlanetarySystemStacker version should be de-installed first using the "Uninstall" entry in the program menu. When the installer is started, a wizard guides the user through the installation process. Apart from the program start entries, PlanetarySystemStacker does not write any parameters into the Windows Registry and can be installed at any file system location. An uninstaller is provided with the software. It removes all installed files. The following temporary files which are created at execution time in the user's home directory are not removed:

- .PlanetarySystemStacker.ini (configuration during the latest PlanetarySystemStacker run)
- PlanetarySystemStacker.log (logfile)

Those files can be deleted manually at any time. Please note that all configuration parameters are lost if the .PlanetarySystemStacker.ini file is deleted.

The Windows installation wizard offers to place a program starter on the desktop. It is decorated with the PlanetarySystemStacker icon automatically, which can be found under the filename "PlanetarySystemStacker.ico" in the installation folder.

3.2 Linux

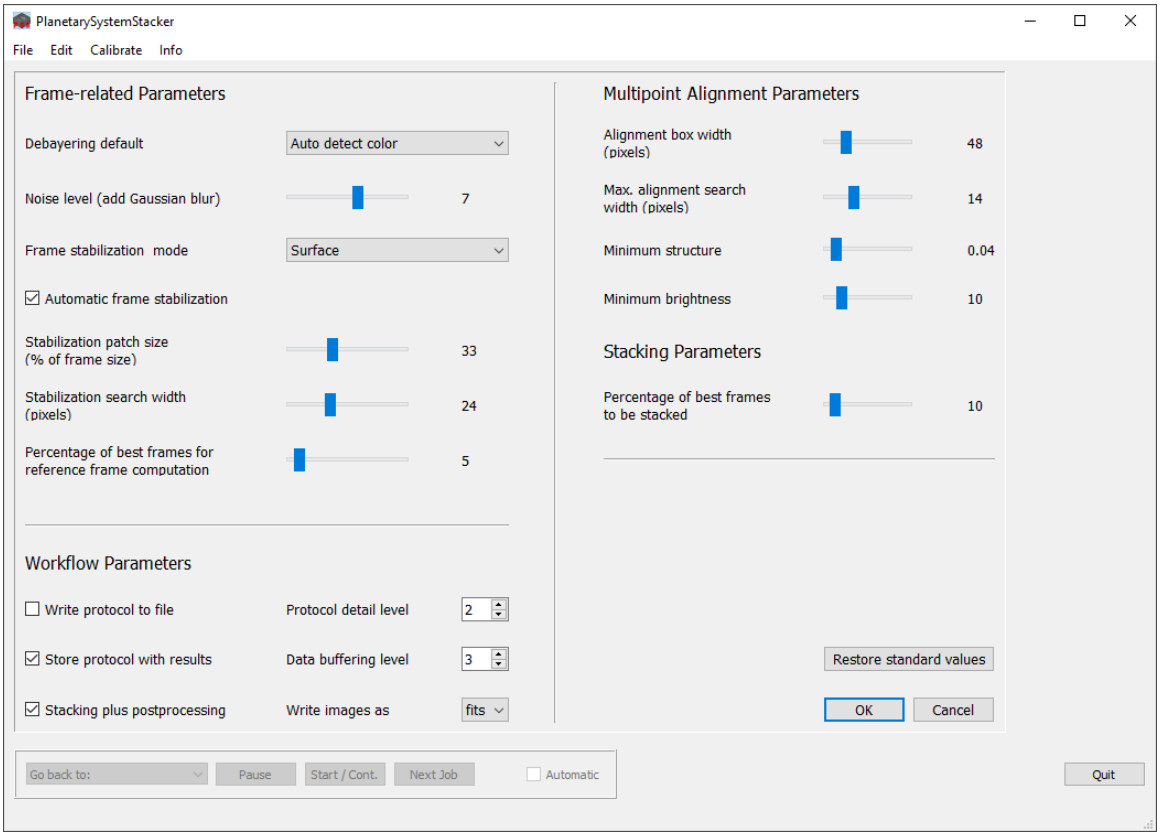
The large variety of Linux distributions makes it difficult to provide a Linux installer that works for all of them. For Ubuntu 16.04 LTS the complete PlanetarySystemStacker software can be obtained from Github as a gzipped tarfile ("PlanetarySystemStacker_V0.7.0_Linux.tgz"). The tarfile can be unpacked at any file system location. This will create the directory "PlanetarySystemStacker" which contains the executable "PlanetarySystemStacker", the user guide "PlanetarySystemStacker _User-Guide.pdf", and the Icon file "PSS-Icon-64.jpg".

For other Linux distributions (or Ubuntu versions), detailed descriptions of how to install the required Python libraries and to run the PSS code can be found in Appendix B: Linux Installation Instructions

4. Program Execution

PlanetarySystemStacker communicates with the user via a graphical user interface (GUI). Good usability was a high-priority design criterion. In particular, at any time the GUI presents to the user only the information which is of current relevance. The user interface was developed using the QT5 toolkit.

4.1 Program Start / Setting Parameters



When the program is started for the first time, a view opens automatically for the input of individual parameters. The parameters are arranged in groups relating to frame stabilization, multi-point alignment, stacking and workflow control.

Predefined parameter values should work okay in many cases, but the user is encouraged to experiment with different settings for optimization. If the mouse pointer hovers over a parameter name, a tooltip appears describing its meaning. A detailed explanation of all parameters can be found in "Appendix A: Configuration Parameters". The pre-defined parameter values can be restored at any time by pressing the button "Restore standard values".

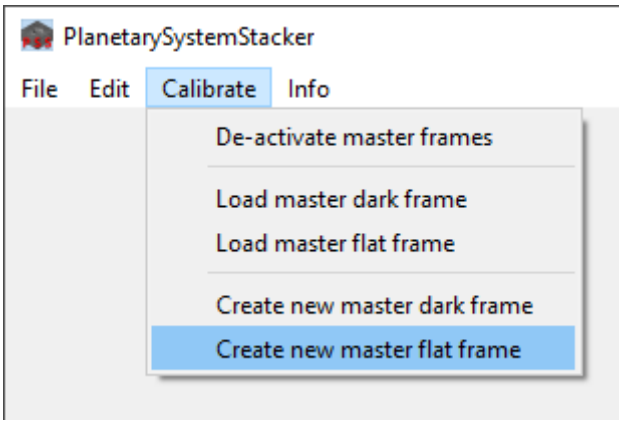
When leaving the parameter dialog, and at program termination, PSS writes the configuration file "PlanetarySystemStacker.ini" into the user's home directory. At later program runs the parameters are loaded from this file silently, the input dialog does not open, and PSS is ready for job input. Parameters can still be changed via the "Edit / Edit configuration" menu.

In addition to the standard configuration file in the home directory, the current parameter configuration can be saved to any file system location via the "File / Save configuration" menu entry. Later on, this configuration can be loaded via "File / Load configuration" to replace the current parameter configuration. Please note that the configuration not only comprises the parameters shown at the dialog above, but also all active postprocessing variants (see Section 4.12).

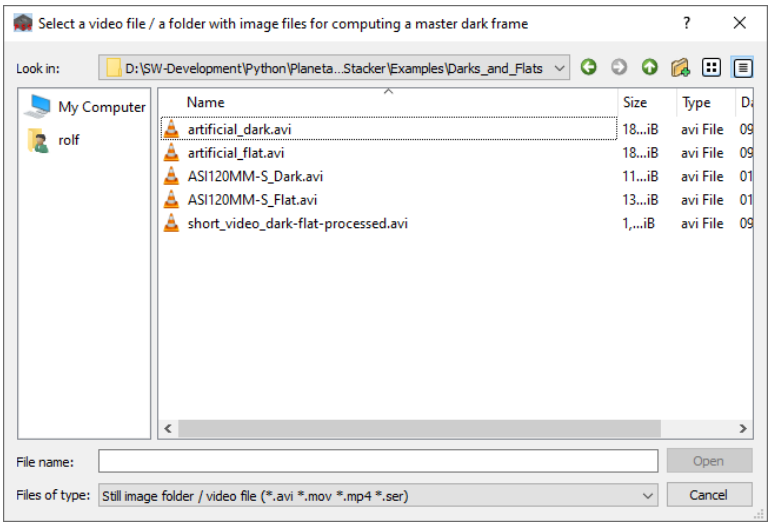
The menu entry "Info / About PSS" gives an overview of the computer environment and the versions of the Python libraries being used.

4.2 Dark / Flat Frame Calibration

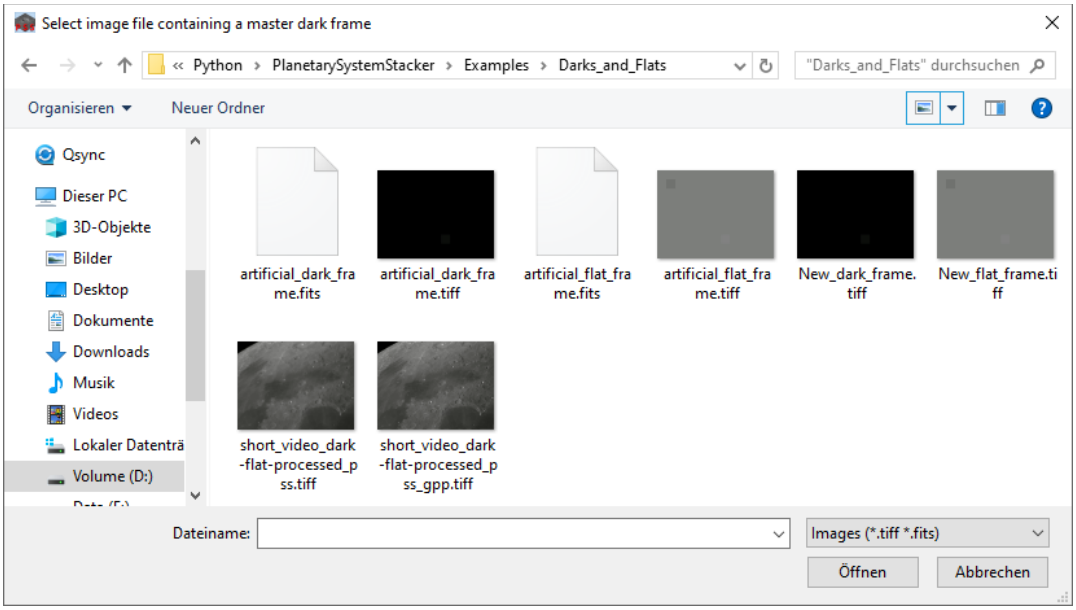
PSS supports the calibration of input frames using master dark and flat frames. This calibration is based on video files or image collections taken with the same camera as used for the stacking input.



First the dark / flat input videos (or directories with still images) have to be processed into so-called "master frames". To do so, select the menu entry "Create new master dark frame" or "Create new master flat frame" from the "Calibrate" menu. When the file chooser opens, select the input video or directory.



When the processing finishes, another file chooser opens to select the location where the master frame is to be stored as a 16bit .tiff (or .fits) file. If you do not want to store the master frame, just press "cancel". In this case the master frame is still kept in memory for image calibration during this PSS session.



Calibration in PSS can be done using either dark frame or flat frame correction, or both. In the latter case, make sure to create the master dark frame first. This way it is used already when the master flat frame is processed.

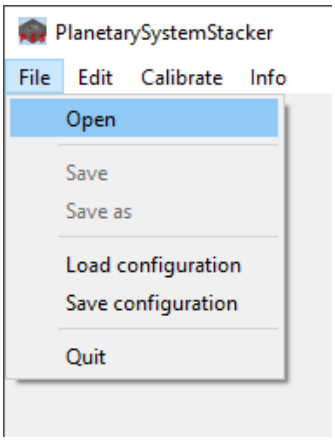
When master dark or flat frames have been stored on disk, they can be reloaded at a later PSS session using the "Load master dark frame" or "Load master flat frame" menu entries, respectively. It is, therefore, not necessary to re-process them from the input videos.

Both master frames and the input material used for stacking must match in terms of pixel dimensions and color type (color vs. monochrome). If, for example, a master flat frame is loaded after a non-matching master dark frame, the dark frame is de-activated automatically and only the flat frame stays active.

If matching master dark / flat frames have been created or loaded before stacking jobs are executed (see below), they are used to calibrate the input frames. If the active master frames do not match the stacking input, however, they are ignored.

4.3 Job Specification

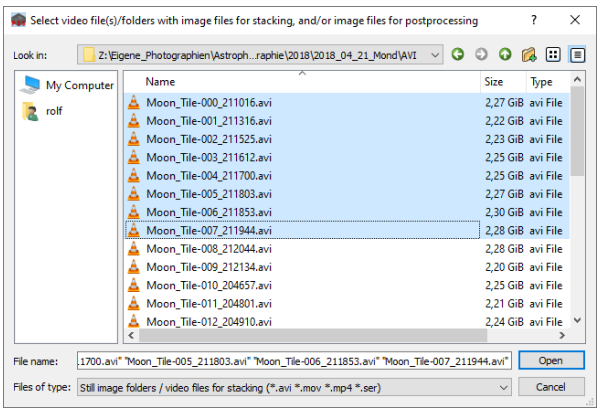
Throughout the program, user instructions are displayed in red print in the status line at the bottom of the main GUI, or within a dialog window. When the parameters are set, the user is asked to open the "File / Open" dialog, and to define the jobs to be processed.

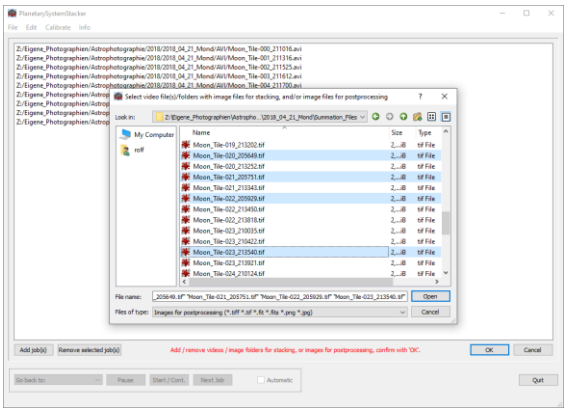


Upon choosing "File / Open", a window opens where jobs can be specified. At this point it is important to understand how PSS processes jobs.

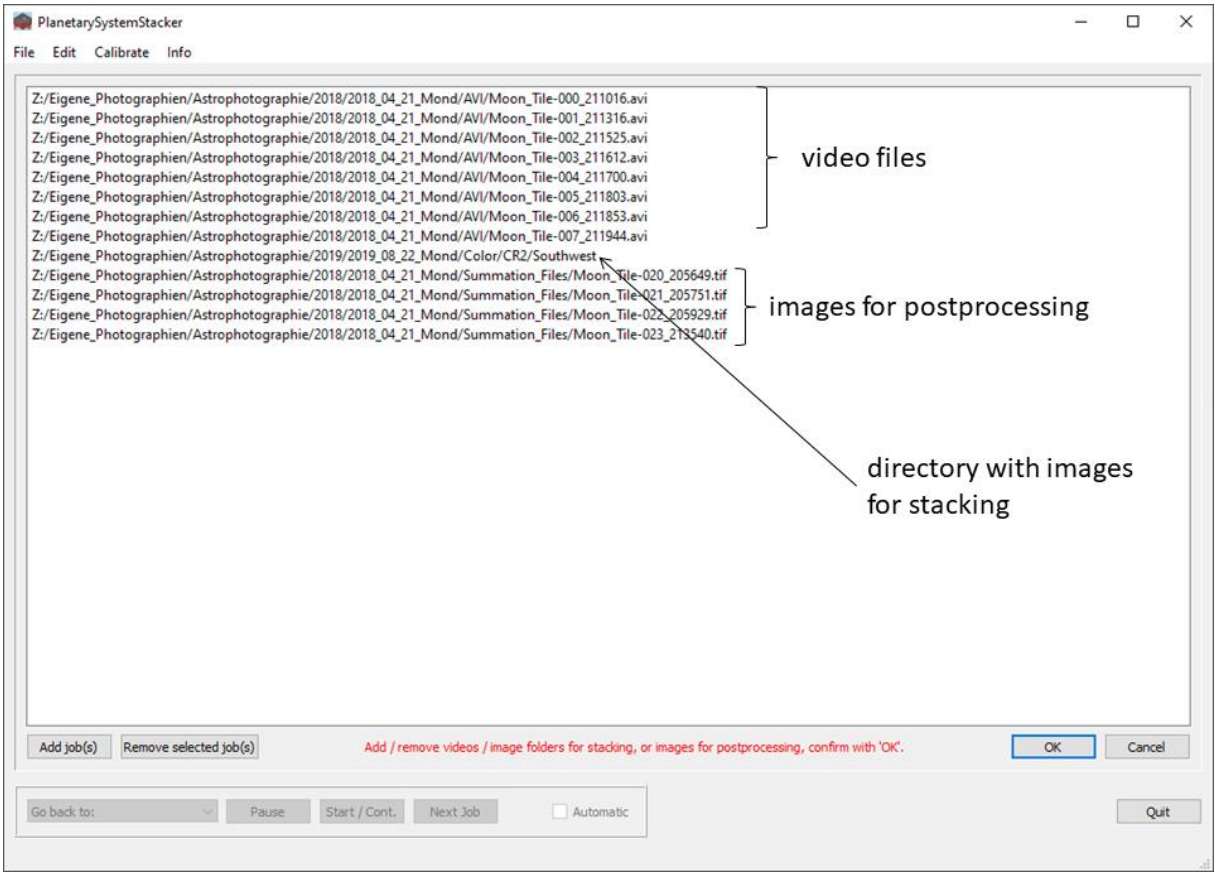
Basically, the user specifies a list of jobs which PSS then processes one after the other. There are two different job types which PSS recognizes by the kind of input specified:

- To define a stacking job, the user selects a video file (extension ".avi", ".mov", ".mp4"or ".ser") or a directory containing still image files with identical pixel dimensions. At the bottom of the file dialog window "Files of type" must be set to "Still image folders / video files for stacking (*.avi *.mov *.mp4 *.ser)". Selecting a directory with image files which do not match will result in an error message later on.
- To define a postprocessing job, the user selects a single image file (with the extension .tiff, .tif, .fit, .fits, .png or .jpg). At the bottom of the file dialog window "Files of type" must be set to "Images for postprocessing (*.tiff *.tif *.fit *.fits *.png *.jpg)"

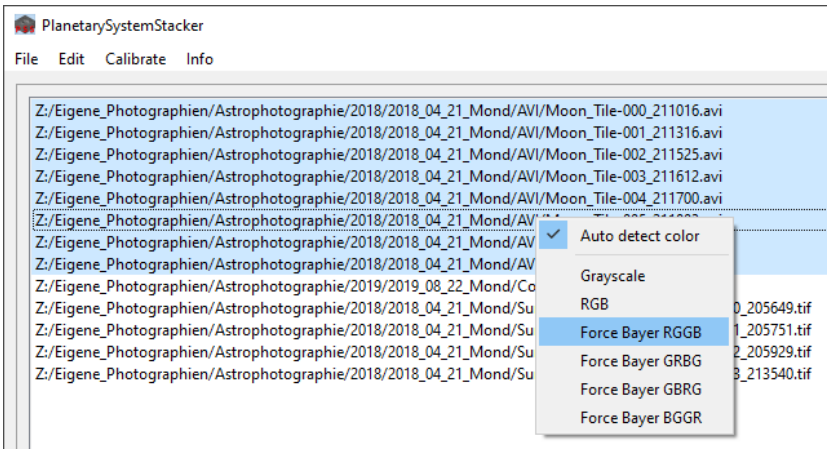




When the user presses "Open" to close the dialog window, the selected jobs are added to the list of jobs presented in the main window. This list may contain stacking and postprocessing jobs in any order. Additional jobs can be defined by pressing "Add job(s)" to re-open the job selection dialog again. Jobs can be removed from the list by clicking on them and pressing "Remove selected job(s)".



For color videos, debayering by default is done automatically. There is, however, the option to specify the Bayer pattern manually. To this end, the user selects a job (or a range / set of jobs) in the job list, and presses the right mouse key to open the context menu:

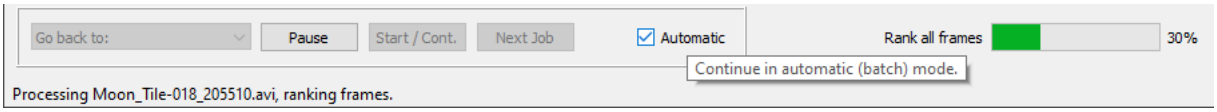


The Bayer pattern currently registered for the job is checked. It can be changed to any other pattern offered by the context menu.

When all jobs are defined, pressing "OK" completes the job specification.

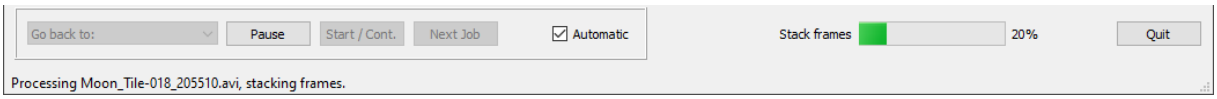
4.4 Starting and Controlling the Workflow

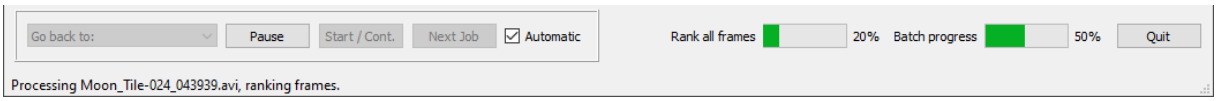
Pressing "Start / Continue" triggers PlanetarySysstemStacker to start processing the jobs in sequential order, either interactively or in batch mode (fully automatic). By checking / unchecking the "Automatic" checkbox at the bottom of the main GUI, the user can switch back and forth between the two modes at any time during execution.



If a large number of similar video files are to be processed, or many images of the same moon panorama are to be postprocessed, the recommended procedure is to process the first job in interactive mode ("Automatic" unchecked). PSS then stops at every processing step and prompts the user to adapt parameters as he or she likes. After the first job the user checks the "Automatic" box, and PSS continues processing all the remaining jobs automatically using the adapted set of parameters.

When batch processing is active, the user at any time can seize control by unchecking the "Automatic" box. PSS just finishes the current processing step and switches back to interactive mode.

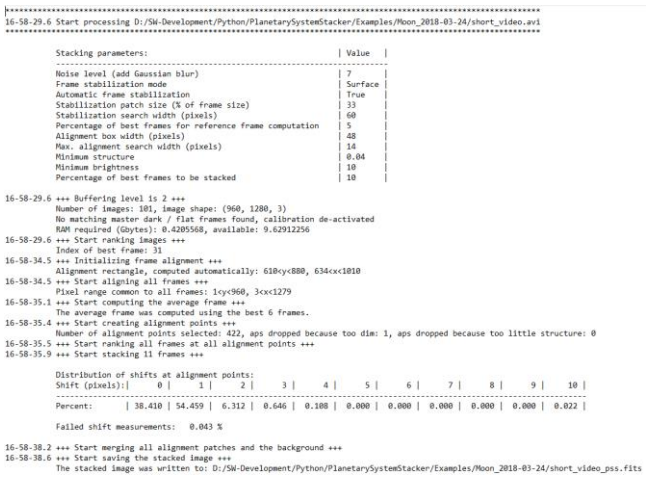




During computationally intensive processing phases PSS displays a progress bar at the bottom of the GUI. If several jobs are processed, a second progress bar shows the fraction of finished jobs. The status line displays information about the files being processed and the current processing step.

4.5 Execution Protocol

Being able to look up the execution details of a job can be very useful, especially if something went wrong in batch mode, or if the resulting image is of lower quality than expected. With PSS the user can choose among various protocol variants.



```
**********************************************************************************************************
16-58-29.6 Start processing D:/SW-Development/Python/PlanetarySystemStacker/Examples/Moon_2018-03-24/short_video.avi
**********************************************************************************************************

        Stacking parameters:                                    | Value   |
        ------------------------------------------------------------------
        Noise level (add Gaussian blur)                         | 7       |
        Frame stabilization mode                                | Surface |
        Automatic frame stabilization                           | True    |
        Stabilization patch size (% of frame size)              | 33      |
        Stabilization search width (pixels)                     | 60      |
        Percentage of best frames for reference frame computation | 5     |
        Alignment box width (pixels)                            | 48      |
        Max. alignment search width (pixels)                    | 14      |
        Minimum structure                                       | 0.04    |
        Minimum brightness                                      | 10      |
        Percentage of best frames to be stacked                 | 10      |

16-58-29.6 +++ Buffering level is 2 +++
        Number of images: 101, image shape: (960, 1280, 3)
        No matching master dark / flat frames found, calibration de-activated
        RAM required (Gbytes): 0.4205568, available: 9.62912256
16-58-29.6 +++ Start ranking images +++
        Index of best frame: 31
16-58-34.5 +++ Initializing frame alignment +++
        Alignment rectangle, computed automatically: 610<y<880, 634<x<1010
16-58-34.5 +++ Start aligning all frames +++
        Pixel range common to all frames: 1<y<960, 3<x<1279
16-58-35.1 +++ Start computing the average frame +++
        The average frame was computed using the best 6 frames.
16-58-35.4 +++ Start creating alignment points +++
        Number of alignment points selected: 422, aps dropped because too dim: 1, aps dropped because too little structure: 0
16-58-35.5 +++ Start ranking all frames at all alignment points +++
16-58-35.9 +++ Start stacking 11 frames +++

        Distribution of shifts at alignment points:
        Shift (pixels)|     0 |     1 |     2 |     3 |     4 |     5 |     6 |     7 |     8 |     9 |    10 |
        ------------------------------------------------------------------------------------------------------
        Percent:      | 38.410 | 54.459 |  6.312 |  0.646 |  0.108 |  0.000 |  0.000 |  0.000 |  0.000 |  0.000 |  0.022 |

        Failed shift measurements:   0.043 %

16-58-38.2 +++ Start merging all alignment patches and the background +++
16-58-38.6 +++ Start saving the stacked image +++
        The stacked image was written to: D:/SW-Development/Python/PlanetarySystemStacker/Examples/Moon_2018-03-24/short_video_pss.fits
```

First of all, the parameter "Protocol detail level" selects the amount of information provided. Level "0" means no output at all. At level "2" detailed info is provided, e.g. on the number of alignment points and the warp distribution. If parameter "Write protocol to file" is checked, all protocol data is appended to the standard file "PlanetarySytsemStacker.log" in the user's home directory. It is a good idea to delete this file every once in a while.

In particular if many jobs are processed, it is

recommended to set the option "Store protocol with results". In this case, in addition to the sequential protocol file the part pertaining to a given job is written as a separate file. Its name is derived from the name of the job result, with the stacking suffix "_pss.tiff" or "_pss.fits" being replaced with "_stacking-log.txt". This way it is easy to associate the log file with the corresponding job output. In the case of postprocessing jobs, the ending "_postproc-log.txt" corresponds to the postprocessing result "_gpp.tiff" or "_gpp.fits".

4.6 Reading Input Data and Buffering

Input data for stacking can be large. In the worst case, the input video file itself is larger than the available RAM. To make things worse, additional to the original video frames PSS uses several image variants which result from applying certain filters:

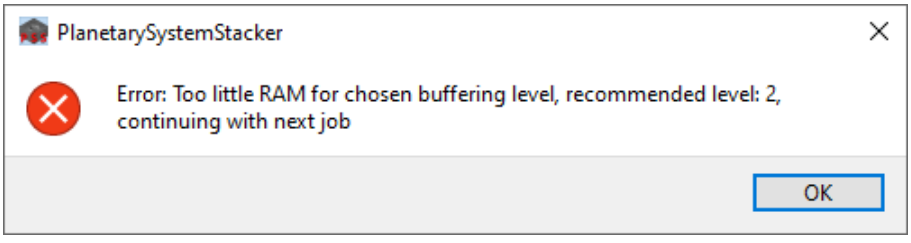
1. Monochrome versions of the original frames (depth as original frames, 8 or 16 bit)
2. Gaussian blur added to 1. (always 16 bit)
3. Laplacian of 2., down-sampled by a factor of 2 in both coordinate directions (always 8 bit)

Since those variants in general are used more than once, processing is fastest if they can be kept in memory. If the RAM is too small, however, this is not possible. Unfortunately, relying on the paging mechanism of the operating system isn't an option, either, since this tends to slow down execution too much.

PSS, therefore, offers a range of buffering levels. Which one is best depends on the size of input data and the size of the available RAM. The user can choose the most appropriate scheme in the configuration dialog (parameter "Data buffering level"). The following options are available:

- Level 0: No buffering. The original frames are read several times during job execution, and derived images are recomputed when needed. Obviously, this mode leads to the maximal computational load.
- Level 1: The "Laplacian of Gaussian" of all frames (variant 3. above) are buffered.
- Level 2: Additionally, the blurred monochrome images (variant 2. above) are buffered.
- Level 3: Additionally, the monochrome images (variant 1. above) are buffered.
- Level 4: Additionally, the original frames are buffered.

On computers with enough RAM Level 4 is the best choice. In general, the default setting "2" is a good compromise between speed and RAM usage.



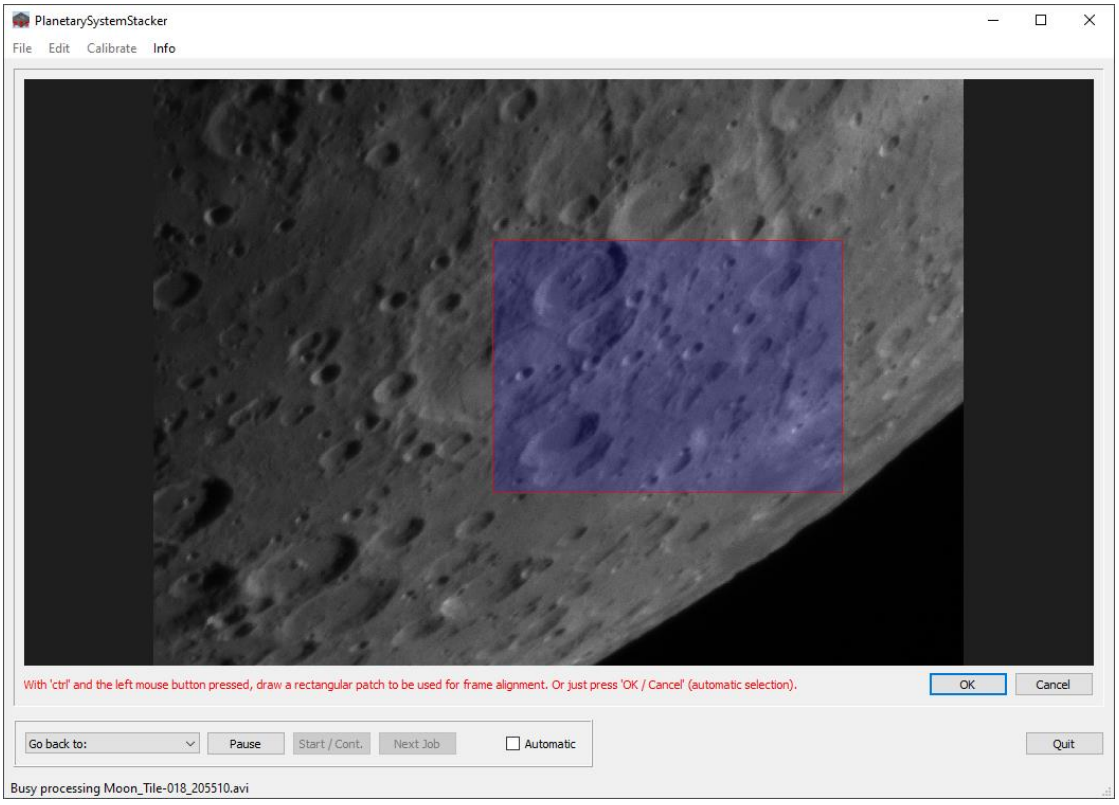
At the beginning of each job, PSS determines how much RAM this job will require, and how much is available. If the protocol level is set to "2", both figures are written to the job protocol. If there is not enough RAM, PSS issues an error and continues with the next job. If the protocol level is set at least to "1", a recommendation of the highest buffering level compatible with the available RAM is written to the job protocol.

4.7 Frame stabilization

Before frames are stacked they first have to be roughly aligned with each other. This way, drift effects of poor mount alignment or guiding inaccuracies are removed. PSS offers two stabilization modes:

- In "Planetary" mode, it is assumed that a planet is located somewhere in the frame, completely surrounded by dark sky. In this case, frame stabilization is very simple and reliable. Therefore, it is strongly recommended to use this option whenever appropriate.
- In all other cases, "Surface" must be used. This is usually the case for moon or sun imaging. In Surface mode, a so-called "stabilization anchor" must be selected in the image. By comparing its position in all frames, PSS determines the drift between them. Obviously, it is crucial to choose the anchor appropriately.

PSS offers to identify the stabilization anchor automatically (check "Automatic frame stabilization"). If this option is chosen, additional parameters ("patch size" and "search width") can be modified to control the automatic algorithm. Obviously, in batch mode automatic stabilization is the only choice, so the configuration parameter "Automatic frame stabilization" has no effect.



If the user chooses to define the stabilization anchor manually in interactive mode, he or she must uncheck the configuration parameter "Automatic frame stabilization". PSS interrupts the workflow, shows the full scene covered by the video to the user and asks to draw the stabilization anchor as a rectangular patch. Since this is the first time in the workflow where the frame viewer appears, this is the time to explain its general features:

- Initially the viewer shows the full frame. Using the scroll wheel (or the keys "+" and "-"), the user can zoom in and out. Moving (panning) the scene laterally is accomplished by moving the mouse while keeping the left mouse button pressed.

- If the contents of the view are to be manipulated, like in this case by drawing the rectangular patch, the "ctrl" key must be kept pressed during the manipulation. As long as the key is pressed, the hand symbol at the mouse pointer location changes into an arrow.
- Drawing a rectangular patch is done by pressing a mouse button (in this case the left one) at one corner of the patch and moving the mouse to the opposite corner.
- Other manipulation options depend on the use case and are described there.

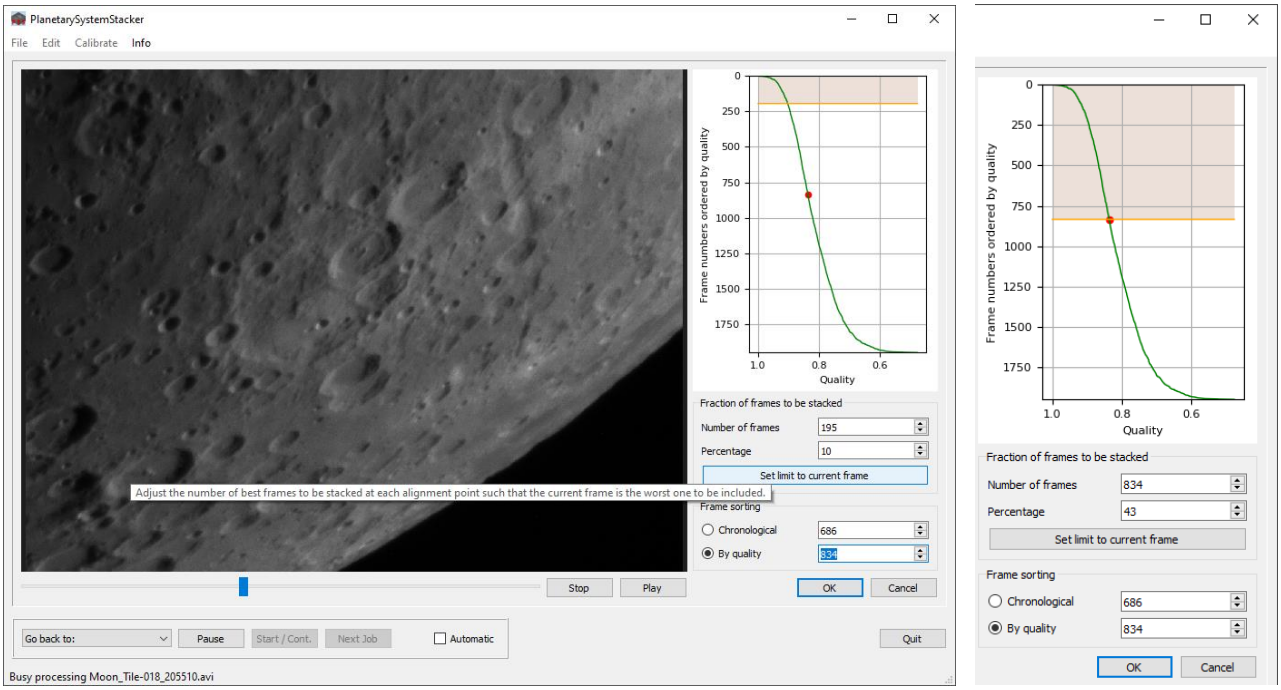
The area chosen as the stabilization anchor should contain as much small-scale contrast (both vertically and horizontally) as possible, and should be large enough. Usually some 30% of the frame size (per co-ordinate direction) is a good choice.

Once the patch is selected, the user acknowledges the choice by pressing "OK". If "Cancel" is pressed instead, PSS defines the stabilization anchor automatically. This also happens if the patch selected by the user is too small (less than 20%) or too large (greater than 70%).

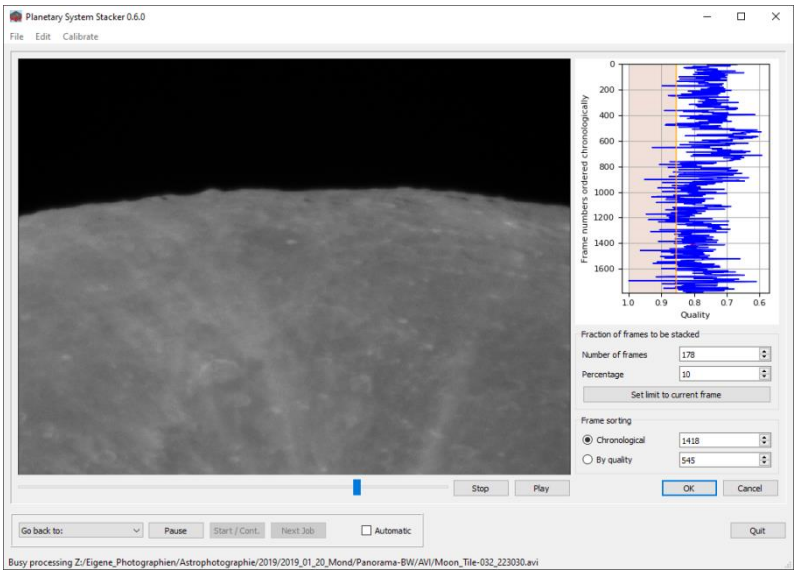
After completing the frame stabilization process, PSS computes the image area common to all frames. This area is the basis for all processing steps from this point on.

4.8 Setting the Stacking Fraction

In the next step the user decides how many frames are used for stacking. The GUI opens a view on the scene where frames can be displayed either ordered by decreasing overall quality or chronologically. With a horizontal slider the user can scroll through all frames. As an alternative, he or she can press "Play" / "Stop" to display frames as a video.



The graph on the right shows the position of the current frame in the selected order (vertical axis) and its relative quality (horizontal axis). The shaded area shows the fraction of frames to be stacked. The stacking limit can be moved by changing the input editor fields for "Number of frames" or "Percentage". As an alternative, pressing the key "Set limit to current frame" moves the border of the shaded area to the position of the frame being displayed. The result is demonstrated in the cutout on the right-hand side.

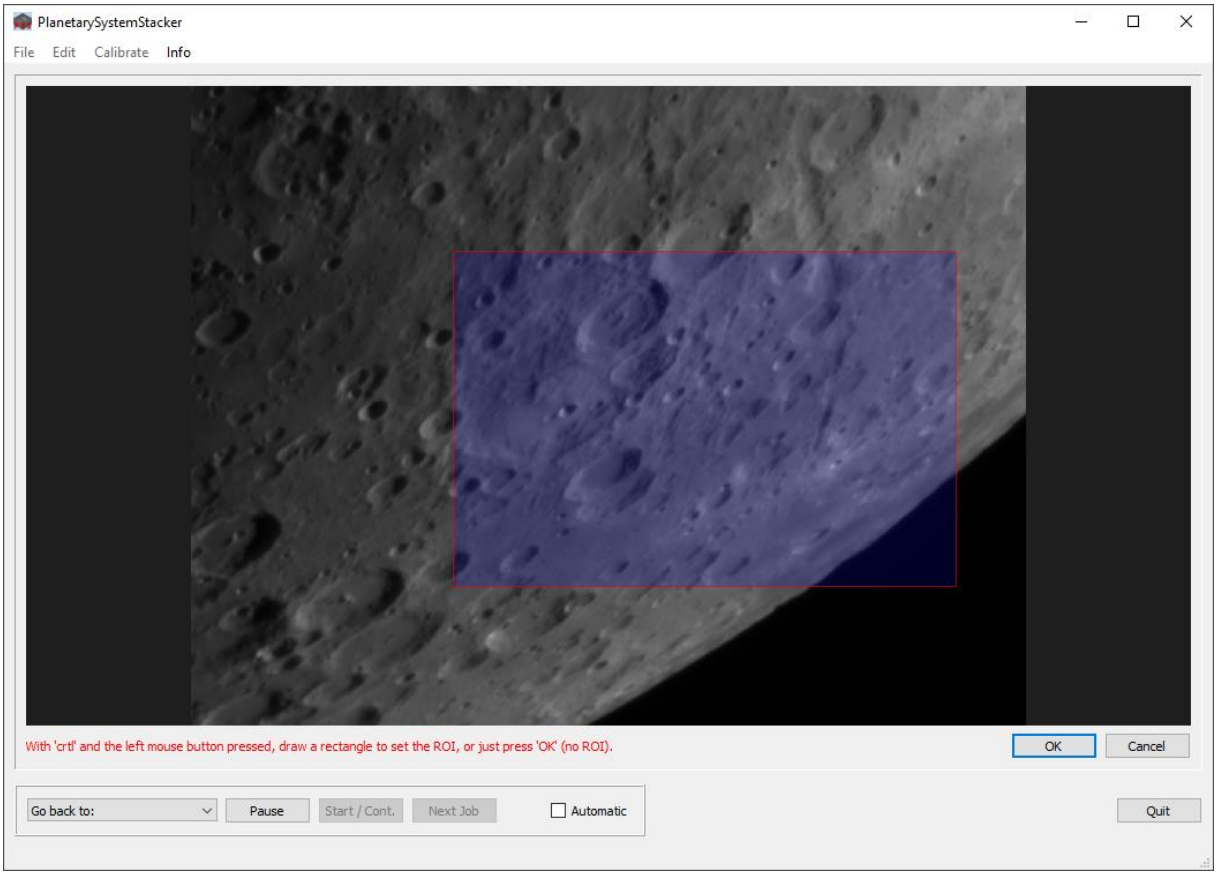


By setting the "Frame ordering" checkbox, frames can be ordered either by quality or chronologically. In the latter case, as demonstrated to the left, the quality graph usually is quite busy, as a consequence of rapid quality variations caused by seeing.

Please note that if chronological ordering is selected, the shaded area showing the frames used for stacking is oriented vertically.

The selection of the stacking limit is acknowledged by pressing "OK".

4.9 Setting a Region of Interest (ROI)



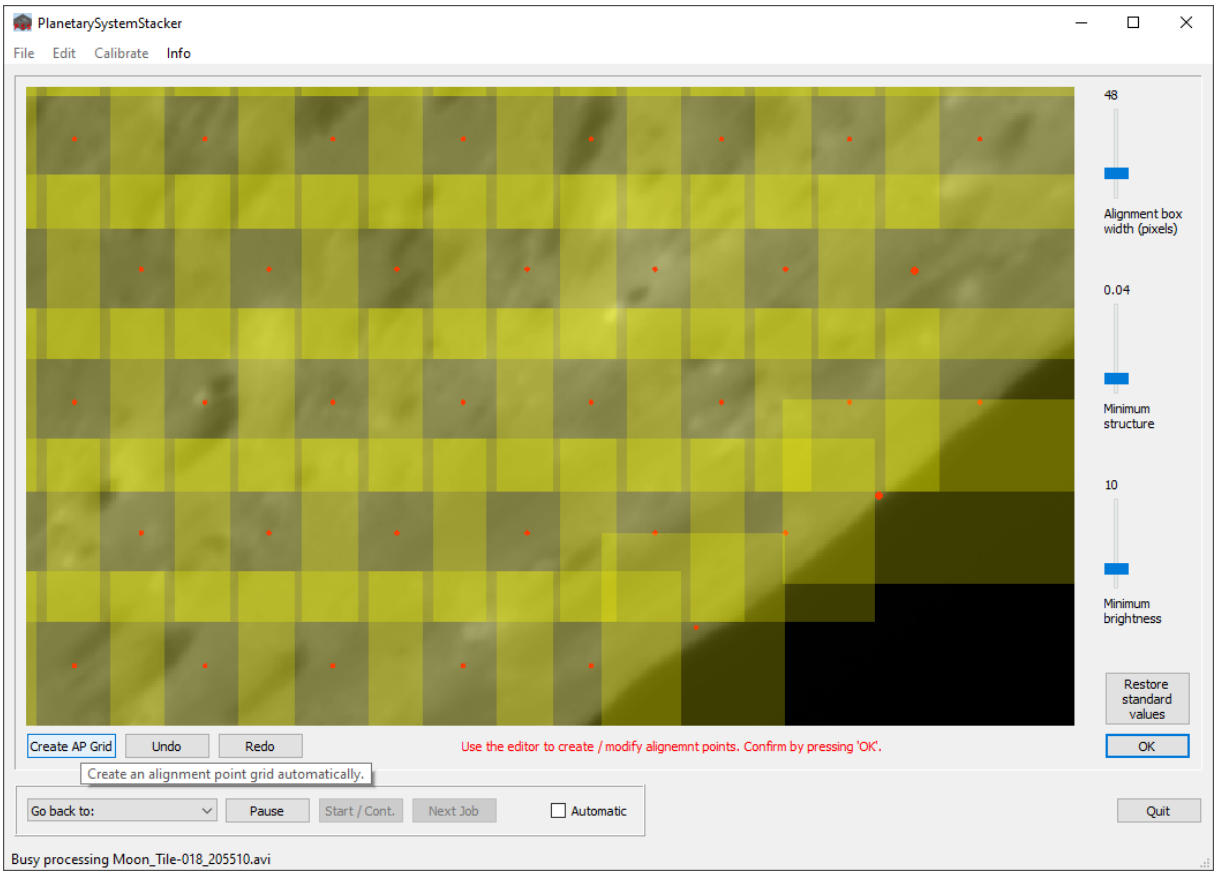
Optionally, stacking can be restricted to a so-called region of interest (ROI) smaller than the intersection of all frames. To specify its size a viewer opens and prompts the user to select the ROI as a rectangular patch (as described in Section 4.7 above). Selecting the full view instead of a ROI is accomplished by just pressing "OK".

4.10 Selecting Alignment Points

In the next view the alignment points (APs) used to de-warp the individual frames are selected. Again, a viewer opens, along with controls to the right and below. Initially, the view shows the complete scene (if a ROI was selected, the view is restricted to this area). Alignment points can be generated automatically or set individually by the user:

- To create APs automatically, press the button "Create AP Grid". The size of individual AP patches is controlled by changing the "Alignment box width" slider on the right. Thresholds for excluding areas which are too dim (e.g. sky background) or contain too little structure can be changed by moving the two other sliders.
- If slider settings have been changed, they can be reset to default values by pressing the button "Restore standard values".
- After changing the slider settings, press "Create AP Grid" again to compute a new AP grid using the current settings.
- APs can be deleted, added and modified manually. This way the user can adapt the grid as required. All these manipulations are done while keeping the "ctrl" key pressed.
    - To delete an AP, place the mouse pointer close to its center (red dot), and press the right mouse button.
    - To delete a whole AP region, open a rectangular patch by moving the mouse while keeping the right mouse button pressed. When the button is released, all APs in the patch are removed.
    - To move an AP, press the left mouse button at an AP and drag the AP with the mouse to the new location.
    - To change the size of an AP patch, move the mouse close to the AP center and use the scroll wheel to change its size.
    - To add an AP, left click on the desired location. An AP with (initially) the standard size (see slider on the right) is created there.
- All those AP operations can be undone / redone by pressing the buttons "Undo" or "Redo". The size of the undo stack is unlimited. And, as with all other PSS viewers zooming and panning is supported here as well.
- It is not required that the APs cover the object completely. If the object "shines through" a hole in the AP grid, the consequences for the stacked image in this area are:
    - There is no correction for local image warping at this place. Only the global frame shift is corrected.
    - The set of frames stacked is computed on the basis of the global frame quality, not on local sharpness.

  In general this leads to a lower resolution in those areas.



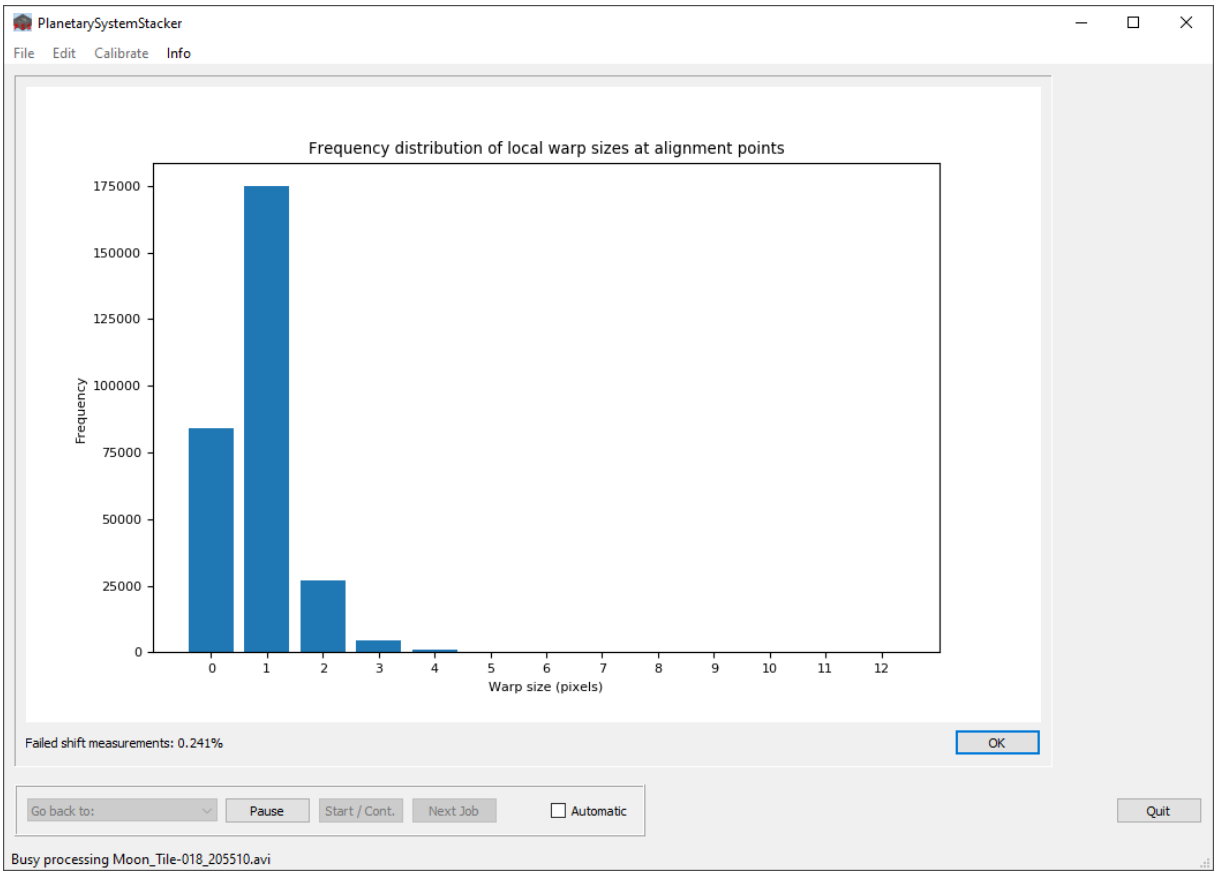
The optimal AP size very much depends on the quality of the input data. If frames show very little local contrast, larger AP patches lead to better results. If the resolution is very good, but seeing-induced warping is strong, small AP patches are better. The best practice is to experiment with different AP sizes, starting with a somewhat larger value (e.g. 100 pixels), and trying smaller sizes until the quality of the stacked image does not improve any further.

4.11 Frame Stacking

When the AP selection is completed, PSS has gathered all the information it needs to stack the frames. First, at every AP it identifies the sharpest frames to be used for stacking. Since the seeing is a very local phenomenon, frame sets will be different for different APs. Then, for every AP and every contributing frame the local displacement relative to a reference frame is measured, and the shifted AP patch added to the stacking buffer. Progress bars are updated regularly throughout the process.

When stacking is completed, a graph shows the frequency distribution of local displacements at all APs and all contributing frames. Usually, small displacements (a few pixels) occur most often. If the distribution extends too much towards larger numbers, this could be due to a low stacking quality. In this case it is recommended to experiment with different parameters, for example:

- Increase the "Noise level (Gaussian blur)" parameter.
- Increase the AP size.
- Eliminate APs in areas with too little structure.



Another quality indicator for the warp shift compensation is the fraction of failed shift measurements which is displayed below the shift distribution graph. If this value is above 5%, it is written in red color. In this case it is a good idea to experiment with different parameter settings, as discussed above. A high value, however, not necessarily means that the results are bad. Other explanations are a very low contrast in the image, or oversampling in the optics / camera configuration.

After inspecting the warp distribution, press "OK" to finish the current job. The stacked summation image is written in 16bit .tiff (or .fits) format to the file system into the same folder where the input is located. The name is taken from the input video file or image directory, extended by the suffix "_pss.tiff" (or "_pss.fits").

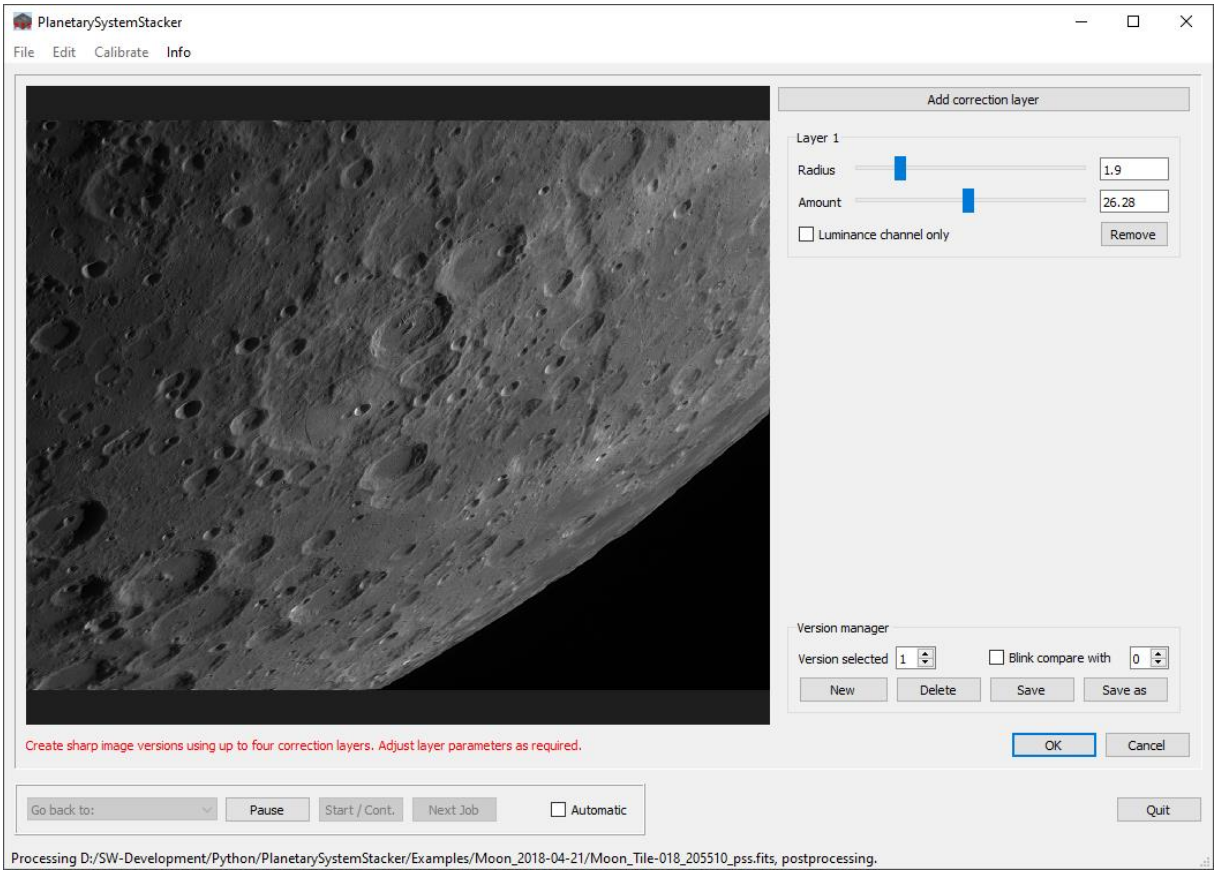
4.12 Postprocessing (Image Sharpening / Smoothing)

As explained in Section 4.3, PSS can sharpen and / or smooth a still image in a postprocessing job. Input to this type of job is a single image file, and tailoring and applying the filter is the only job activity.

As an alternative, postprocessing of a stacked summation image can be included as the final step in a stacking job. In this case, the configuration parameter "Stacking plus postprocessing" must be checked before the end of the stacking job.

This section describes how postprocessing works in PSS. As explained above for stacking, postprocessing can be performed either interactively, or in fully automatic batch mode. In the case of several postprocessing jobs with similar images, executing the first job interactively and then continuing in batch mode (by setting the "Automatic" checkbox) is recommended as best practice. After all, no reasonable default values can be specified for sharpening / smoothing, so finding a good set of parameters is possi-

ble only in an interactive trial and error loop. PSS was designed to support this interactive process as effectively as possible.

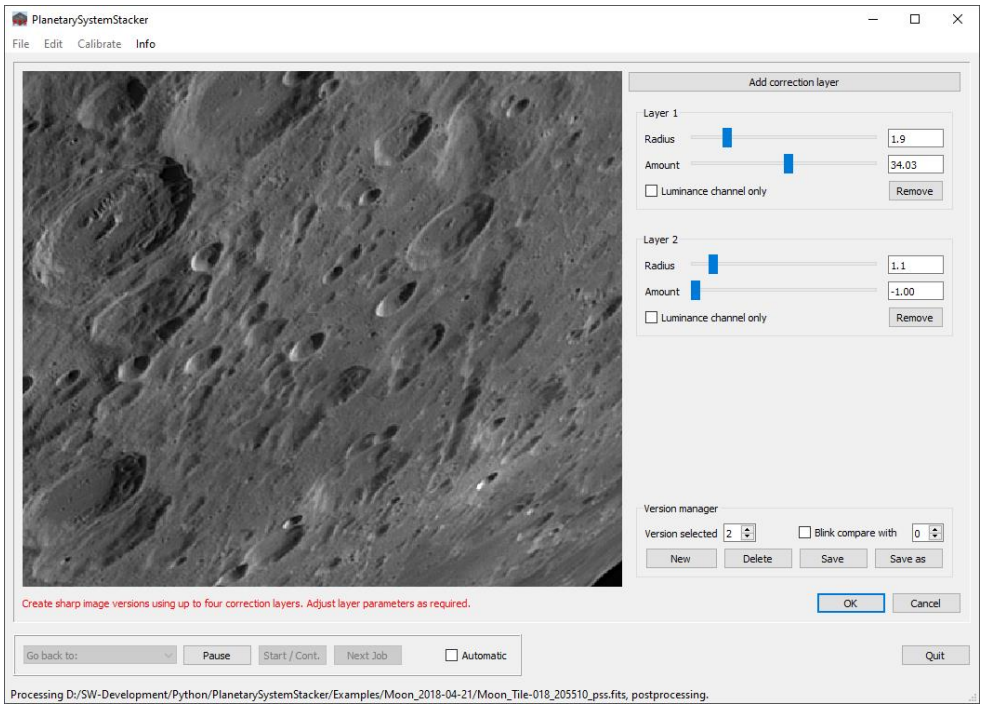


To support experimentation with different parameter sets, PSS offers to define and compare an arbitrary number of processing versions. With the "version manager" at lower right new versions can be created. With the "Version selected" spin box a different version can be selected. Other GUI elements in this view, such as "Delete", "Save", "Save as", and the filter details in the panel at upper right refer to the version currently selected. Version number "0" is reserved for the original image, with no correction being applied.

Sharpening in PSS is implemented using a multi-layer unsharp masking algorithm. A hierarchy of up to four layers can be defined. Each layer is defined by

- Its spatial frequency (in pixels).
- The strength (or amount) with which the layer is applied. Choose a positive value to sharpen the image, or a negative value to smooth it. Smoothing can be useful to remove high-frequency noise created by a previous sharpening layer.

A layer can either affect the luminance channel only, or all color channels. The choice is made by setting a checkbox accordingly.

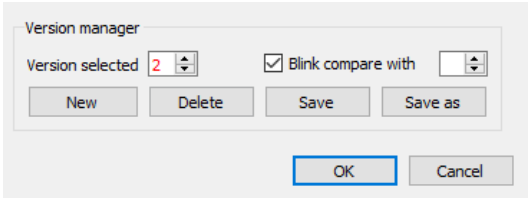


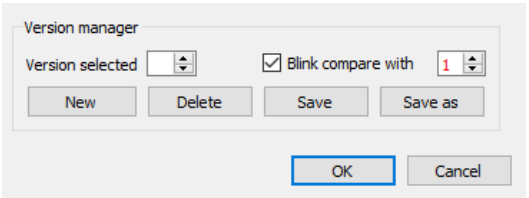
In the example on the left, a second layer (with a negative amount) was added to remove high-frequency noise in the luminance channel.

The viewer's zoom function is used to look at the filter effect in detail.

When the postprocessing view opens in the GUI, the viewer to the left shows the input image, with the sharpening model selected in the version manager being applied. If PSS is called for the first time, in addition to the original image (Version "0") it starts with an initial version 1, consisting of a trivial single correction layer (Radius = 1.0, Amount = 0., i.e. no correction is being applied, sharpening in all color channels).

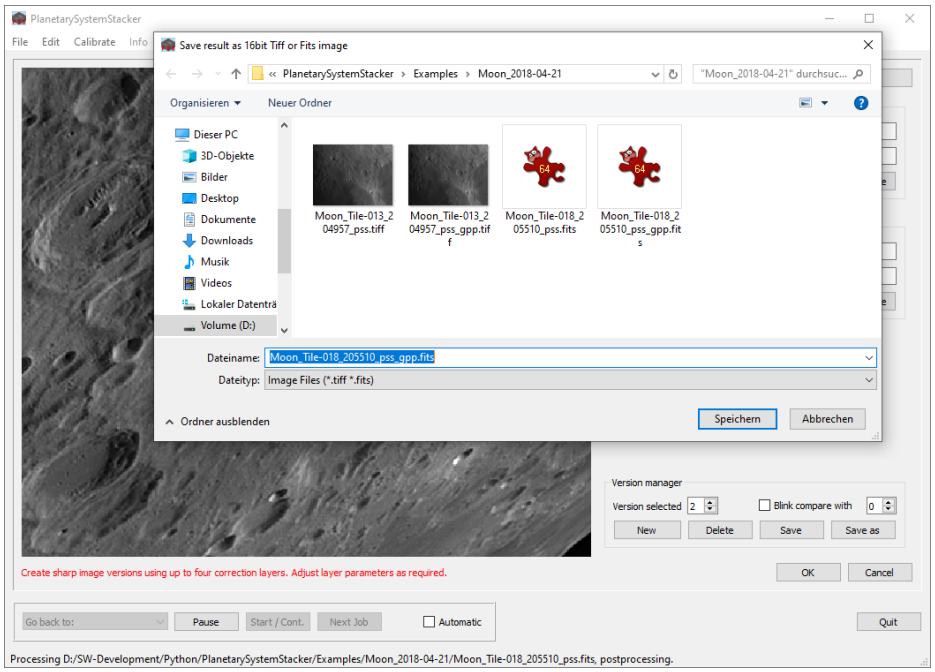
Layers are added / removed by pressing the buttons "Add correction layer" and "Remove", respectively. When layer parameters are changed, the image viewer is updated immediately. In the case of large images, a "busy" message appears in the status line until the update is finished.





Different versions can be compared with each other using the "blink comparator". First, in addition to the selected version, another version is chosen in the spin box on the right. When both versions are set, checking the "Blink compare with" box causes the viewer to alternate between them. The number of the version currently displayed in the viewer is shown in red. While the blink comparator is running, the layer parameters of the selected version can be changed. Layers can be added or removed. Even the selection of the two versions displayed can be changed. Last but not least, as with all PSS viewers, zooming and panning can be used to inspect the effect of the sharpening model in detail.

Once the user is satisfied with the selected version, pressing the "OK" button triggers PSS to save this image version along with the input image (as a 16bit .tiff or .fits image). The file name is the one of the input image, extended by the suffix "_gpp.tiff" or "_gpp.fits". The parameters of all postprocessing ver-



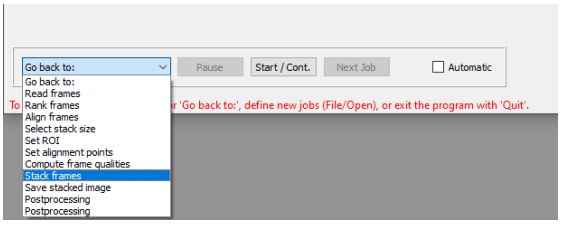
sions and the number of the selected version are saved in the configuration file. When the postprocessing view opens next time, all versions and the selection index are restored.

Additionally, the user can save the selected postprocessing version to an arbitrary file system location by pressing the "Save as" button.

Different postprocessing configurations can be saved and restored at any time using the menu entries "File / Save configuration" and "File / Load configuration" (see above in Section 4.1)..

4.13 End of Program

When a job is finished in interactive mode, the user can instruct PSS to repeat the execution starting with a selected job phase. Pressing the combo box "Go back to" opens a panel with the available phases. This can be useful if one wants to change parameters affecting later job phases only, without repeating the whole job.

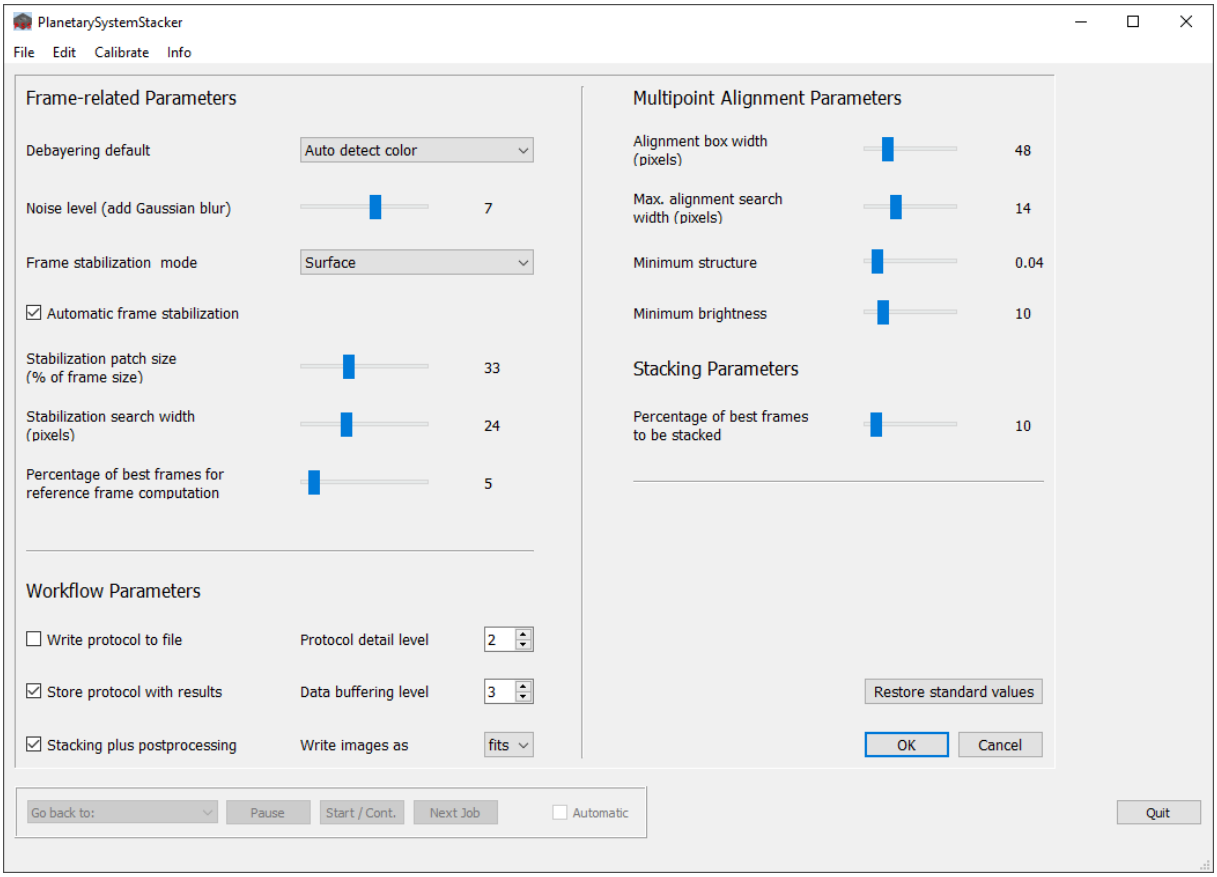


When all jobs are completed, PSS returns to its initial state. New jobs can be selected using the "File / Open" dialog. If no more jobs are left, the user can quit the program by pressing the "Quit" button. As a last activity PSS saves the current configuration in the standard configuration file and closes all protocol files.

Appendix A: Configuration Parameters

The configuration dialog allows entering user-specific values for various parameters. They are arranged in the following groups:

- Frame-related parameters
- Multipoint alignment parameters
- Stacking parameters
- Workflow parameters



The following tables contain definitions of all parameters.

Frame-related parameters

Debayering default	Debayering pattern used in stacking jobs if no other pattern is specified for that job explicitly (via the context menu) in the job editing dialog (see Section 4.3).
Noise level	Width of the Gaussian filter used to blur the original images before shifts are determined. For noisy images this value should be increased. If the images show very little noise, a lower value can improve the precision.
Frame stabilization mode	For planetary imaging, choose "Planet". This option is valid only if the object in all directions is surrounded by dark sky. In

	all other cases, "Surface" must be selected.
Automatic frame stabilization	Only used in "Surface" mode (see above). In this case, the global alignment of all frames relative to each other is based on the shift measured at a selected surface patch. The patch should contain good local contrast in both coordinate directions. If this option is checked, PSS selects the patch automatically. Otherwise, the user is prompted to select the patch manually. In batch mode, this option is ignored, and PSS always selects the patch itself.
Stabilization patch size	Size of the stabilization patch in both coordinate directions relative to the frame size (in percent).
Stabilization search width	Maximum allowed shift between two consecutive frames in pixels per coordinate direction. If no matching is found within this search width, the shift is set to zero, and a warning is added to the protocol.
Percentage of best frames for reference frame computation	The reference frame is computed by averaging the best frames after global frame stabilization. This parameter determines the fraction of frames used for the reference frame.

Multipoint alignment parameters

Alignment box width	Standard size of the quadratic alignment patches around each alignment point used for measuring the local shift. In the case of noisy images or low local contrast, a larger box size is recommended. This parameter is used by the automatic AP grid generation.
Max. alignment search width	Maximum local warp displacement (after global frame stabilization). The warp is measured using a local search algorithm. If within the specified search width no matching is found, the local warp shift is set to zero.
Minimum structure	Threshold used by automatic AP grid generation to avoid alignment points in places with too little local structure. Increase this value if APs with too little structure have been created.

Minimum brightness	Threshold used by automatic AP grid generation to avoid alignment points in places with too little light. The main purpose of this feature is to avoid APs in the background sky.

Stacking parameters

Percentage of best frames to be stacked	At every AP a fixed number of best frames are used for stacking. This parameter specifies the fraction of frames used.

Workflow parameters

Write protocol to file	If checked, a protocol is written (appended) to the file "PlanetarySystemStacker.log" in the user's home directory.
Protocol detail level	Level of protocol detail. 0: No protocol; 1: Only major steps listed; 2: Detailed info for each processing step.
Store protocol with results	If checked, in addition to the global protocol file, for every job the protocol section related to it is stored in a separate file along with the job result (extension "_stacking-log.txt" or "_postproc-log.txt", depending on the job type).
Data buffering level	Specifies how much data PSS should keep in RAM for reuse. Possible values range from 0 (no buffering) to 4 (keep all intermediate results in RAM). It is recommended to use as high a value as possible. If not all data fit in memory, however, it is better to use a lower buffering level than relying on the paging mechanism of the operating system.
Stacking plus postprocessing	Only used for stacking jobs: If checked, postprocessing is included as an additional phase at the end of the stacking job. In this case, both the stacked summation image and the sharpened version are written to the file system.
Write images as	Images resulting from stacking or postprocessing jobs can be written either in .tiff or .fits format. This parameter selects the format to be used in automatic save operations. Independent of this choice, the user can save images in both formats by pressing the "Save as" buttons.

Parameters can be modified either in the configuration dialog (menu "Edit / Edit configuration") or during job processing in interactive mode. The configuration also includes the current sharpening models as defined in the postprocessing dialog.

Before PSS terminates, the current parameter set is written into the configuration file "PlanetarySystemStacker.ini" in the user's home directory. The next time PSS starts it restores the configuration using that file. Additionally, at any time during execution the current parameter set can be saved to or loaded from a file of the user's choice (menu "File / Save configuration" and "File / Load configuration").

Appendix B: Linux Installation Instructions

As an alternative to using a self-contained installer, the following instructions show for various Linux distributions how the PSS source code can be installed together with the required Python 3 environment. First, OS-specific instructions show how to prepare the Python 3 environment. Finally, the last section shows hot PSS itself is downloaded, installed and run. This last part is independent of the Linux distribution.

Ubuntu / Debian:

The following instructions were tested using Xubuntu 18.04 LTS 64bit and Debian 10.1.0 amd64.

These libraries available through the normal installation repository are needed to have PSS run under Ubuntu, or to install PSS from github. This assumes python3 was installed during the installation of Ubuntu itself. They can also be installed through the xwindows program "synaptic".

- python3-opencv
- python3-matplotlib
- python3-psutil
- python3-pyqt5
- python3-scipy
- python3-astropy
- python3-skimage
- python3-pip
- git

Of course some of these may already be installed, and apt-get will tell you if you try to install them again. (Either apt or apt-get can be used below.)

This command will do it (from a terminal):

```
sudo apt install python3-opencv python3-matplotlib python3-
psutil python3-pyqt5 python3-scipy python3-astropy python3-
skimage python3-pip git
```

Alternatively you can do them individually (from a terminal):

```
sudo dnf install python3-opencv
sudo dnf install python3-matplotlib
sudo dnf install python3-psutil
sudo dnf install python3-pyqt5
sudo dnf install python3-scipy
sudo dnf install python3-astropy
sudo dnf install python3-skimage
sudo dnf install python3-pip
sudo dnf install git
```

In addition to these, you need to install the math library mkl using the python-pip installation manager.

```
sudo pip3 install mkl
```

or as the program suggests:

```
pip3 install --user mkl
```

This tries to install both the python2.7 and python3 versions. Only the python3 version is needed, and it may give an error if python2.7 is not installed. One can ignore that error message.

This is a somewhat long installation and takes a few minutes or so. All the required dependencies are now installed.

Fedora:

These libraries available through the normal installation repository are needed to have PSS run under Fedora, or to install PSS from github. This assumes python3 was installed during the installation of Fedora itself. They can also be installed through the xwindows program "dnfdragora"

- python3-opencv
- python3-matplotlib
- python3-psutil
- python3-qt5
- python3-scipy
- python3-astropy
- python3-scikit-image
- python3-pip
- git

Of course some of these may already be installed, and dnf will tell you if you try to install them again.

This command will do it (from a terminal):

```
sudo dnf install python3-opencv python3-matplotlib python3-
psutil python3-qt5 python3-scipy python3-astropy python3-
scikit-image python3-pip git
```

Alternatively you can do them individually (from a terminal):

```
sudo dnf install python3-opencv
sudo dnf install python3-matplotlib
sudo dnf install python3-psutil
sudo dnf install python3-qt5
sudo dnf install python3-scipy
sudo dnf install python3-astropy
sudo dnf install python3-scikit-image
sudo dnf install python3-pip
sudo dnf install git
```

In addition to these, you need to install the math library mkl using the python-pip installation manager.

```
sudo pip install mkl
```

or as the program suggests:

```
pip install --user mkl
```

This tries to install both the python2.7 and python3 versions. Only the python3 version is needed, and it may give an error if python2.7 is not installed. One can ignore that error message.

This is a somewhat long installation and takes a few minutes or so. All the required dependencies are now installed.

Installation of PSS (same for all Linux distributions):

When the Python3 environment is completed according to the OS-specific instructions above, to install PSS from github the following steps are needed. It needs to be installed in the /opt directory:

```
cd /opt

sudo git clone https://github.com/R...stemStacker.git
```

When the download is complete:

```
cd /opt/PlanetarySystemStacker/Source
```

Now you can run PSS with the following command:

```
python3 planetary_system_stacker.py
```

Or from your home directory (this command can be put into a launcher to automate running):

```
python3
/opt/PlanetarySystemStacker/Source/planetary_system_stacker.py
```