

# Geometric calibration of Colour and Stereo Surface Imaging System of ESA’s Trace Gas Orbiter

Stepan Tulyakov, Anton Ivanov

*Lausanne, Switzerland*

---

## Abstract

There are many geometric calibration methods for “standard” cameras. These methods however can’t be used for calibration of telescopes with large focal length and complex off-axis optic. Moreover, specialized calibration methods for telescopes, unfortunately, are not available in literature. This motivated our paper, in which we describe telescope calibration method that we developed for Colour and Stereo Surface Imaging System (CaSSIS) telescope on board of ExoMars Trace Gas Orbiter (TGO). Although our method is described in the context of CaSSIS, with camera-specific experiment, it is general and can be applied to other telescopes. We further encourage re-use of the proposed method by making our calibration code and data available on-line.

*Keywords:* geometric calibration, lens distortion, off-axis, star-field, CaSSIS, TGO, telescope, rational distortion model, telephoto

---

## 1. Introduction

On March 15, 2016 ESA’s Trace Gas Orbiter (TGO) was launched to Mars, as a part of ExoMars project. It’s aim is to find trace gases - evidences of geological or biological activity on Mars. Color and Stereo Surface Imaging System (CaSSIS) is a TGO’s imaging system that provides visual context for sites, identified as potential sources of the trace gases.

CaSSIS[20] is a multi-spectral pushframe camera[8] with 4 rectangular color filters covering it’s sensor. When spacecraft is moving along the orbit, each part of filmed location become visible in each filter at different time. By acquiring and mosaicking multiple images acquired at different times CaSSIS is able to reconstruct large color image of filmed location.

CaSSIS is also a stereo camera. It is capable of acquiring two images of a target area from two distinct points on the same orbit. While approaching the target area it acquires the first image, when it gets mechanically rotated and acquires the second image, while distancing from the target area. By combining two images CaSSIS is able to compute Digital Elevation Model (DEM) of a target area.

To compute scientific products, such as color images and DEMs from raw CaSSIS images, one needs geometric camera parameters, such as focal length and lens distortion model. While their nominal values are known from technical specification, their actual values might be different from the nominal due to imprecise manufacturing or mounting of the camera. Therefore, nominal values must be validated and refined. This is the main goal of geometric calibration.

There are many geometric calibration methods [11, p178-193][22, 12, 21] and tools [2, 5, 4] for “standard” cameras. However, these off-the-shelf tools can’t be used for calibration of telescopes, such as CaSSIS, for two reasons. Firstly, most of off-the-shelf tools require images of calibration targets, such as checkerboard chart. For telescopes with large focal length, however, such targets must be very large ( $\approx km^2$ ) and should be placed very far away from the telescope ( $\approx 10 km$ ), which is impractical. Secondly, telescopes often have off-axis optic with complex lens distortion, that can’t be handled by off-the-shelf tools. Therefore, there is a need for specialized calibration methods, which, unfortunately, are not available in literature.

This motivated our paper, in which we describe telescope calibration method that we developed for Colour and Stereo Surface Imaging System (CaSSIS) telescope on board of ExoMars Trace Gas Orbiter (TGO). Although our method is described in the context of CaSSIS, with camera-specific experiment, it is general and can be applied to other telescopes. We further encourage re-use of the proposed method by making our calibration code and data available on-line<sup>1</sup>.

The rest of this paper has the following structure. In § 2 we discuss related works and in § 3 we describe geometric camera model, adopted in the paper. Next, in § 4 we explain distortion model selection procedure, based on lens simulation. In § 5 we describe on-ground calibration using images of dotted calibration target captured through collimator. Next, in § 6 we

---

<sup>1</sup><https://github.com/eSpaceEPFL/CASSISgeometry.git>

describe in-flight calibration using star field images. Finally, in § 7 we show how refined geometric parameters improve quality of map projected CaSSIS images.

## 2. Related Work

### 2.1. Lens distortion models

Off-the-shelf calibration tools typically assume radial or brown-conrady lens distortion model. The radial [11, p189-193] is a simplest 5 degree-of-freedom (DOF) model, only accounting for radially symmetric distortion. Brown-Conrandy [9] is a more complex model with 7 DOF, that in addition to radially symmetric component accounts for tangential decentring. These models, however, can not represent complex distortion in camera with off-axis optic, such as CaSSIS, as we show in § 4. Complex distortion are better modeled by bi-cubic [14] or rational model [10] with 20 and 18 DOF respectively. In our work we adopted rational distortion model, that we discuss in § 3.

### 2.2. Star field calibration

For geometric calibration of a camera one needs images of calibration target - object with known 3D coordinates. Since angular positions of stars are well known and documented in star catalogs [7], such as MASS2 and Tycho2, star field can become a perfect calibration target. Indeed, star field calibration is widely used in star trackers, that are integral part of every spacecraft [18, 17, 13]. Star field calibration also can be used for one shoot calibration of consumer level cameras [15]. Unfortunately, all known star field based calibration methods assume simplistic lens distortion model, and therefore, can't applied for telescope calibration. Before stars from a image can be used for calibration, they should be identified with the stars from a star catalog. Fortunately, this task is automated [16] and library is available on-line [1].

## 3. CaSSIS camera model

The camera model consists of: (1) intrinsic model, (3) rational lens distortion model and (3) extrinsic model. In this section we discuss each part of camera model in details.

### 3.1. Intrinsic Model

Intrinsic model [11, p153-158] describes transformation from 3D camera frame coordinates  $\mathbf{X} = \{X, Y, Z\}$  to 2D image coordinates  $\mathbf{x} = \{x, y\}$  as follows:

$$(x, y) = \left( \frac{\mathbf{K}_1^T \mathbf{X}}{\mathbf{K}_3^T \mathbf{X}}, \frac{\mathbf{K}_2^T \mathbf{X}}{\mathbf{K}_3^T \mathbf{X}} \right), \quad \mathbf{K} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where  $f$  is a focal length of camera, measured in pixels, and  $x_0, y_0$  are coordinates of a principal point in the image. In case of CaSSIS we assume that  $x_0$  and  $y_0$  correspond to center of an image. Therefore, CaSSIS intrinsic model has only 1 DOF.

### 3.2. Rational lens distortion model

Intrinsic camera model is complemented with a lens distortion model, that describes transformation from distorted image coordinates  $\mathbf{i} = (i, j)$  to ideal image coordinates  $\mathbf{x} = (x, y)$ . We use rational distortion model [10]:

$$(x, y) = \left( \frac{\mathbf{A}_1^T \chi_6}{\mathbf{A}_3^T \chi_6}, \frac{\mathbf{A}_2^T \chi_6}{\mathbf{A}_3^T \chi_6} \right), \quad \chi_6 = [i^2 \ i j \ j^2 \ i \ j \ 1]^T, \quad (2)$$

where  $\mathbf{A}_{1\dots 3}^T$  are rows of  $3 \times 6$  rational distortion matrix.

Interestingly, while not being invertible, rational model can very precisely represent inverse of itself [19]. We use this property and simultaneously estimate 2 rational models: one for distortion and another for correction.

### 3.3. Extrinsic Model

Extrinsic model[11, p155-156] describes transformation from reference (spacecraft) frame coordinates  $\mathbf{X}_{\text{ref}} = (X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}})$  to camera frame coordinates  $\mathbf{X} = (X, Y, Z)$  as follows

$$\mathbf{X} = \mathbf{R} \mathbf{X}_{\text{ref}} + \mathbf{T}, \quad (3)$$

were  $\mathbf{R}$  is a  $3 \times 3$  rotation matrix and  $\mathbf{T}$  is a  $3 \times 1$  translation vector. The rotation matrix  $\mathbf{R}$  is a function of 3 Euler angles  $\mathbf{R} = F(\alpha, \beta, \gamma)$ . The translation vector is typically ignored, since camera is much closer to the reference frame, than to the scene. The extrinsic model has 3 DOF in total.

## 4. Distortion model selection

Before CaSSIS became available, we were provided with lens distortion data, shown in Table 1, computed using ray-tracing simulation. To find out what distortion model better represent CaSSIS lens distortion, we fitted radial, brown-conrady, rational and bi-cubic lens distortion models (see § 2.1) to the data and compared average euclidean error of the models using leave-one-out cross-validation.

	$x$ , [mm]	$i$ , [mm]	$y$ , [mm]	$j$ , [mm]
1	0	0	0	0
2	0	0	-3.3911	-3.3846
3	0	0	-6.7437	-6.7538
4	0	0	3.4094	3.3846
5	0	0	6.8165	6.7538
6	-5.1358	-5.1385	0.0022	0
7	-5.1207	-5.1385	-3.3866	-3.3846
8	-5.1029	-5.1385	-6.737	-6.7538
9	-5.1478	-5.1385	3.4093	3.3846
10	-5.1568	-5.1385	6.8142	6.7538
11	-10.2482	-10.2769	0.0089	0
12	-10.2183	-10.2769	-3.3733	-3.3846
13	-10.2133	-10.3077	-6.7171	-6.7538
14	-10.2722	-10.2769	3.4094	3.3846
15	-10.2901	-10.2769	6.8075	6.7538
16	5.1358	5.1385	0.0022	0
17	5.1207	5.1385	-3.3866	-3.3846
18	5.1029	5.1385	-6.737	-6.7538
19	5.1478	5.1385	3.4093	3.3846
20	5.1568	5.1385	6.8142	6.7538
21	10.2482	10.2769	0.0089	0
22	10.2183	10.2769	-3.3733	-3.3846
23	10.183	10.2769	-6.7173	-6.7538
24	10.2722	10.2769	3.4094	3.3846
25	10.2901	10.2769	6.8075	6.7538

Table 1: CaSSIS lens distortion data, computed using ray-tracing simulation.  $x$ ,  $y$  are ideal image coordinates and  $i$ ,  $j$  are distorted image coordinates given relative to image center.

Resulting distortion fields and errors for each model are shown in Figure 1. Clearly, simple radial and brown-conrady models with more than 1 pixel error failed to represent CaSSIS distortion, while bi-cubic and rational models with less than 0.1 pixel error performed well. Therefore we decided to use rational model with CaSSIS.

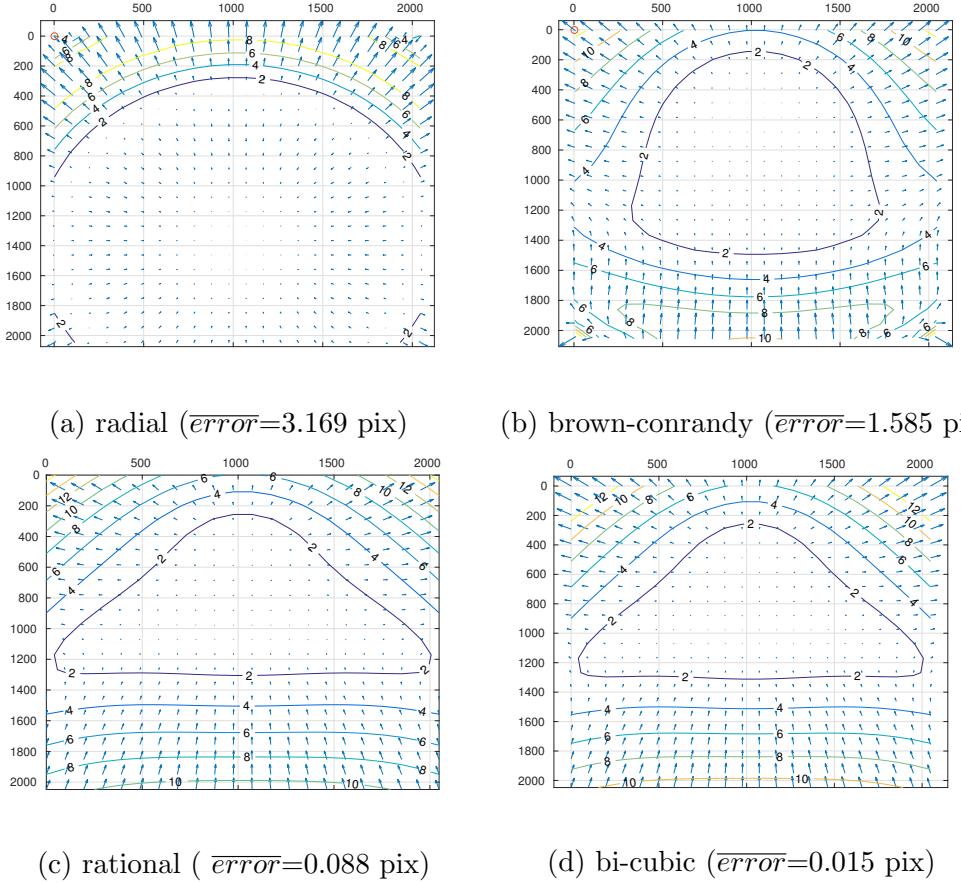


Figure 1: Distortion models “fitted” to simulated CaSSIS lens distortion data. Vectors show transformation from distorted to ideal image. Contour lines show magnitude of this transformation. Errors are average euclidean distances between position of ideal pixel predicted by the model and their actual position. Note, that simple radial (1a) and brown-conrady models (1b) with more than 1 pixel error fail to represent CaSSIS distortion, while bi-cubic (1d) and rational (1c) models with less than 0.1 pixels error both perform well.

## 5. On-ground calibration

After CaSSIS camera became available, we attempted to estimate distortion model from image of dotted calibration target, as in [10]. Since focal length of CaSSIS is too large to acquire in-focus images of the target from a reasonable distance, we used setting with collimator, shown in Figure 3.

After image was acquired, we applied adaptive thresholding and connected components detection methods to find dots in the image. Then, we found dots' centers using center-of-mass method. Finally, we fitted the regular rectangular grid to the dots centers, using simple greedy algorithm that starts from arbitrary selected dot and expands the grid in horizontal and vertical directions, until no new dots can be added to the grid.

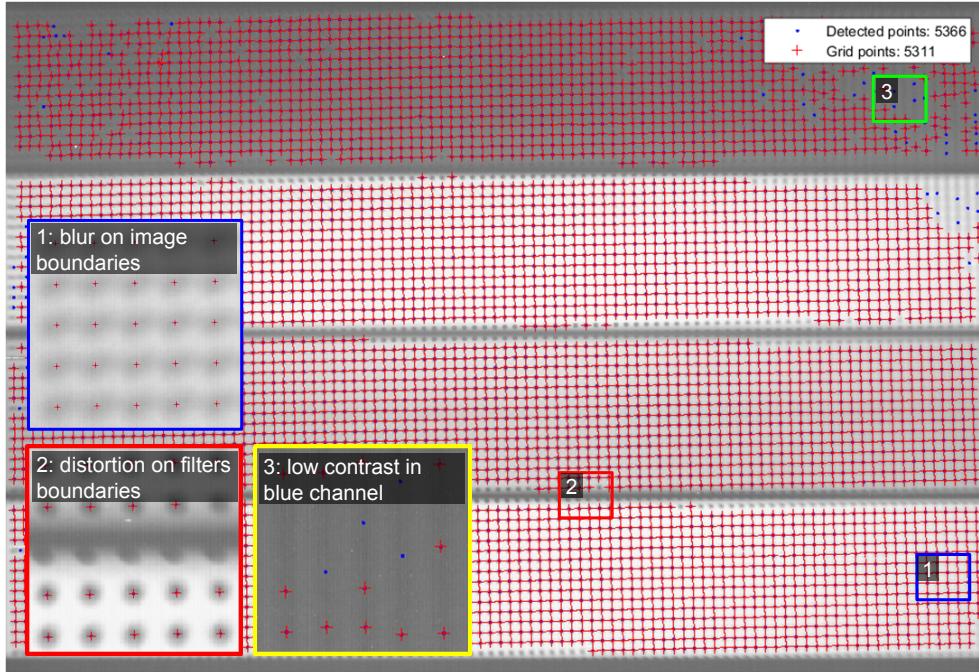


Figure 2: Image of the dotted target overlaid with the fitted grid. Red crosses show dots that were added to the grid. Blue points show dots, that were not added to the grid.

The acquired image with fitted grid is shown in Figure 2. Analysis of the grid confirmed presence of small lens distortion in the image: the grid rows and columns appeared not as a straight lines, but as a high-order curves. However, we failed to estimate distortion field resembling Figure 1c using

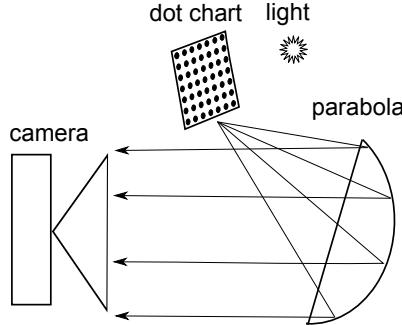


Figure 3: On-ground calibration setting. To acquire in-focus image of dotted calibration target from reasonable distance, we put it in focus of parabolic collimator.

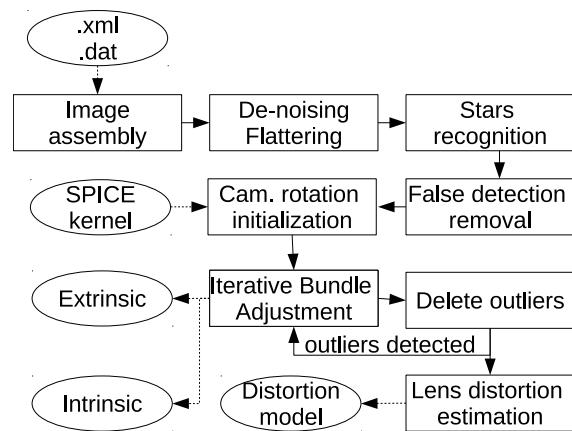


Figure 4: Work flow of in-flight calibration. Ellipses show input and output data and rectangles show processing stages.

the grid. This is probably due to the fact that experimental data was contaminated with unknown distortion coming from the collimator.

## 6. In-flight calibration

During TGO commissioning and mid-cruise checkout CaSSIS acquired multiple images of star fields, that we used for in-flight calibration. In § 6.1 we describe our in-flight calibration method and in § 6.2 we show calibration results.

### 6.1. Method

Overall work flow of in-flight calibration is shown in Figure 4 and each individual procedure is described below.

*Image assembly.* First of all, we assemble full-sensor images from several data packets according to information in XML files, supplied with each data packet.

*De-noising and flattening.* Next, we denoise images by subtracting median of several images “looking” at different star fields from every image. This procedure helps us to get rid of fixed-pattern noise and hot pixels. Then we flatten each image by applying Difference-of-Gaussian (DoG) filter.

*Star-field recognition.* Next, we perform star detection and recognition using open-source Astrometry.net library[1] and 2MASS star catalog. The library takes an image of a star field as an input and outputs  $(x, y)$  coordinates of stars in the image and their corresponding  $(Ra, Dec)$  coordinates in equatorial frame J2000.

*False detections removal.* Then, we collect information about detected stars from all images and filter out erroneous detections using simple heuristic. Since we know that calibration image sets consists of a sequences of 3-4 almost identical images, we mark star as a false detection, if it is not re-detected in a similar position in at least 2 images.

*Camera rotation initialization.* On this stage we find camera rotation for every image independently. During the estimation, we set focal length of the camera to nominal and search for camera rotation that minimizes projection error - euclidean distance between observed and predicted star positions in each image individually. The optimization is performed by nonlinear least square method (lsqnonlin in MATLAB). We initialize the optimization with rotation angles from SPICE kernel<sup>2</sup>, available online [3, 6].

*Iterative Bundle Adjustment (BA).* On this stage we search for refined focal length and rotations that minimize projection error for all images simultaneously. The optimization is performed by nonlinear least square method (lsqnonlin in MATLAB). We initialize the optimization with focal length and rotation matrices that we found on the previous step. After each BA iteration, stars that have large residual projection errors, compare to their neighbors are rejected as outliers and BA is performed again, until no new outliers is found. Without this outliers rejection procedure subsequent lens distortion estimation fails.

*Rational lens distortion estimation.* On this stage we “freeze” intrinsic and extrinsic camera models and search for rational lens distortion model that minimizes remaining projection error. The optimization is performed by nonlinear least square method (lsqnonlin in MATLAB). We initialize the optimization process using ”no distortions” hypothesis.

## 6.2. Results

---

<sup>2</sup>SPICE kernels contain information about orientation and position of spacecraft and its elements, received from its sensors for any moment in time.

We performed our experiments on 3 datasets: “mcc motor” and “pointing cassis” acquired on July 2016 during mid-cruise checkout and “commissioning2”, acquired on April 2016 during near-Earth commissioning. We selected these datasets, since they contain images of dense star fields acquired with long 1.92 seconds exposure. We estimated camera parameters using combined “mc motor” and “pointing cassis” set, which we called training set, and validated results on “commissioning2” set, which we called validation set.

Set Name	No. images	No. detected stars
Commissioning 2	12	670
Mcc motor	92	2573
Pointing CaSSIS	45	539

Table 2: Datasets summary. Note that all calibration sets consist of sequences of 3-4 almost identical images, acquired within short time interval.

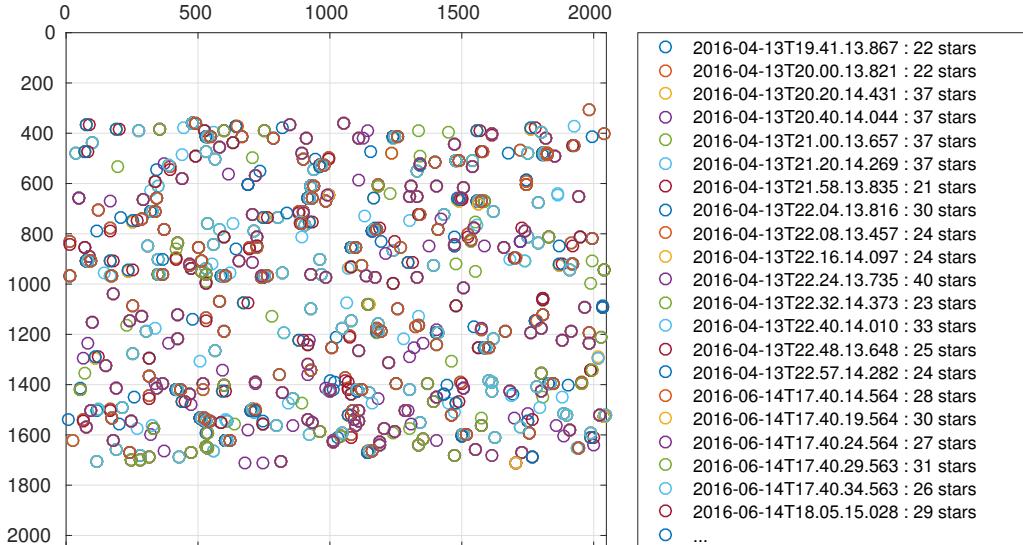


Figure 5: Position of all stars detected in combined “mcc motor” + “pointing CaSSIS” set on the image sensor. Note, that the detector is almost uniformly filled.

Number of images and recognized stars in every set is shown in Table 2. As shown in Figure 5 stars from the training set cover sensor densely and uniformly, allowing for lens distortion estimation.

Using stars, detected in the training set, we refined camera rotations obtained from SPICE kernels for every image individually, while keeping focal length of camera fixed to nominal. By refining the rotations, we reduced median euclidean distance between observed and predicted star position in

training set images from 147.41 to 3.42 pixels.

Next, we used estimated camera rotations and nominal focal length to initialize iterative bundle adjustment process, that refined camera rotations and focal length using all images simultaneously, while ignoring lens distortion. Iterative bundle adjustment process converged after 4 iterations. The effect of iterative outliers rejection scheme is shown in Figure 6. Note that after the 1<sup>st</sup> iteration, BA residuals contain gross outliers, while after the last iteration residuals form clear spatial pattern, suggesting presence of lens distortion. BA reduced average euclidean distance between observed and predicted star positions in training set images from 3.56 to 2.56 pixels. The refined focal length found by BA is 875.93 mm, that is slightly shorter than nominal focal length equal to 880 mm.

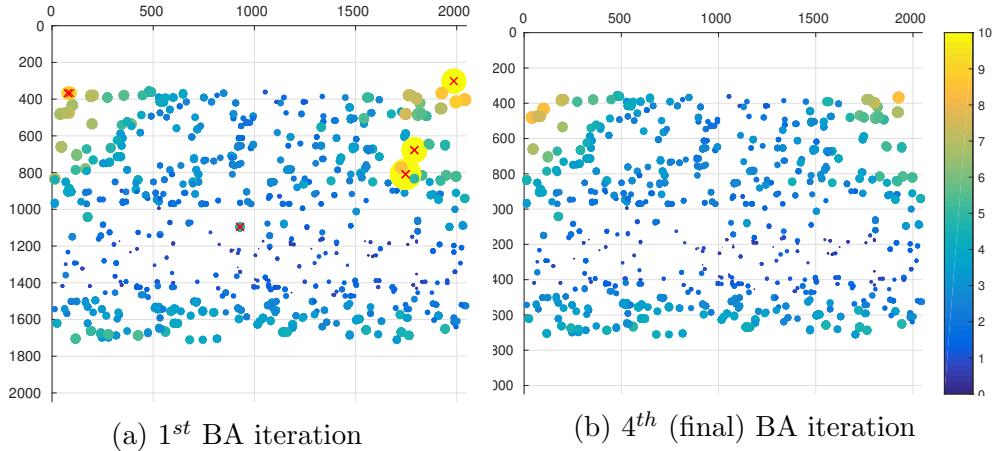


Figure 6: Residuals after 1<sup>st</sup> and 4<sup>th</sup> BA iterations. Color coding shows actual scale of the residuals. Crossed out residuals correspond to the identified outliers. In top and bottom part of the sensor we don't have observations, since they are covered by nontransparent mask. Note that after the 1<sup>st</sup> iteration (6a), the residuals contain gross outliers, while after the 4<sup>th</sup> iteration (6b) residuals form clear spatial pattern, suggesting presence of lens distortion. Average euclidean error before BA is 3.65 pixels, after 1<sup>st</sup> iteration is 2.78 pixels and after the 4<sup>th</sup> iteration is 2.56 pixels.

Then, we “freezed” focal length and camera rotations and estimated rational distortion model. The estimated distortion field is shown in Figure 7. Note, that it’s shape resembles distortion field obtained by fitting the rational model to lens simulation data, shown in Figure 1c. The difference is probably caused by relative shift of the lens with respect to the sensor.

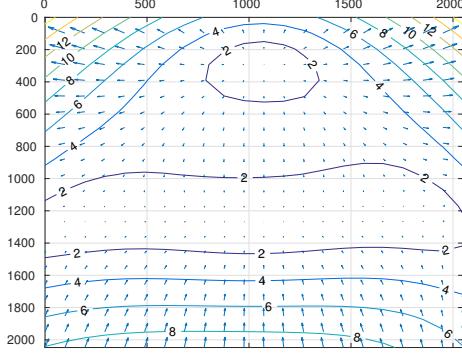


Figure 7: Distortion field estimated using star-field images. Vectors show transformation from distorted to ideal image. Contours show magnitude of the transformation. Note, that this distortion field resembles distortion field from Figure 1c obtained by fitting rational model to lens simulation data. The difference is probably caused by shift of the lens with respect to the sensor.

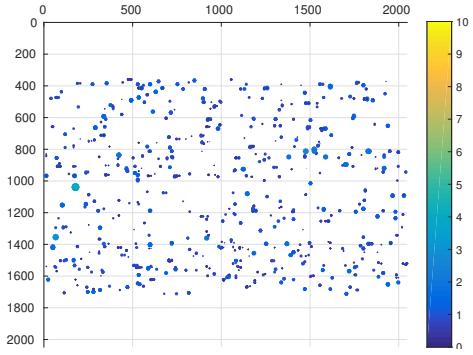


Figure 8: Residual errors in pixels after lens distortion estimation. Average error is 0.66 pixels. Color coding shows actual error scale, that is similar to Figure 6b. Note, that the errors are small and spatially uniform than compared to the residual errors after BA from Figure 6b. This suggests that they come from inaccurate star detection.

Parameters of the estimated distortion model are shown in Table 3. Distortion model fitting reduced average euclidean distance between observed and predicted star positions in training set images from 2.54 to 0.66 pixels. Moreover, as shown in

Figure 8, residuals after fitting the lens distortion model became spatially uniform and small when compared to the bundle adjustment residuals from Figure 6b. This suggests that the remaining residual errors probably come from inaccurate star detection.

Finally, computed error of estimated camera model on separate validation set, while effectively ignoring extrinsic model. With the refined camera model average projection error is 0.47 pixels, while with the nominal camera model the error is 3.56 pixels. This result suggests that our geometric calibration results are valid.

A11	A12	A13	A14	A15	A16
0.0643	0.4091	-0.0011	1.0003	0.0003	-0.0000
A21	A22	A23	A24	A25	A26
-0.0043	0.0635	0.4065	0.0002	0.9952	0.0004
A31	A32	A33	A34	A35	A36
-0.0501	0.0071	-0.0305	0.0636	0.4401	1.0000

Table 3: Parameters of rational distortion model from § 3.2., estimated using star field images.

## 7. Color image experiment

During orbit insertion CaSSIS captured several color images of Mars. To check effectiveness of our calibration we map projected these images using nominal and refined geometric models and compared quality of resulting images. In this section in § 7.1 we describe work flow of our color image experiment and in § 7.2 we discuss the results.

### 7.1. Method

The work flow of color image experiment is shown in Figure 9, with each individual procedure described below.

First of all, given folder with CaSSIS data packets, we find all data packets belonging to particular sequences and color band. Next, we correct lens distortion in every data packet. Then, we convert each data packet to ISIS “.cub” format (*cassis2isis*) and add information from SPICE kernel to each “.cub” (*spiceinit*). After that, we project all “.cub” that correspond to single band of a image sequence into sinusoidal map (*cam2map*), while keeping resolution of the projections consistent. Next, we mosaic all projected “.cubs” into one image, corresponding to single band of a sequence (*automos*). We repeat process described above for every band. After that, we select one of the bands as a reference and match map-projections of all other bands to the reference (*map2map*). This is required, since “by default” map projection of every band has its own resolution and coordinate limits. Finally, we combine individual color bands into multi-band cube (*cubeit*).

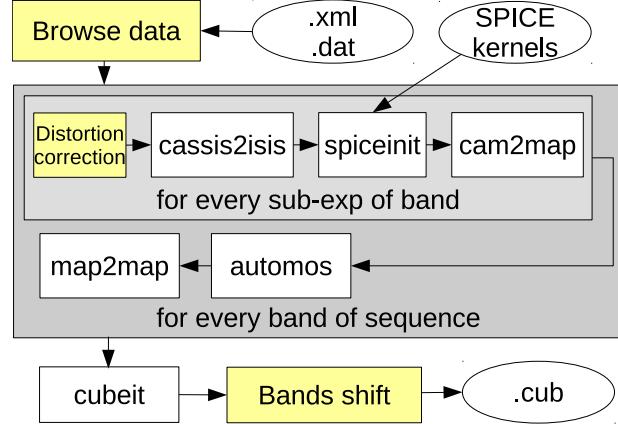
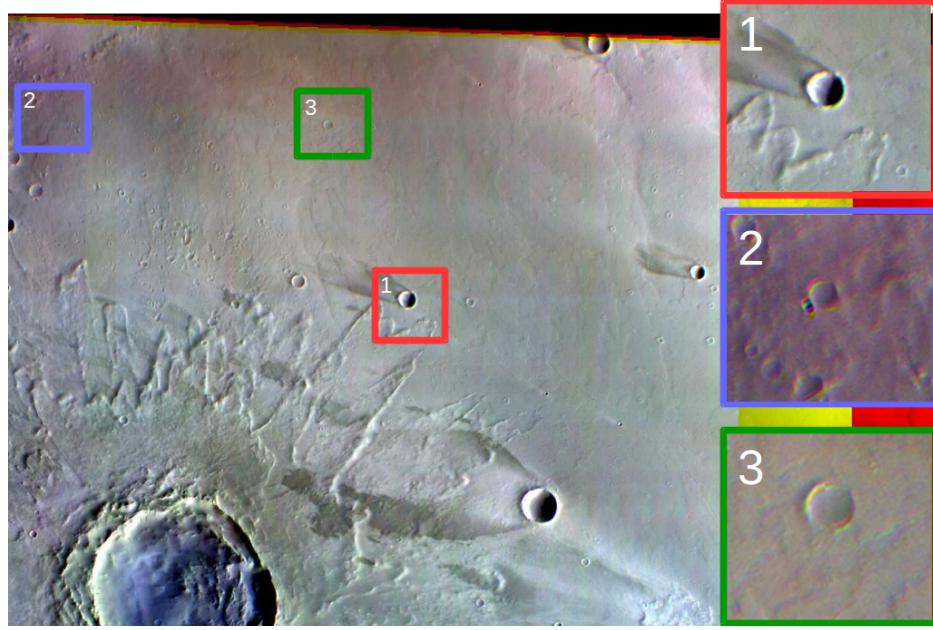


Figure 9: Work flow of color image experiment. Ellipses show data. White rectangular boxes show standard ISIS functions, while yellow boxes show scripts implemented in Python.

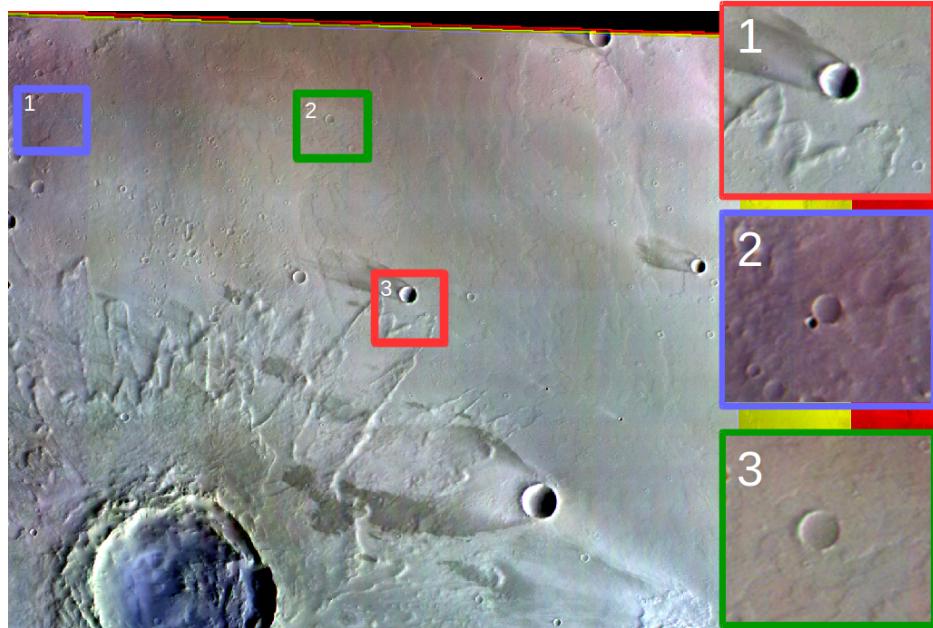
## *7.2. Results*

During our experiments we noticed that map-projected images of individual color bands are misaligned along the track by 1-10 pixels, depending on the sequence. This fact, probably caused by camera pointing error, needs to be properly investigated. Meanwhile we decided to check validity of lens distortion model, by verifying that color band images are distortion free. For that we compensated color band misalignment by simple shift and compared color band images.

Results of this comparison are shown in Figure 10. As seen from the figure, when we use nominal camera parameters, the projected image (10a) has color fringes (close-up #1,2 and 3) and stitching artifacts (close-up #3), whereas when we use refined parameters, the projected image (10b) is almost perfect. This confirms that developed calibration method works and improves quality of final scientific products, such as color images.



(a) Nominal parameters



(b) Refined parameters

Figure 10: Map projection of CaSSIS image. Images are shown in false colors. Color bands are aligned as described in §7.2. Note that, when we use nominal camera parameters, the projected image (10a) has color fringes (close-up #1,2 and 3) and stitching artifacts (close-up #3), while when we use refined parameters, the projected image (10b) is almost perfect.

- [1] Astrometry.net star recognition tool. <http://astrometry.net/>. Accessed: 2017-05-24.
- [2] Caltech camera calibration tool. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/). Accessed: 2017-05-23.
- [3] Exomars trace gas orbiter spice kernels. <https://naif.jpl.nasa.gov/pub/naif/EXOMARS2016>. Accessed: 2017-05-24.
- [4] Matlab camera calibration tool. <https://www.mathworks.com/videos/camera-calibration-with-matlab-81233.html>. Accessed: 2017-05-23.
- [5] Opencv camera calibration tool. [http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html). Accessed: 2017-05-23.
- [6] Spice toolkit. <https://naif.jpl.nasa.gov/naif/toolkit.html>. Accessed: 2017-05-24.
- [7] Visier star catalog library. <http://vizier.u-strasbg.fr/>. Accessed: 2017-05-24.
- [8] J. Anderson and M. Robinson. Challenges Utilizing Pushframe Camera Images. *40th Lunar and Planetary Science Conference*, 2009.
- [9] D. Brown. Decentering Distortion of Lenses. *Photometric Engineering*, 1966.
- [10] D. Claus and a.W. Fitzgibbon. A rational function lens distortion model for general cameras. *CVPR*, 2005.
- [11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [12] J. Heikkilä and O. Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. *CVPR*, 1997.
- [13] X. Junfeng, J. Wanshou, and G. Jianya. On-orbit Stellar Camera Calibration Based on Space Resectioning. pages 71–76, 2005.
- [14] E. Kilpel. Compensation of systematic errors of image and model coordinates. *Photogrammetria*, 1981.
- [15] A. Klaus, J. Bauer, K. Karner, P. Elbischger, R. Perko, and H. Bischof. Camera calibration from a single night sky image. *CVPR*, 2004.
- [16] D. Lang. *Astrometry.net: Blind astrometric calibration of arbitrary astronomical images*. PhD thesis, 2010.
- [17] M. Pal and M. S. Bhat. Autonomous Star Camera Calibration and Spacecraft Attitude Determination. *Journal of Intelligent & Robotic Systems*, 2014.
- [18] M. a. Samaan, T. Griffith, P. Singla, and J. L. Junkins. Autonomous on-orbit calibration of star trackers, 2001.

- [19] Z. Tang, R. V. Gooi, P. Monasse, and J. Morel. Self-consistency and universality of camera lens distortion models. 2012.
- [20] O. N. Thomas, G. Cremonese, M. Banaszkiewicz, J. Bridges, S. Byrne, V. Da Deppo, S. Debei, M. R. El-Maarry, E. Hauber, C. J. Hansen, A. Ivanov, W. Markiewicz, M. Massironi, A. S. McEwen, C. Okubo, P. Orlanski, A. Pommerol, P. Wajer, and J. Wray. The Color and Stereo Surface Imaging System (CaSSIS) for ESA’s Trace Gas. *Eighth International Conference on Mars (2014)*, 2014.
- [21] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 1987.
- [22] Z. Z. Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 1999.