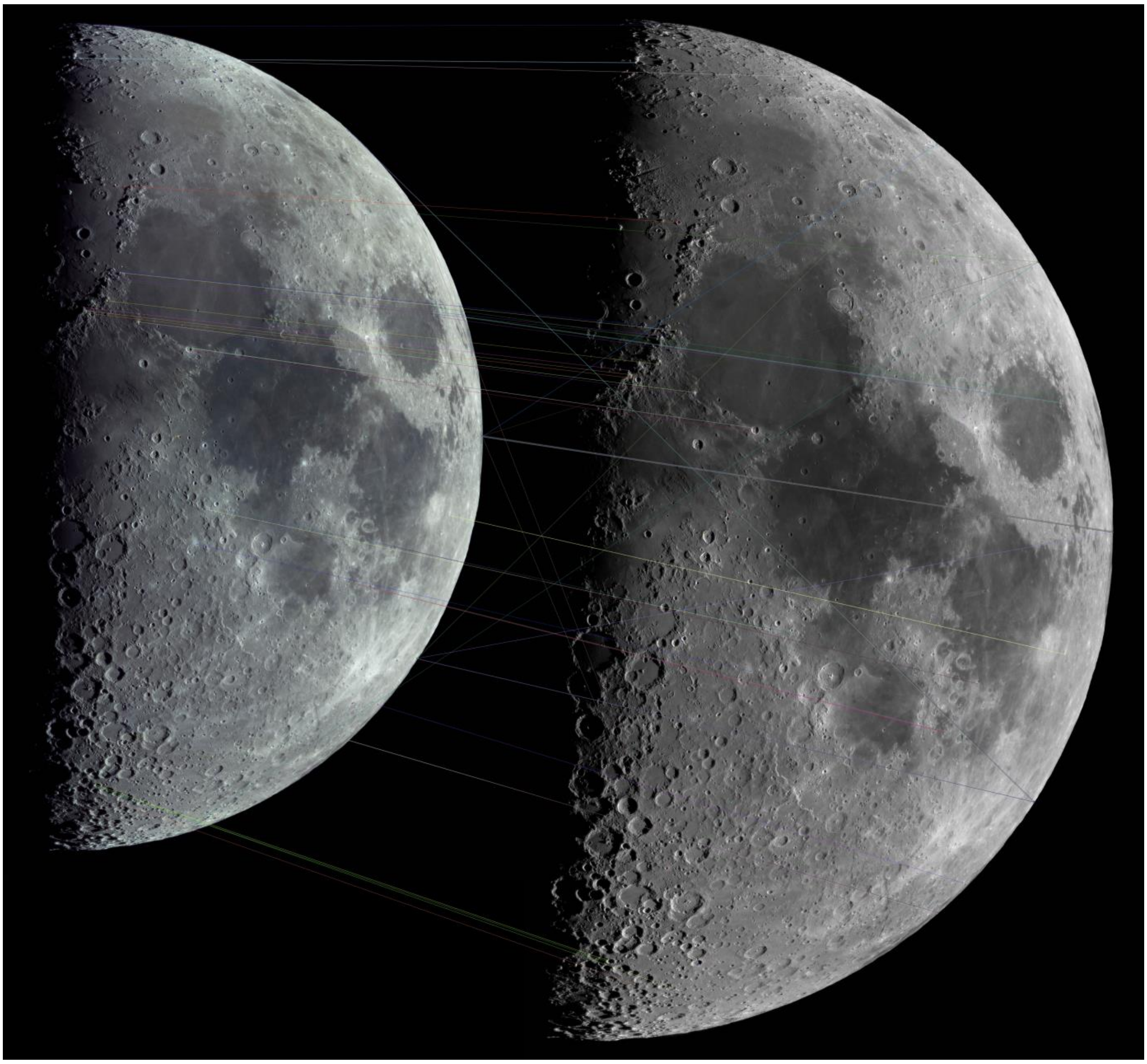
PlanetarySystemLRGBAligner for the automatic creation of LRGB composites of the moon and planets



March 24, 2018. Celestron 11, Color (ASI178MC) and B/W (ASI120MM-S) panoramas and keypoint matches. © Rolf Hempel, 2018

User Guide (Version 0.5.1, Nov. 28, 2018)

Rolf Hempel

Table of Contents

1. Introduction

Imaging of planetary system objects, such as the Moon or planets, at high magnification suffers from the degrading effect of the Earth's atmosphere. Since the effect grows with decreasing wavelength, images are sharpest if taken through red or infrared filters. Color images, on the other hand, suffer from strong seeing effects at shorter wavelengths.

By combining a sharp B/W image taken in the near infrared and a color image with much less resolution, the LRGB technique has the potential to create a color image with the full resolution of the near-infrared image. Technically, the B/W image is assigned to the "luminance" channel of the combined image, while the color image's RGB channels only provide the color information.

For this to work properly, both images first have to be registered pixel-wise with high precision. For extended objects such as the moon, this requires more than a simple affine mapping. Instead, it is necessary to perform a local warping of the color image across the whole frame to match it with the B/W one. This cannot be done manually by using standard image processing software.

PlanetarySystemLRGBAligner does this image registration automatically using a two phase approach: First, it computes a rigid homography mapping to approximately match the color and B/W images. In a second phase, an "optical flow" algorithm adjusts the registration of both images at every pixel location.

2. Changelog

2.1 Minor bug fixes: 0.5.1 (November 2018)

- Bug fixed if configuration parameter "Use Gaussian filter" was unchecked
- Error messages added if something goes wrong in loading images
- Window title of error message corrected.
- Minor inconsistencies in user guide corrected.

2.2 First published version: 0.5.0 (November 2018)

- Fully operational software for Windows 7 / 8 / 10

3. System Requirements and Software Installation

PlanetarySystemLRGBAligner is written in Python 3.5 and only uses standard Python libraries. It should, therefore, run on any operating system where this Python version is available. So far, however, an automatic installer has been built for Windows only. If users want to run the program on other platforms, they can download the source code from the Github repository and execute the file "planetary_system_LRGB_aligner.py" with the Python 3.5 interpreter.

3.1 Windows (7 / 8 / 10)

The PlanetarySystemLRGBAligner software is distributed as a single file: the Windows installer "PlanetarySystemLRGBAligner_V0.5.0_Windows-Installer.exe". If this installer is used, it is not necessary to have Python installed on the machine. Before installation, any earlier PlanetarySystemLRGBAligner 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, PlanetarySystemLRGBAligner 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 completely removes the installed software.

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

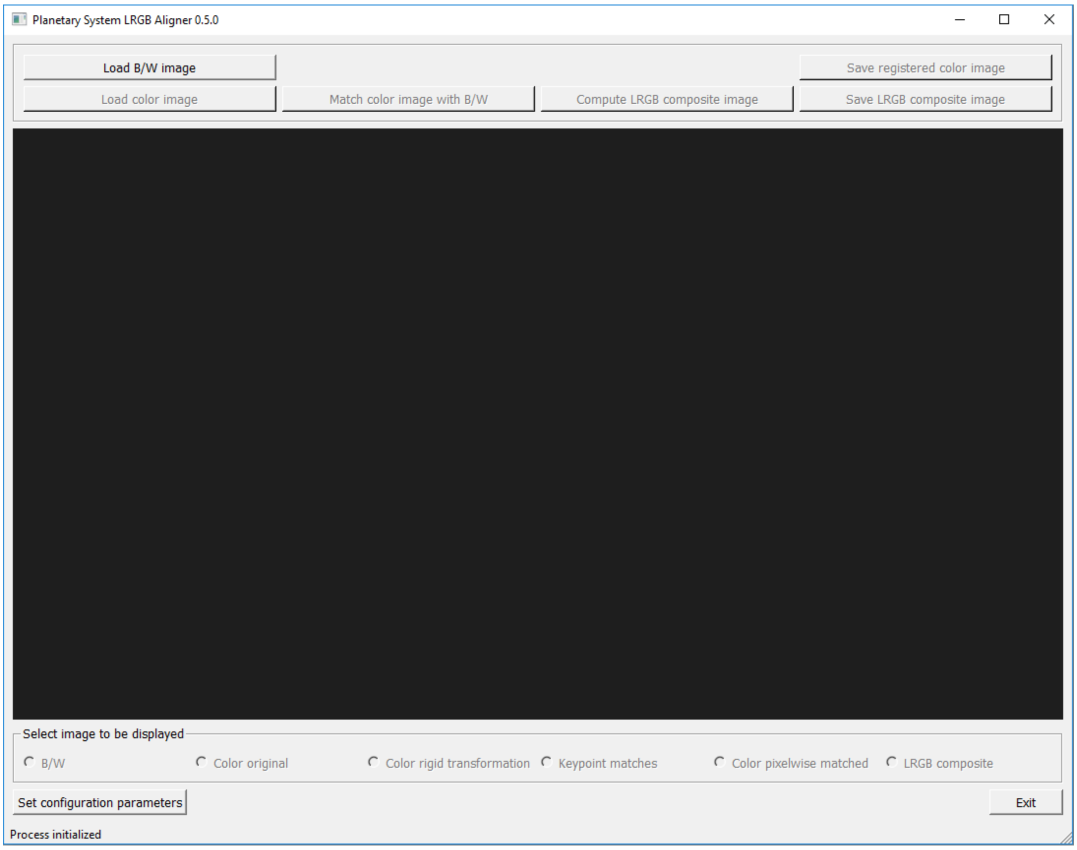
The whole software was developed on an Acer laptop computer (type "Acer Aspire V5-573G", Intel Core i5-4200U, 12 GBytes RAM) running Windows 10 Professional, version 1803. It was tested successfully on a Windows 7 system as well.

The folder "Images", located in the installation directory, contains an example B/W and color image pair with which all software functions can be tested.

4. Program Execution

PlanetarySystemLRGBAligner communicates with the user via a graphical user interface (GUI). The GUI delegates all computations to a separate "workflow" thread. The GUI, therefore, stays fully responsive even if a long-running computation is going on in the background.

4.1 Layout of the Main Window



The main window contains five sections (from top to bottom):

- The control section contains buttons to start the various program functions. If currently a function does not make sense or cannot be used, the corresponding button is grayed out and de-activated. For every button a tooltip gives a short explanation of its function. To make it visible, just move the mouse over the button.
- An image viewer can be used to inspect the input images and those which are created at runtime. Panning and zooming are performed by using the left mouse button and scroll wheel, respectively.
- The user presses one of the radio buttons in the third section to select the image to be shown in the image viewer. As soon as an image becomes available during the workflow, the corresponding radio button is activated. Panning and zooming affect all images in the same way. By zooming in and then switching back and forth between images the user can therefore check in detail how well they are registered with respect to each other.
- The two buttons "Set configuration parameters" and "Exit" can be used at any time. If configuration parameters are changed during the workflow, images computed so far are invalidated by de-activating the corresponding control and radio buttons. They become active again after the user has started a new computation.

- The status line summarizes the current state of the computation process. When a computation is active in the background, the word “busy” appears at the end of the status line. Even in this case the user may use the GUI functions to inspect images which are available already.

4.2 Loading input images

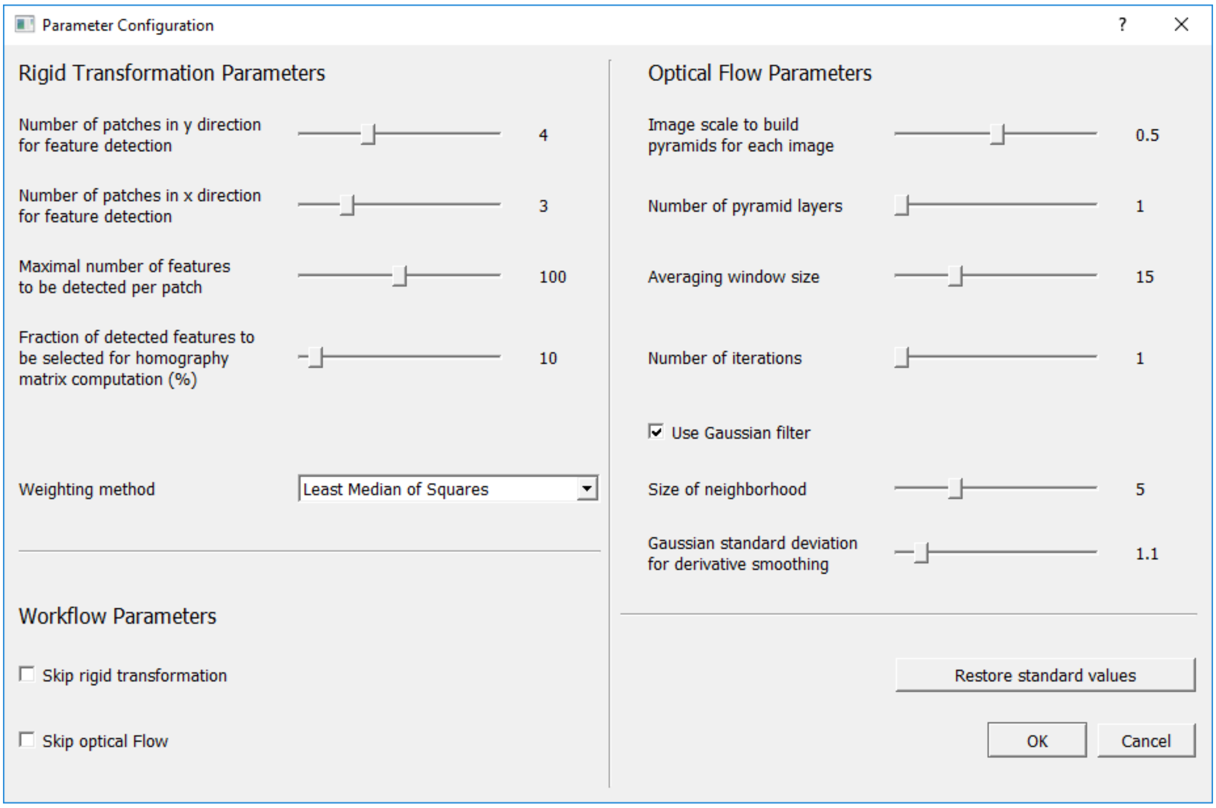
The B/W image is loaded by pressing the button “Load B/W image”. A standard Windows file chooser opens to select the file. Image formats “tif”, “png” and “jpg” are supported. For best results, it is recommended to use 16bit “tif” (which can also be written as “tiff”). All images created later in the workflow have the same size (pixel counts in y and x directions) as the B/W image.

The color image is loaded by pressing the button “Load color image”. It may differ in file shape, resolution and orientation from the B/W image. The differences should not be too large, however, or otherwise the image registration may fail. As soon as both images are available, the button “match color image with B/W” is activated.

Input images may have either 8 or 16 bit resolution per channel. In the latter case, the file format must be “tif” (or “tiff”). If both the B/W and color images are 16bit, the output images will be 16bit as well. To store them at full resolution, again use the “tif” format. If one of the input images is 8bit only, or a file format other than “tif” / “tiff” is used for storing, output images are reduced to 8bit resolution per channel.

4.3 The configuration dialog

Before starting the image registration workflow, please have a look at the parameters used by the two registration algorithms. To start the configuration dialog, press the GUI button “Set configuration parameters”. The following window opens:



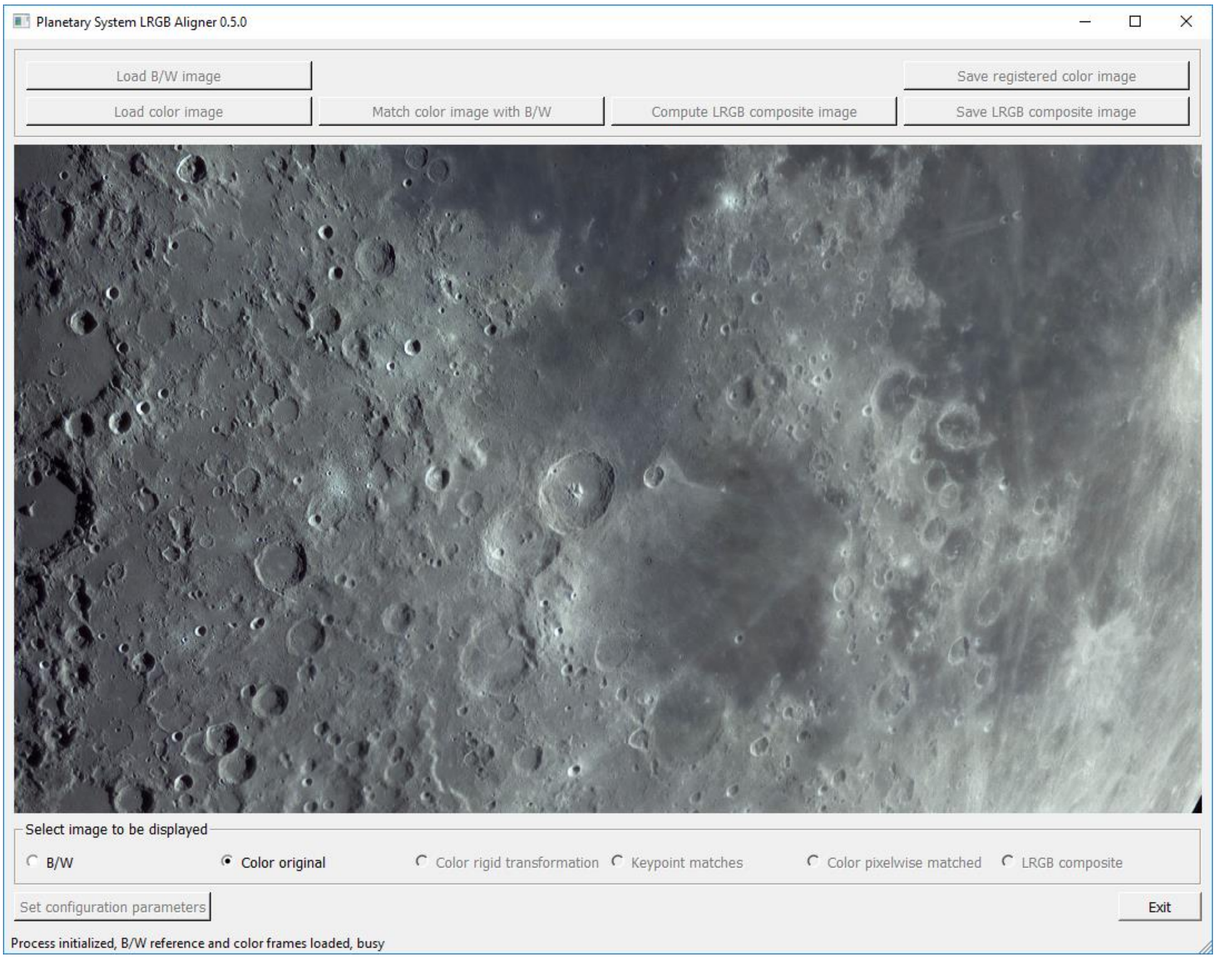
Parameters are organized in three sections:

- Under the heading "Rigid Transformation Parameters" the user controls the behavior of the first registration phase. If things go wrong, it is most likely in this phase. The [ORB algorithm] identifies so-called keypoints in both images and tries to match the corresponding points with each other. Those keypoint pairs are then used to compute the rigid transformation, which consists of translation, rotation, (unisotropic) scaling and skew.
  If the image contains areas with very prominent surface contrast (such as the moon's terminator region) and ones with very low contrast (such as the moon's limb areas), most likely keypoints are not distributed well across the image. In the example above, typically they would all be located close to the terminator. In such a situation the mapping between the color and B/W images becomes almost singular, and the result is very inaccurate.
  As a remedy, the program offers to split the image in a grid of rectangular patches. The user can specify the numbers of these patches in y and x directions. The keypoint detection algorithm works on each patch separately, so it is forced to look for keypoints in regions with low contrast as well. If the result is bad, it is a good idea to try again with a different patch grid size.
  The user can specify the maximal number of keypoints per patch, and the fraction of "good" keypoints which are selected for computing the transformation.
  In principal, the computation of the homography transformation requires four keypoint pairs only, so usually the problem is over-determined, and the solution is computed as a best fit. For the fitting, the user can choose between two weighting methods. "Least median of Squares" typically gives satisfactory results.
- If everything works well, the rigid transformation should leave registration errors of a few pixels only. They are removed in the second phase with a "dense optical flow" algorithm. Several parameters can be modified to control this process. For a detailed explanation of those parameters, please refer to the OpenCV documentation.
- Two checkboxes can be used to skip one of the above registration phases. If the first phase is skipped, the color input image must match the shape (pixel numbers) of the B/W image, and both images must be registered up to a few pixels already.

Detailed tooltips are provided for all parameters. They appear as soon as the user moves the mouse over a parameter description.

The user can press the button "Restore standard values" when after experiments with different parameter settings he or she wants to return to the (recommended) standard values.

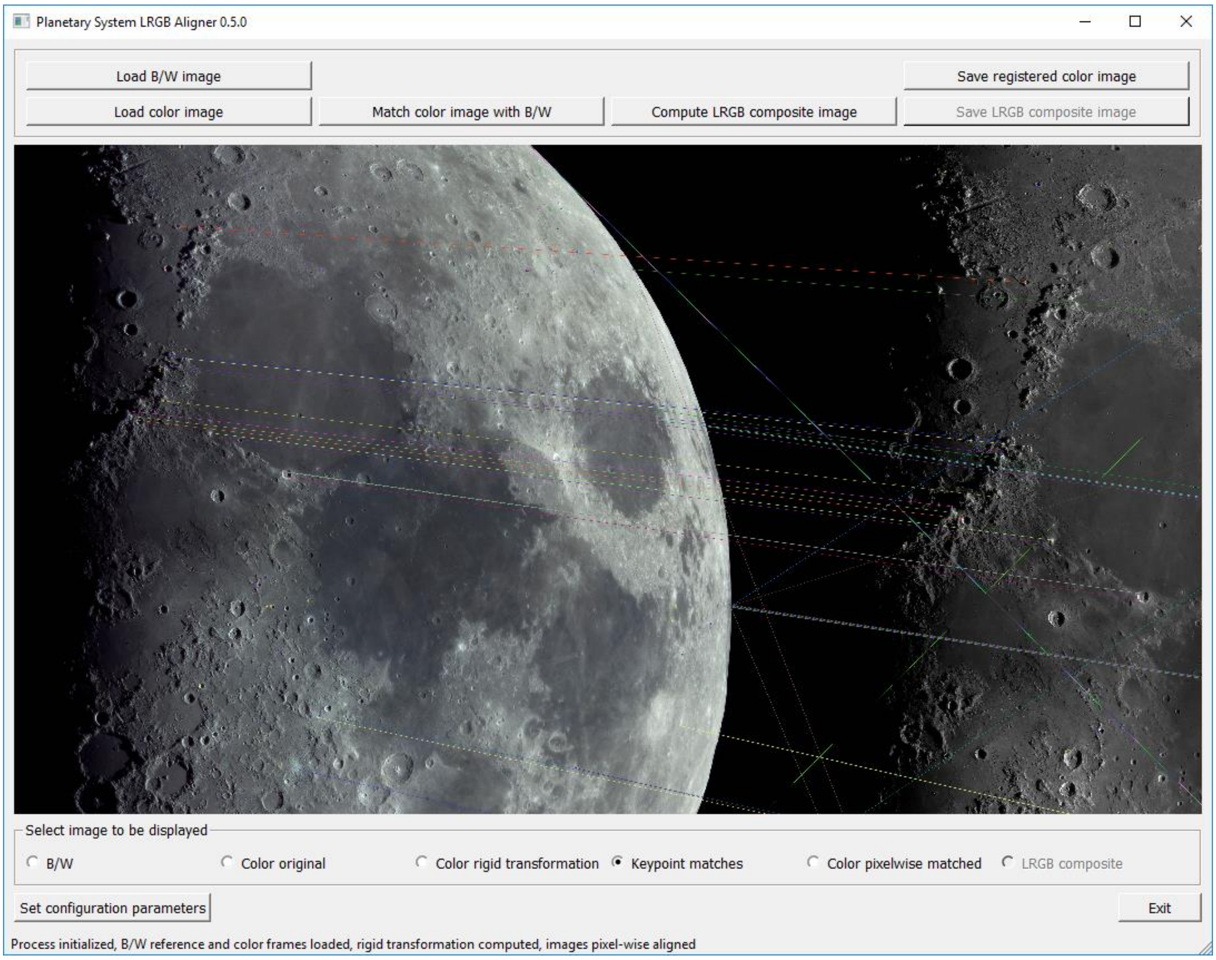
4.4 Starting the registration process



Once the input images are loaded and parameters are selected, the user starts the computation by pressing "Match color image with B/W". Since the computation takes place in the background (as shown by the word "busy" in the status line), the user can still use the image viewer to inspect the input images.

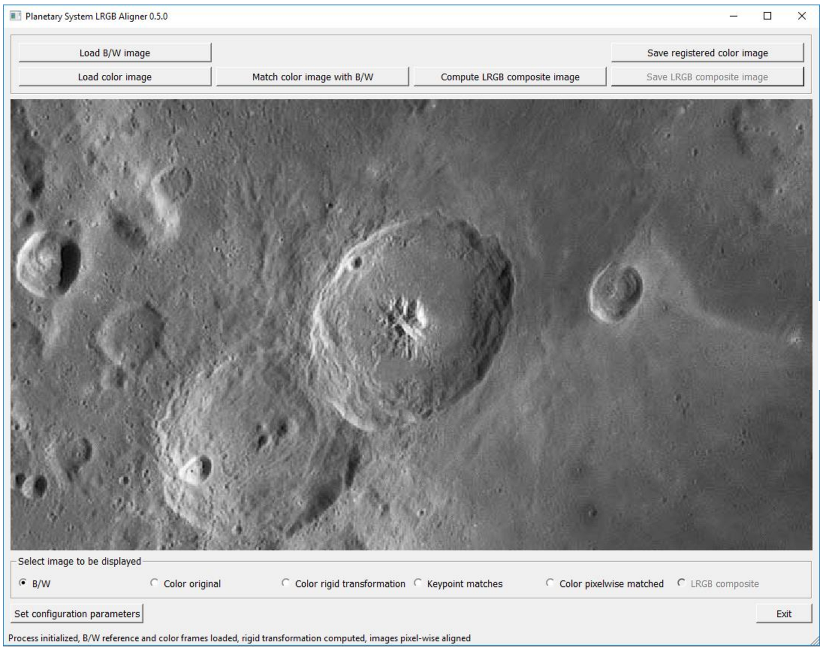
4.5 Inspecting the registration quality

As soon as the result of the first registration phase becomes available, the corresponding radio buttons under the image viewer are activated. "Color rigid transformation" displays the image resulting from the first registration phase. By switching back and forth between this image and the B/W input, the registration accuracy of the first phase can be checked. The next radio button "Keypoint matches" shows the two input images together with the keypoint pairs used in the homography computation. This illustration may help to find out if keypoint pairs are reasonably well distributed across the image.

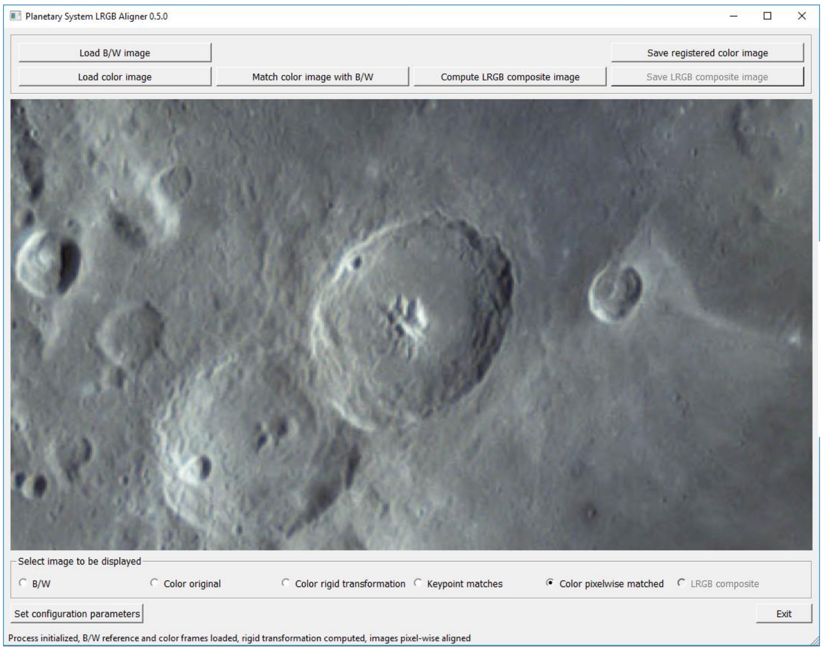


When the second phase is finished, the result of the optical flow algorithm can be inspected by pressing the "Color pixelwise matched" button.

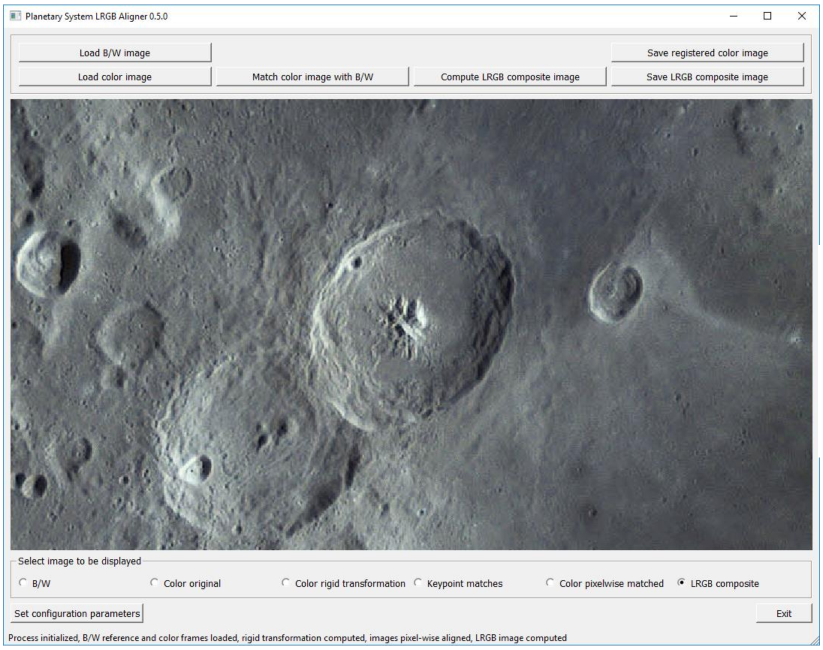
After both registration phases are finished, the image viewer can be set at a high zoom rate to check the registration quality in different image locations. Here is an example view:



Original B/W image



Color image after second registration phase. Please note that the resolution is much lower than in the B/W case.

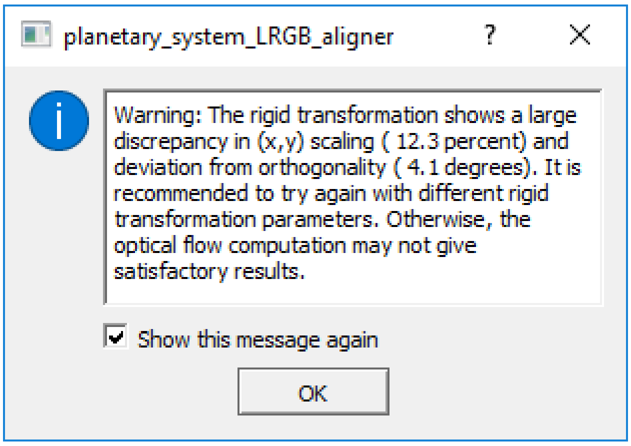


LRGB composite (see section 4.7). The color info of the second input image is combined with the resolution of the first image.

When switching images, surface features should stay in the same location.

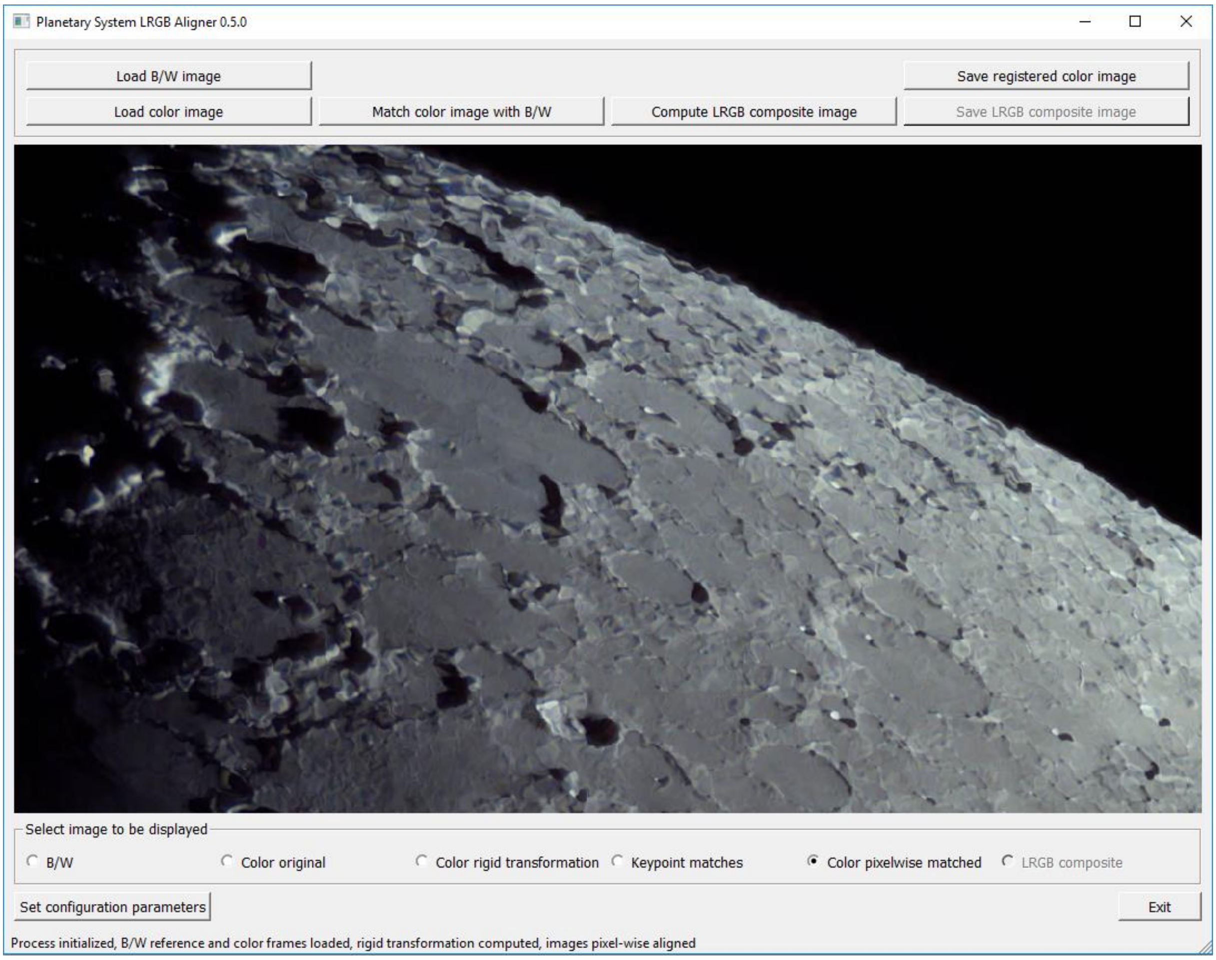
4.6 What if the registration failed?

If the rigid transformation shows a high degree of scaling anisotropy or does not preserve angles well, a warning appears as in this illustration.



If this warning appears it is unlikely that the optical flow algorithm (second phase) can compensate the low precision of the first phase. In this case it is recommended to change parameters (e.g. increase the number of (x,y) patches) and to try again.

What happens if the second registration phase continues after an unsuccessful rigid transformation can be seen in the illustration below.



Crater rims and the limb contour are strongly distorted. Also, as can be checked by alternating between the radio buttons “B/W” and “Color pixelwise matched”, surface details jump back and forth as a result of bad registration quality.

4.7 LRGB image composition

Once the image registration has finished successfully, pressing "Compute LRGB composite image" starts the combination of the B/W and the transformed color images into an LRGB composite. Once this is done, the result can be inspected by pressing the radio button "LRGB composite". This image should match the B/W image in resolution, and it should not display noticeable color artefacts.

4.8 Saving the resulting images

The (pixelwise) registered color image can be saved to disk by pressing "Save registered color image". It is recommended to use the (uncompressed) "tif" file format. This image file can be loaded together with the original B/W image as two layers in a program such as Adobe Photoshop for manual image processing. This way the user has full control of the LRGB combination process.

If PlanetarySystemLRGBAligner was used to compute the LRGB composite automatically, the resulting image can be saved to disk by pressing "Save LRGB composite image". Again, it is recommended to choose the "tif" file format to retain the full dynamic range and to avoid compression artefacts.