Technical Report: Geometric calibration of CaSSIS using star-field images

Stepan Tulyakov, Anton Ivanov

Lausanne, Switzerland

Abstract

This report describes geometric calibration of CaSSIS camera of ExoMars Trace Gas Orbiter using star-field images.

1. Camera model

The geometric model of CaSSIS camera consists of following components:

1. Extrinsic model[1, p155-156]
2. Intrinsic model[1, p153-158]
3. Rational lens distortion model[2]

1.1. Extrinsic Model[1, p155-156]

Extrinsic model describes coordinate transformation from reference frame to camera frame. Having this model and coordinates of the 3D point in the reference frame $\mathbf{X_{ref}} = (X_{ref}, Y_{ref}, Z_{ref})$, we can compute point's coordinate in the camera frame $\mathbf{X} = (X, Y, Z)$ as following

$$\mathbf{X} = \mathbf{R}(\mathbf{X_{ref}} + \mathbf{t}), \tag{1}$$

were $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. The rotation matrix $\mathbf{R}$ is a matrix function of 3 Euler angles $\mathbf{R} = \mathbf{F}(\alpha, \beta, \gamma)$. Besides Euler angle and rotation matrix, there are other useful representations of 3D rotation such as angle-axis (Rodrigues) and quaternion representations. The extrinsic camera model has 6 degree-of-freedom (DOF) in total, however when dealing with star-field images we typically assume that translation of the camera is zero.

1.2. Intrinsic Model[1, p153-158]

Intrinsic model describes coordinate transformation from the 3D camera frame. Having this model and coordinates of the 3D point in the camera frame $\mathbf{X} = \{X, Y, Z\}$, we can compute it's coordinates in the 2D image frame $\mathbf{x} = \{x, y\}$ as following

$$(x, y) = \left(\frac{\mathbf{K}_1^T \mathbf{X}}{\mathbf{K}_3^T \mathbf{X}}, \frac{\mathbf{K}_2^T \mathbf{X}}{\mathbf{K}_3^T \mathbf{X}} \right), \tag{2}$$

where $\mathbf{K}_{1,2,3}^T$ are rows of 3×3 upper triangular camera calibration matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where f is focal length of the camera, measured in pixels, and x_0, y_0 are coordinates of a principal point in the image frame. The intrinsic camera model has 3 DOF in total.

1.3. Rational lens distortion model[2]

Intrinsic camera model is usually complemented with a lens distortion model, that describes transformation between undistorted (ideal) $\mathbf{x} = (x, y)$ and distorted (actual) image coordinates $\mathbf{i} = (i, j)$. To model distortions of CaSSIS lens we use Rational distortion model. This is an algebraic lens distortion model with 17 DOF, that can describe different complex lens distortion. Lets denote χ as 6×1 vector of "lifted" distorted image coordinates:

$$\chi = \begin{bmatrix} i^2 & ij & j^2 & i & j & 1 \end{bmatrix}^T \tag{4}$$

In this case, undistorted (ideal) image coordinates are computed as

$$(x, y) = \left(\frac{\mathbf{A}_1^T \chi}{\mathbf{A}_3^T \chi}, \frac{\mathbf{A}_2^T \chi}{\mathbf{A}_3^T \chi} \right), \tag{5}$$

where $\mathbf{A}_{1\ldots3}^T$ are rows of 6×3 rational distortion matrix.

2. Calibration Procedure

We use following consecutive steps to find parameters of geometric model of CaSSIS camera.

1. Raw image assembly
2. Dark-frame substraction
3. Star-fields recognition
4. False detection removal
5. Training and test set initialization
6. Camera rotation initialization
7. Bundle adjustment
8. Rational lens distortion estimation
9. Rotation sensor precision check

Raw image assembly. First of all, we assemble data-stripes into raw images, according to information in XML file supplied with each data-stripe.

Dark-frame substraction. After assembling the images, we flatten each raw image by subtracting from it it's median-filtered copy. Then we substract dark-frame from each flattened image. The dark-frame we compute as median of several flattened images. This procedure helps us to get rid of fixed-pattern noise and hot pixels.

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

False detection removal. Some of the detections are erroneous. These detections might significantly degrade the calibration result. Therefore, we design simple heuristic filtering process to detect and eliminate the false detections. We mark detected star as a correct, only if it is re-detected in a similar position in 3 consecutive frames close in time.

Training and test set initialization. On this stage, we randomly divide all data into training set, that is used for parameters estimation, and test set that is used for computing unbiased error of the estimated parameters.

[1] `http://astrometry.net/`

Camera rotation initialization. On this stage we estimate camera rotation for every image independently. During the estimation, we fix focal length and position of principal point of the camera to factory specs and search for camera rotation that minimizes projection error between observed and predicted star positions in each frame individually. To find the parameters we use nonlinear optimization with constraint (fmincon in MATLAB), multiple restarts and random initialization. The constraint forces L2 norm of rotation quaternion to be equal to one.

Bundle adjustment. On this stage we combine data from all images in one system of equations and search for focal length, position of principal point of the camera and rotation matrices that minimize projection error for all images simultaneously. We initialize focal length and position of principal point using factory specs and rotation matrices using values, that we found on the previous step. The optimization is performed by nonlinear least square method with search range constraints (lsqnonlin in MATLAB). To reduce effect of outliers on calibration result we use robust to outliers Huber loss instead of L2 loss.

Rational lens distortion estimation. On this stage we search for Rational lens distortion model that minimizes re-projection error, when applied "on the top" of already estimated intrinsic and extrinsic camera models. We initialize the optimization process using "no distortions" hypothesis. The optimization is performed by nonlinear least square method (lsqnonlin in MATLAB) with robust to outliers Huber loss.

Rotation sensor precision check. On this last stage, we check precision of rotation estimation based on counting steps of stepper motor. For that we compare image-based rotation estimations with corresponding steps counter-based estimations. Before preforming the comparison, we factor out systematic rotation error - constant rotation error between image-based and sensor-based rotation estimations. To quantify difference between the two rotation matrices we firstly compute additional rotation matrix which, when multiplied to the first rotation matrix, makes it similar to the second rotation matrix. Then, as a measure of the matrices difference, we use angle from angle-axis representation of this additional rotation matrix.

3. Results

With the help of Astrometry.net library we detected and recognized about 770 stars in Commissioning 2 dataset (other datasets are useless due to low density of star-fields). Out of these 770 stars 579 stars passed false detection removal procedure. As shown in Figure 1, these stars cover entire sensor's area fairly uniformly.

Next, we randomly assigned 97 percents of all stars from each image to training set and 3 percent to test set. After this we have 561 stars in the training set and 18 in the test set in total.

Next, using the detected stars, we estimated Euler angles of camera rotations for every frame, while keeping the focal length and the position of principal point of the camera fixed to factory specs. Results of this process are summarized in Table 1.

After this, we used these Euler angles of camera rotations together with factory specs for the focal length and the position of principal point to initialize the bundle adjustment process. As a result of the bundle adjustment, we got refined values for the Euler angels, the focal length and the position of principal point shown in Table 2 and Table 3 correspondingly. From the Table 2 it is clear that the estimated focal length and the position of principal point do not deviate too much from the factory specs. From the Table 3 it is evident that the camera rotation parameters also do not change too much from their initial values, shown in Table 1. In Figure 2 we show residual errors after the bundle adjustment. These errors have clear patter: they are small in image center, and large on image boundaries. This observation suggests presence of lens distortion.

Next, we fitted Rational distortion model to the residuals of distortion-free camera model. The parameters of the estimated model can be visualized as a distortion field shown in Figure 4. Note, that it resembles distortion field obtained by fitting rational model to the table of distorted-ideal points received from RUAG and calculated through ray tracing (Figure 5). Parameters of the estimated model together with residual errors are shown in Table 4. As shown in Figure 3, the residual errors after fitting the lens distortion model are uniform and small when compared to the bundle adjustment residuals in Figure 2). This suggests that the remaining residual errors probably come from inaccurate star detection.

Finally, we evaluated precision of camera rotation sensor. For that, we compared image-based rotation estimations with sensor based estimations.

Before comparing the matrices, we factored out constant systematic rotation error $\mathbf{R_{err}} = \begin{bmatrix} 1 & 0.0041 & -0.0009 \\ 0.0041 & 1 & 0.0023 \\ 0.0009 & -0.0023 & 1 \end{bmatrix}$. As we can see from column four of Table 5, the rotation error is small in zero position of the camera, but increases as we turn the camera. This suggests that there is an accumulating error, that increase as the camera rotates. Also note, that the error becomes smaller as time passes after the rotation. This suggests that there is small camera oscillation caused by the camera rotation that gradually fades with the time.

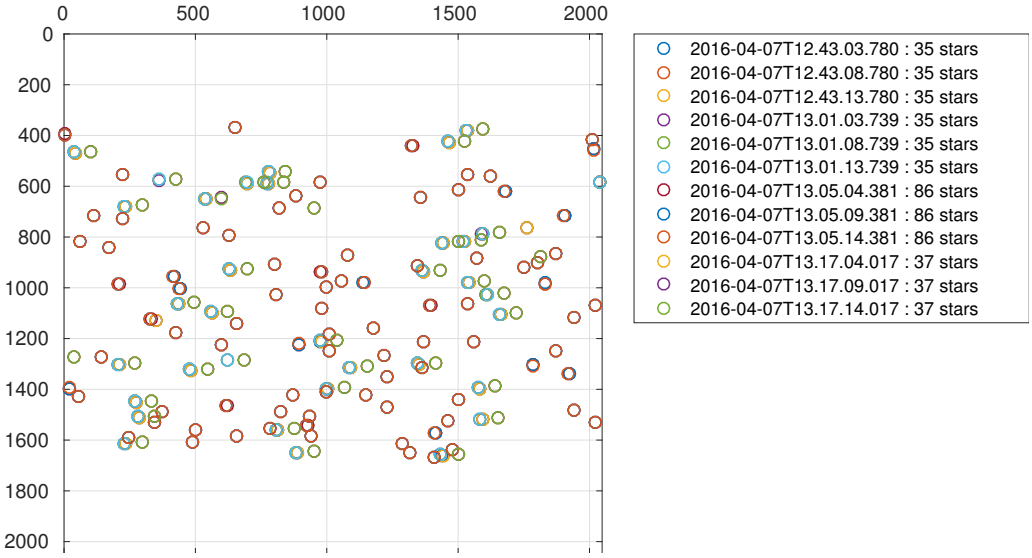

Figure 1: Position of all stars detected in Commissioning 2 dataset on the image sensor. Total number of stars is 579. Note, that the detector is almost uniformly filled.

Time	α_x, **[deg]**	α_y, **[deg]**	α_z, **[deg]**	**Train err, [pix]**	**Test err, [pix]**
2016-04-07T12.43.03.780	172.3635	-12.121	-72.9237	3.93	3.9
2016-04-07T12.43.08.780	172.3635	-12.121	-72.9187	3.93	3
2016-04-07T12.43.13.780	172.3634	-12.1209	-72.9215	3.92	3.07
2016-04-07T13.01.03.739	172.3664	-12.1235	-73.0413	3.85	3.61
2016-04-07T13.01.08.739	172.3662	-12.1234	-73.044	3.87	1.19
2016-04-07T13.01.13.739	172.366	-12.1234	-73.0516	3.76	4.84
2016-04-07T13.05.04.381	178.0875	7.0364	106.8476	3.9	1.45
2016-04-07T13.05.09.381	178.0874	7.0365	106.8529	3.88	2.23
2016-04-07T13.05.14.381	178.0872	7.0364	106.8507	3.85	1.37
2016-04-07T13.17.04.017	172.3232	-12.1133	-72.8759	3.4	5.29
2016-04-07T13.17.09.017	172.3232	-12.1133	-72.8863	3.44	4.88
2016-04-07T13.17.14.017	172.3232	-12.1132	-72.8818	3.37	8.26

Table 1: Euler angles of camera rotation found by rotation initialization procedure together with training and test set errors. Note, that the training and the test errors are large, since lens distortion is not yet taken into account and focal length and position of principal point of the camera are not yet refined.

Focal length, [mm]	**Principal point x0 [pix]**	**Principal point y0, [pix]**	**Train err, [pix]**	**Test err, [pix]**
875.2 (880)	1023.3 (1024)	1023.1 (1024)	2.6	2.46

Table 2: Focal length and position of principal point found by bundle adjustment procedure together with training and test set errors. Note, that the estimated values are very close to the "factory" specs shown in brackets. Note, also that the training and the test errors are large, since lens distortion is not yet taken into account, but they are smaller than after the rotation initialization stage (Figure 1)

Time	α_x, **[deg]**	α_y, **[deg]**	α_z, **[deg]**	**Train err, [pix]**	**Test err, [pix]**
2016-04-07T12.43.03.780	172.3633	-12.1203	-72.9279	2.6	2.46
2016-04-07T12.43.08.780	172.3633	-12.1203	-72.9113	2.6	2.46
2016-04-07T12.43.13.780	172.3633	-12.1201	-72.922	2.6	2.46
2016-04-07T13.01.03.739	172.3663	-12.1228	-73.0435	2.6	2.46
2016-04-07T13.01.08.739	172.3661	-12.1228	-73.0459	2.6	2.46
2016-04-07T13.01.13.739	172.3661	-12.1227	-73.0536	2.6	2.46
2016-04-07T13.05.04.381	178.0878	7.0354	106.8634	2.6	2.46
2016-04-07T13.05.09.381	178.0877	7.0355	106.8636	2.6	2.46
2016-04-07T13.05.14.381	178.0875	7.0354	106.8527	2.6	2.46
2016-04-07T13.17.04.017	172.3231	-12.1126	-72.8793	2.6	2.46
2016-04-07T13.17.09.017	172.3231	-12.1125	-72.8886	2.6	2.46
2016-04-07T13.17.14.017	172.3232	-12.1124	-72.8838	2.6	2.46

Table 3: Camera rotation parameters (Euler angles) found by bundle adjustment procedure together with training and test set errors. Note, that the Euler angles do not deviate much from the values estimated during the rotation initialization (Figure 1). The training and the test set errors are large, since lens distortion is not yet taken into account, but they are smaller than compared to errors after the rotation initialization stage (Figure 1)

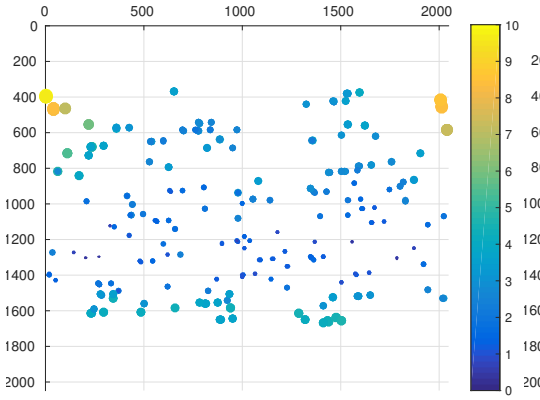

Figure 2: Residual errors in pixels after bundle adjustment. Average error is 2.6 pixels. Color coding shows actual error scale. Note, that the errors have clear patter: errors are small in the center and large on the borders of the image. This suggests presence of lens distortions.

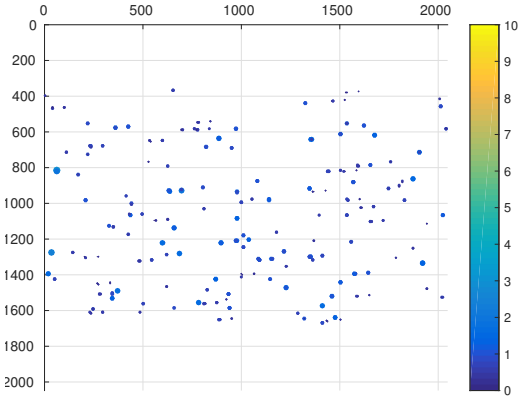

Figure 3: Residual errors in pixels after lens distortion estimation. Average error is 0.66 pixels. Color coding shows actual error scale. Note, that the errors do not have any pattern and they are small than compared to the residual errors after the bundle adjustment (Figure 2). This suggests that they come from inaccurate star detection.

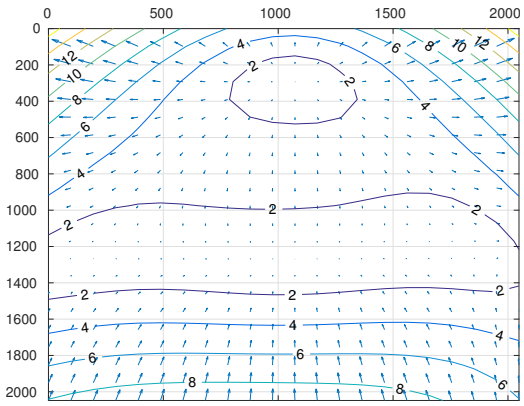

Figure 4: Distortion field estimated using star-field images. Vectors show transformation from distorted pixels to ideal pixels. Note, that it is very similar to the distortion field estimated using data obtained from RUAG (Figure 5).

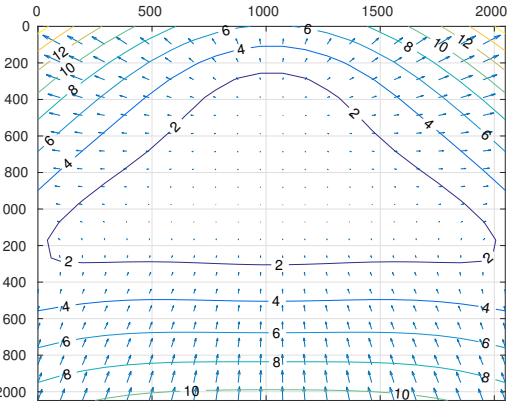

Figure 5: Distortion field estimated using table of distorted-ideal coordinates, received from RUAG. Vectors show transformation from distorted pixels to ideal pixels.

A11 - A16	**A21 - A26**	**A13 - A36**	**Train err, [pix]**	**Test err, [pix]**
-0.6106, -0.5445, -0.0015, 0.9998, 0.0002, 0	-0.0049, -0.607, -0.5433, -0.0005, 0.9943, 0.0005	-0.0496, -0.0257, -0.0703, -0.6092, -0.5101, 1	0.66	0.66

Table 4: Parameters of Rational lens distortion model estimated using star-field images. Note that we do not to show corresponding parameters of the model estimated using data received from RUAG, since it is hard to judge about similarity of models by just looking at models' parameters. In fact, different parameters can represent very similar distortions. Therefore, it is much more convenient to compare distortion fields shown in Figure 4 and 5. Note, that when lens distortions are taken into account, the training and the test errors are much lower than compared to the errors after the bundle adjustment shown in Table 2.

Time	**Rotation command, [deg]**	**Before systematic error compensation, [deg]**	**After systematic error compensation, [deg]**
2016-04-07T12.43.03.780	0	0.2945	0.021
2016-04-07T12.43.08.780	0	0.2803	0.0077
2016-04-07T12.43.13.780	0	0.2894	0.0156
2016-04-07T13.01.03.739	0.2	0.2301	0.0535
2016-04-07T13.01.08.739	0.2	0.2321	0.0511
2016-04-07T13.01.13.739	0.2	0.2383	0.0435
2016-04-07T13.05.04.381	180	0.2502	0.2428
2016-04-07T13.05.09.381	180	0.2501	0.2427
2016-04-07T13.05.14.381	180	0.2592	0.2417
2016-04-07T13.17.04.017	0	0.2725	0
2016-04-07T13.17.09.017	0	0.2805	0.0073
2016-04-07T13.17.14.017	0	0.2764	0

Table 5: Error between image-based and sensor-based rotation estimations before and after systematic error compensation. Looking at the last column we can notice that the larger is the rotation, the larger is the rotation error. This suggests that there is an accumulating error, that increase as the camera rotates. We also can notice that the error becomes smaller as time passes after the rotation. This suggests that there is small camera oscillation caused by the camera rotation that gradually fades with the time.

[1] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[2] D. Claus, a.W. Fitzgibbon, A rational function lens distortion model for general cameras, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) 1 (2005).