Technical Report and User Manual

– MRT Camera Calibration Toolbox –

Daniela Aguirre-Salazar, B. Sc.

Supervisors:
Sebastian Schramm, M. Sc.
E-Mail: *sebastian.schramm@mrt.uni-kassel.de*

Johannes Rangel, M. Sc.
E-Mail: *johannes.rangel@mrt.uni-kassel.de*

Version 1.0

University of Kassel
Mechanical Engineering
Institute for System Analytics and Control
Department of Measurement and Control
Univ.-Prof. Dr.-Ing. Andreas Kroll

Contents

1 General Information

1.1 System Overview

The MRT Camera Calibration Toolbox is an application, developed in Python using OpenCV and tkinter, which determines the parameters of a camera's perspective projection by performing an intrinsic and extrinsic geometric calibration. The application provides intrinsics, extrinsics and lens distortion parameters for each camera (two for stereo mode). For the stereo mode, the transformations between the individual camera coordinate systems are given as well. The information for each camera pose can be loaded using images or text files with the 2D points of the pattern. The calibration can be also made by using random subgroups from the total set of images. The computed parameters are the averaged over all iterations (subgroups), and both the final results and results per calibration can be exported to text files.

1.2 Organization of the Manual

This user's manual consists of four sections: *General Information*, *System Summary*, *Getting Started*, and *Using The System*. The *General Information* section explains in general terms the system and the purpose for which it is intended. The *Geometrical Camera Calibration* section explains the motivation of the program, background theory and possible applications (see section 2). The *System Summary* section provides a general overview of the system. The summary outlines the systems configuration and user access levels (see section 3). The *Getting Started* section explains how to get MRT Camera Calibration Toolbox and to install it on the computer. The section presents briefly system's menu (see section 4). The *Using The System* section provides a detailed description of system functions (see section 5).

2 Geometrical Camera Calibration

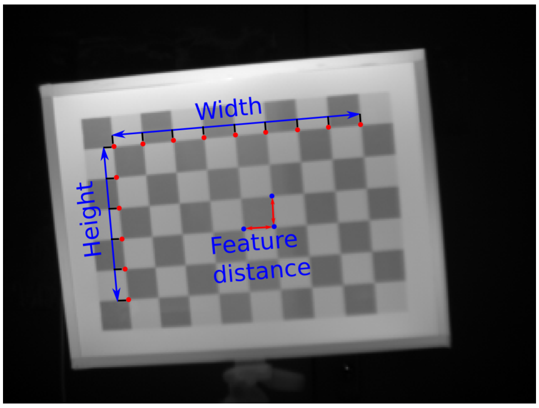
2.1 Motivation

The parameters of the perspective projection are required for the reconstruction of spatial data from cameras. The MRT Camera Calibration Toolbox was developed for calculating these parameters via an intuitive graphical user interface with feedback during the calibration process, visualization of the calculated parameters, handling of different sized images for stereo mode and handling of typical user errors.

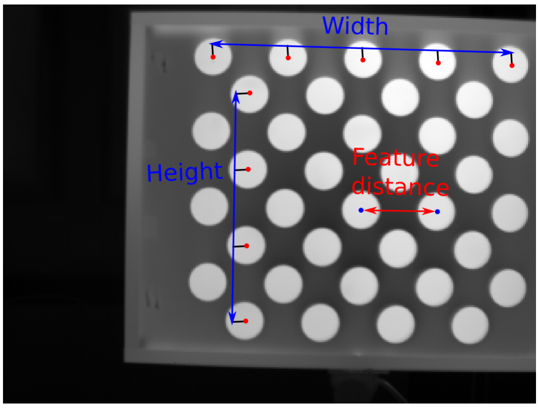
2.2 Background

2.2.1 Calibration Targets

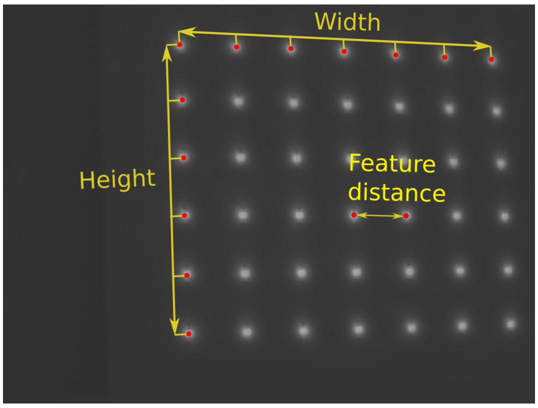
To get the parameters of the perspective projection, a calibration pattern is required. This is constructed with defined geometrical dimensions (e.g. width, height and distance) so that every view is described by several 3D-2D point correspondences. The used calibration patterns are chessboard, asymmetric and symmetric circle grid, as shown in Figure 1. For the symmetrical target, the circular features are realized by means of LEDs.



(a) Pattern type: Chessboard, pattern width: 9, height: 6



(b) Pattern type: Asymmetric grid, pattern width: 9, height: 4



(c) Pattern type: Symmetric grid, pattern width: 9, height: 6

Figure 1: Calibration Pattern

2.2.2 Camera calibration

Using the detected 2D-points of a target (*image points* $\boldsymbol{x}$) and their corresponding 3D-Points (*object points* $\boldsymbol{op}$) from several poses of a calibration pattern, the OpenCV function *calibrateCamera()* finds the camera and distorsion parameters (intrinsics). For stereo images, the function *stereoCalibrate()* calculates not only the intrinsics but also the spatial transformation between the coordinate systems of the two cameras.

2.2.3 Object points projections

Using the 3D object point $\boldsymbol{op_i}$, image point $\boldsymbol{x_{ji}}$, camera parameters $\boldsymbol{c_j}$ and distorsion parameters $\boldsymbol{d_j}$ (j for each camera and i for each pattern feature), the object pose is estimated in terms of its rotational matrix $\boldsymbol{R_{ji}}$ and translation vector $\boldsymbol{T_{ji}}$) with the OpenCV functions *solvePnPRansac()* and *Rodrigues()*. Using the OpenCV function *projectPoints()*, the corresponding projected points $\boldsymbol{\hat{x}_{ji}}$ are calculated.

In stereo case, a second set of projected points $\boldsymbol{\hat{x}^s_{ji}}$ is calculated using the object pose between the cameras given by a rotational matrix $\boldsymbol{\hat{R}_{ki}}$ and a translation vector $\boldsymbol{\hat{T}_{ki}}$ (k is the other camera). These vectors are calculated using $\boldsymbol{R_{ji}}$, $\boldsymbol{T_{ji}}$ and the extrinsic parameters given by the rotational matrix $\boldsymbol{R}$ and translation matrix $\boldsymbol{T}$.

For the projection of the image points from the first to the second camera, the Equation (1) is used to find $\boldsymbol{\hat{R}_{2i}}$ and $\boldsymbol{\hat{T}_{2i}}$ and then, using $\boldsymbol{c_2}$, $\boldsymbol{d_2}$ and the function *projectPoints()*, the corresponding projected points stereo $\boldsymbol{\hat{x}^s_{2i}}$ are calculated.

$$\begin{aligned} \boldsymbol{\hat{R}_{2i}} &= \boldsymbol{R} \cdot \boldsymbol{R_{1i}} \\ \boldsymbol{\hat{T}_{2i}} &= \boldsymbol{R} \cdot \boldsymbol{T_{1i}} + \boldsymbol{T} \end{aligned} \tag{1}$$

For the projection of the image points from the second to the first camera, the Equation (2) is used to find $\boldsymbol{\hat{R}_{1i}}$ and $\boldsymbol{\hat{T}_{1i}}$ and then, using $\boldsymbol{c_1}$, $\boldsymbol{d_1}$ and the function *projectPoints()*, the corresponding projected points stereo $\boldsymbol{\hat{x}^s_{1i}}$ are calculated.

$$\begin{aligned} \boldsymbol{\hat{R}_{1i}} &= \boldsymbol{R}^{-1} \cdot \boldsymbol{R_{2i}} \\ \boldsymbol{\hat{T}_{1i}} &= \boldsymbol{R}^{-1} \cdot \boldsymbol{T_{2i}} - \boldsymbol{R^{-1}} \cdot \boldsymbol{T} \end{aligned} \tag{2}$$

The OpenCV stereo calibration function calculates the return error only using the projections with the intrinsic parameters. In the MRT Camera Calibration Toolbox, the calculated error considers also the projection between the two cameras using the extrinsics parameters.

2.2.4 Different sized images

The *stereoCalibrate()* has only one *imageSize* parameters, which does not allow to use images of different sizes, common case with the use of multiple cameras. For that reason, a transformation in the image points coordinates $\boldsymbol{x_q}$ of the camera with the least resolution (w_{min},h_{min}) is performed so that the resolution of the image will be the same as the other camera (w_{max},h_{max}). The transformation is shown in Equation (3).

$$
\begin{aligned}
w_{adj} &= \frac{w_{max} - w_{min}}{2} \\
h_{adj} &= \frac{h_{max} - h_{min}}{2} \\
\tilde{\boldsymbol{x}}_{\boldsymbol{q}} &= \boldsymbol{x}_{\boldsymbol{q}} + (w_{adj}, h_{adj})
\end{aligned} \tag{3}
$$

These transformed image points $\boldsymbol{x}_{\boldsymbol{q}}$ are used as inputs for the *stereoCalibrate()* function. Neither the distorsion parameters or the focal lengths are affected through this transformation when the calibration is performed but the principal point does, for which the c_x and c_y parameters are corrected with the Equation (4), where ($\tilde{c}_x$,$\tilde{c}_y$) is the principal point obtained as result of the calibration.

$$
\begin{aligned}
c_x &= \tilde{c}_x - h_{adj} \\
c_y &= \tilde{c}_y - w_{adj}
\end{aligned} \tag{4}
$$

2.2.5 Root Mean Square (RMS) error calculation

The calibration function returns an RMS reprojection error, which is "the sum of squared distances between the observed projections imagePoints and the projected (using *projectPoints()*) objectPoints" [1]. Since the MRT Camera Calibration Toolbox performs also clustering calculation for getting the camera parameters, this error is calculated multiple times as an indicator of the variance of the computed values using Equation (5), the image points $\boldsymbol{x}_{\boldsymbol{ji}}$ and its projection $\hat{\boldsymbol{x}}_{\boldsymbol{ji}}$ for a pose p.

$$
RMS_{pj} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_{\boldsymbol{ji}} - \hat{\boldsymbol{x}}_{\boldsymbol{ji}}\|^2} \tag{5}
$$

For a set of m different poses, the RMS error for each camera is:

$$
RMS_j = \sqrt{\frac{1}{m}\sum_{p=1}^{m} (RMS_{pj})^2} \tag{6}
$$

The total RMS error in stereo mode is:

$$
RMS_t = \sqrt{\frac{1}{2m}\left(\sum_{p=1}^{m} (RMS_{p1})^2 + \sum_{p=1}^{m} (RMS_{p2})^2\right)} \tag{7}
$$

Complementing the error accuracy visualization, the pixel distance error is also calculated using the Equation (8).

$$
d_i = \|\boldsymbol{x}_{\boldsymbol{ji}} - \hat{\boldsymbol{x}}_{\boldsymbol{ji}}\| \tag{8}
$$

2.2.6 Averaging Rotational Matrices

For calculating the mean of rotational matrices, quaternions are averaged using by using the algorithm described in [2]. The python implementation is based on [3].

2.3 Application

A high accuracy on of the perspective projection parameters is required for a successful reconstruction of spatial data from cameras, for example, when creating 3D thermograms from depth sensors and thermal imaging cameras.

By means of the RMS and the distance error, it can easily be determined which calibration images are not suitable for the calibration process. By using the cluster calibration, different subsets can be made for statistical analysis for the different parameters of the camera.

3 System Summary

3.1 System Configuration

The MRT Camera Calibration Toolbox was tested on devices with Linux and Windows operating system. The modules shown in Listing 1 have to be installed.

Listing 1: Required Python modules for the MRT Camera Calibration Toolbox

```
OpenCV 4.1
tkinter 8.6
Pillow 5.4.1
numpy 1.16.4
matplotlib 1.3.1
```

3.2 License

The MRT Camera Calibration Toolbox is licensed under the GNU General Public License v3.0 (See: https://www.gnu.org/licenses/gpl-3.0.de.html).

4 Getting Started

4.1 Installation

The latest source code version can be obtained from https://github.com/MT-MRT/MRT-Camera-Calibration-Toolbox, which can be directly executed on the computer through the terminal, accessing first to the corresponding folder as shown in Listing 2.

Listing 2: Executing MRT Camera Calibration Toolbox

```
$ cd path/to/your/toolbox
$ python3 main.py
```

4.2 Start interface

Figure 2 shows the default graphical interface of the MRT Camera Calibration Toolbox.

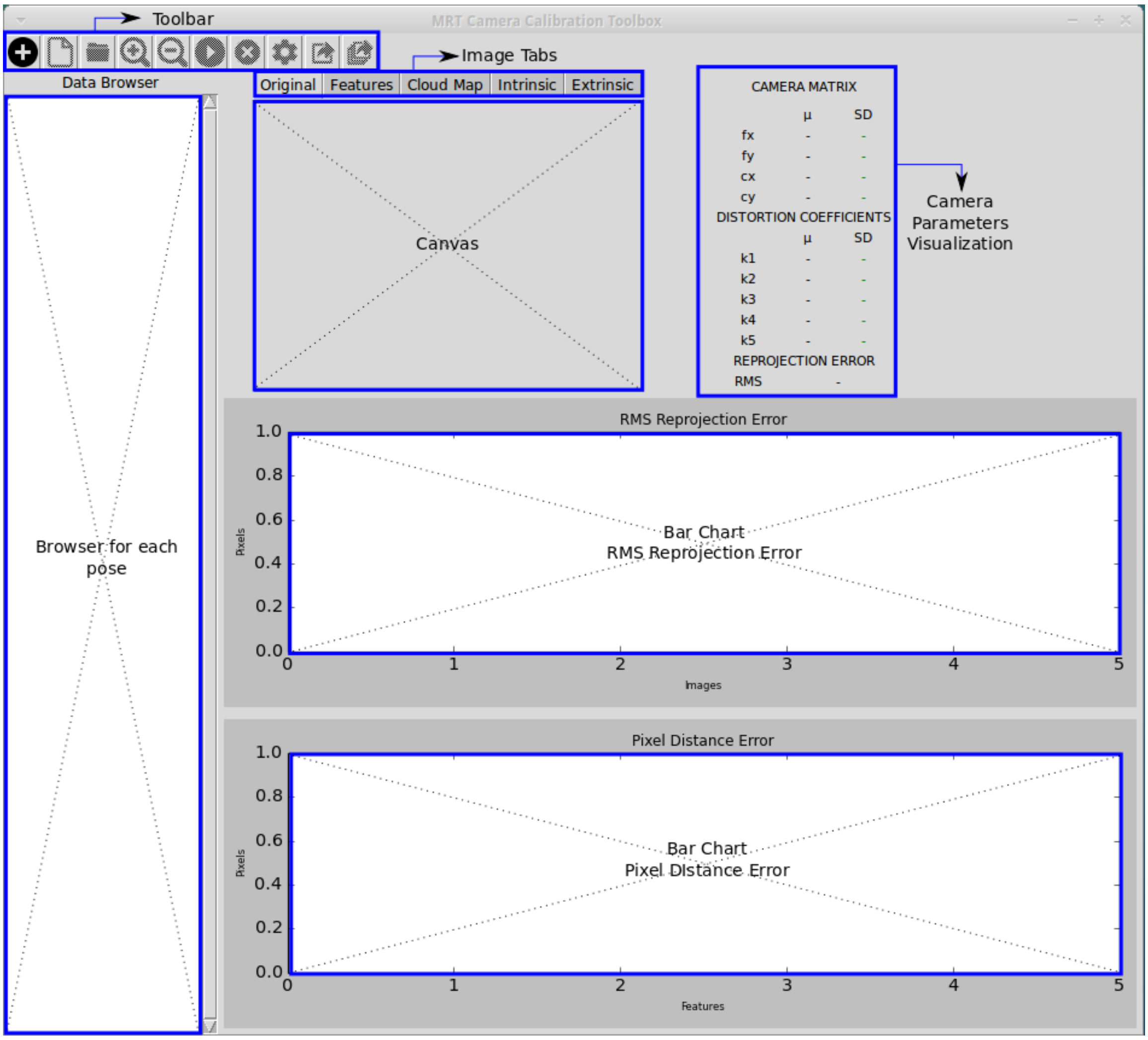


Figure 2: Overview of the MRT Camera Calibration Toolbox interface

4.3 System Menu

Figure 3 shows the 9 buttons of the menu toolbar in the upper left corner of the application (See: appendix A to see the source and licenses of the icons).

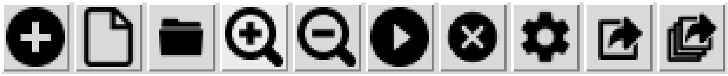


Figure 3: Menu buttons. Left to right: add session, add images/text per file, add images/files per folder, zoom in, zoom out, run calibration, delete session, calibration settings, export parameters of the perspective projection and export parameters of the perspective projection for each calibration group

4.3.1 “Add session” button

This button launches a pop-up window to add a new session, which by default is configured for *Images* (see Figure 4). The MRT Camera Calibration Toolbox supports both upload of images (JPG and PNG) and text-files containing feature descriptions. The pattern configuration is represented in the right picture of the pop-up window, depending on the pattern type, its width and height (see Figure 5). Thus, the stereo mode can be selected through a checkbox.

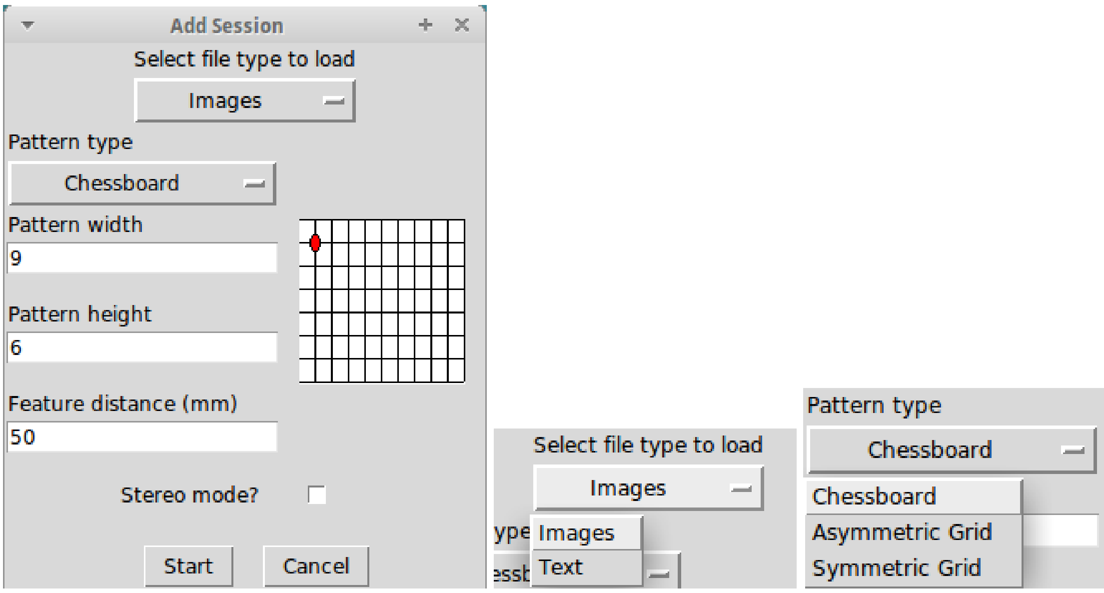
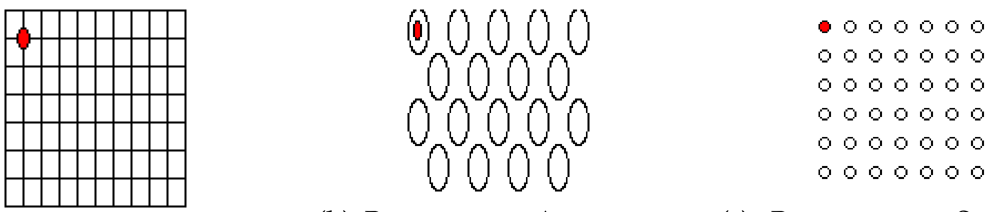


Figure 4: Add session for loading images pop-up window



(a) Pattern type: Chessboard, pattern width: 9, height: 6

(b) Pattern type: Asymmetric grid, pattern width: 9, height: 4

(c) Pattern type: Symmetric grid, pattern width: 9, height: 6

Figure 5: Pattern representation. An example of the parameters is shown in Figure 1

When *Text* is selected (see Figure 6), the image width and height has to be set, and a text file containing the 3D-points of the pattern, which format is shown in Listing 3.

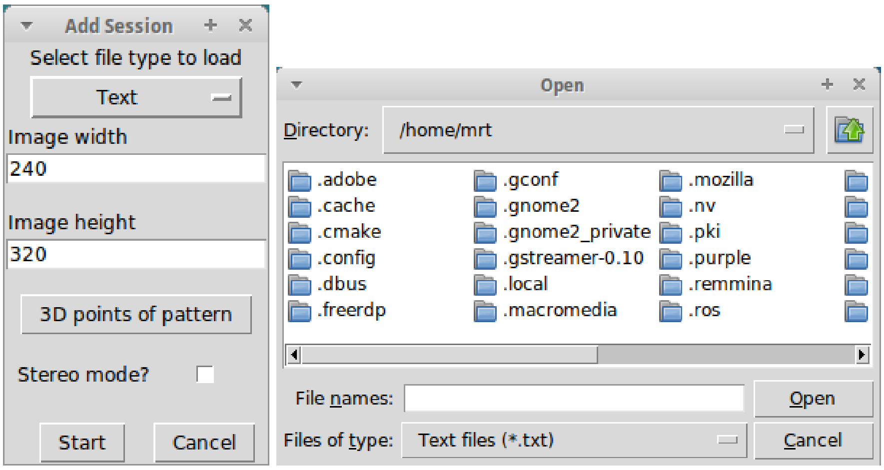
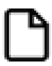


Figure 6: Add session for loading text pop-up window

Listing 3: 3D-points file *p3d.txt*, for a pattern with two object points $op_1 = (0, 0, 0)$ and $op_2 = (0, 50, 0)$

```
0.0,0.0,0.0,0.0,50.0,0.0
```

4.3.2 "Add images/text per file" button



This button shows the user a pop-up window for file selection. When the session is configured for *Images*, the selection is limited only for the supported images types (JPEG and PNG), otherwise, when configured for *Text*, the selection is limited for .txt files, which format is shown in Listing 4. For selecting multiple files, while holding down the Ctrl key, click on each of the files you want to select. In case you want to select a contiguous group of files, use the Shift key.

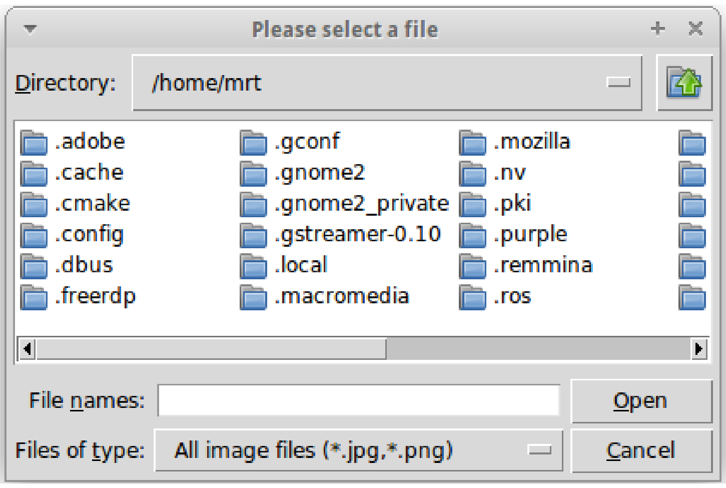


Figure 7: Add file pop-up window

Listing 4: 2D-points file *f1.txt*, for a pattern with two image points $x_{1,1} = (198, 19)$ and $x_{1,2} = (199, 37)$

```
198.0,19.0,199.0,37.0
```

4.3.3 “Add images/text files per folder” button

A pop-up window for the folder selection is launch by this button. When the session is configured for *Images*, all supported images are imported, otherwise, when configured for *Text*, all .txt files are imported. Because of the implemented sorting algorithm, it is recommended that the names of the images start with at least one character followed by a number.

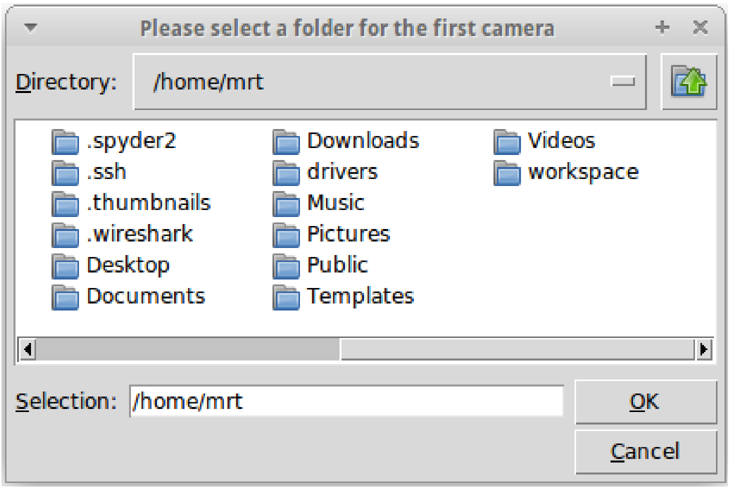


Figure 8: Add folder pop-up window

4.3.4 “Zoom in” button

This button will sink itself, relief the *zoom out* button and for every click over the canvas will make the images bigger and centered in the clicked point. The maximum zoom is ten times the original image scale in the canvas. In case this button is already sunken, it will relief itself to deactivate the *zoom in* event. Scrolling mouse wheel forward activates also this zoom-in event.

4.3.5 “Zoom out” button

This button will sink itself, relief the *zoom in* button and for every click over the canvas will make the images smaller centered on the clicked point. The minimum zoom is the original image scale in the canvas. In case this button is already sunken, it will relief itself to deactivate the *zoom out* event. Scrolling mouse wheel forward activates also this zoom out event.

4.3.6 “Run calibration” button

This button will show a pop-up window for running camera calibration, which by default is configured for *Clustering Calculation* (see Figure 9). By hitting the F5 key, the run calibration pop-up window will also be launched.

For the *Clustering Calculation*, multiple calibrations are made with different groups of images to finally average the results and thus obtain the extrinsic and intrinsic parameters. Here, the number groups k and its number of elements per group r are set. The percentage of completion of the calibration process is represented by a progress bar. Below this bar is a summary of the different steps for the calibration, its status, approximate execution time, remaining time and total time.

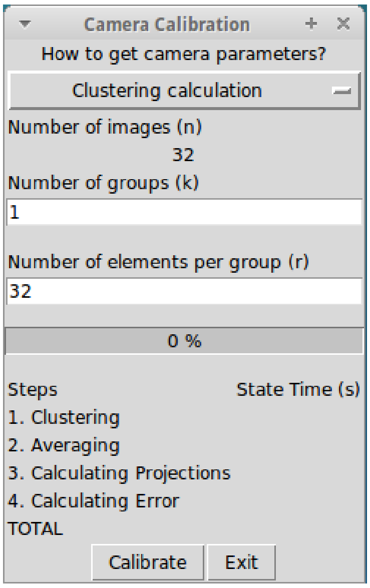


Figure 9: Run calibration for clustering pop-up window

When *Load from file* is selected (see Figure 10), the necessary text files for intrinsics and extrinsics have to be selected. The text format is shown in Listing 5 and 6.

Listing 5: Intrinsic parameters file structure

```
Camera Matrix
778.7386528528 0.0000000000 155.3761289266
0.0000000000 778.8499458806 142.8577609802
0.0000000000 0.0000000000 1.0000000000

Distorsion Coeficients
-0.5693527579
-2.1592220898
-0.0091852577
-0.0048091900
13.4723701420
```

Listing 6: Extrinsic parameters file structure

```
Rotation Matrix between Cameras
0.9992704515 -0.0039493328 -0.0379864117
0.0031515163 0.9997736744 -0.0210396772
0.0380609071 0.0209046130 0.9990567374

Translational Vector between Cameras
202.9093129274
-1.1670048157
-3.9250568566
```

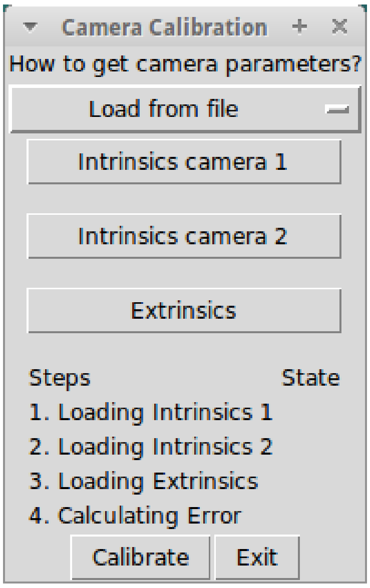


Figure 10: Run calibration for load files pop-up window

4.3.7 "Delete session" button

This button will erase the entire session, which is the pattern parameter configuration, camera mode, all the loaded images, and their calibrations results. A confirmation alert will ask first for confirmation before erasing the session, as shown in Figure 11.

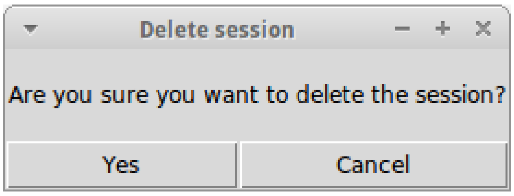
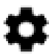


Figure 11: Delete pop-up window

4.3.8 "Calibration settings" button



This button allows setting the multiples flags for the camera calibrations process. A more detailed description of the effect of each flag can be found in [1].

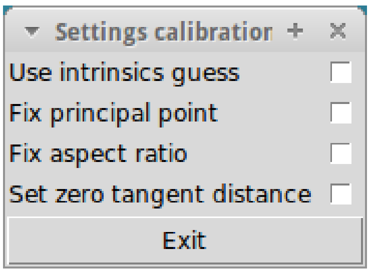


Figure 12: Settings pop-up window

4.3.9 “Export parameters of the perspective projection” button

After the required intrinsics and extrinsics are calculated, the results can be exported to a text file by clicking this button, which shows a dialog pop-up window to save the data in the given files.

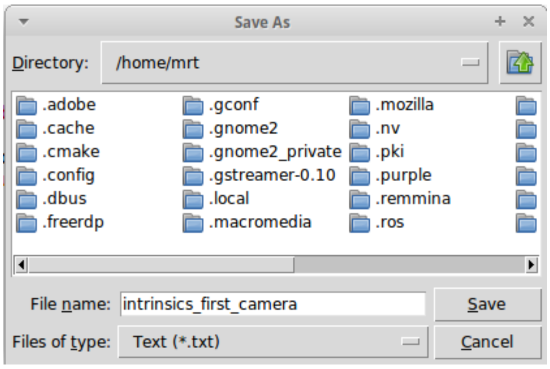


Figure 13: Export pop-up window

4.3.10 “Export parameters of the perspective projection for each calibration group” button

When the required intrinsics and extrinsics are calculated using calibration with clusters (see subsection 5.4), the report of all the results per group can be exported to text files by clicking this button, which shows a dialog pop-up window to select the destination folder.

4.4 Exit System

The MRT Camera Calibration Toolbox can be closed by clicking the ‘X’ in the top right corner of the application, or by pressing the key sequence Alt + F4.

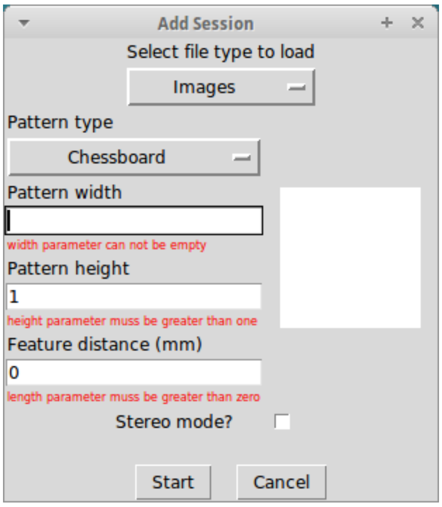
5 Using the System

5.1 Start new session

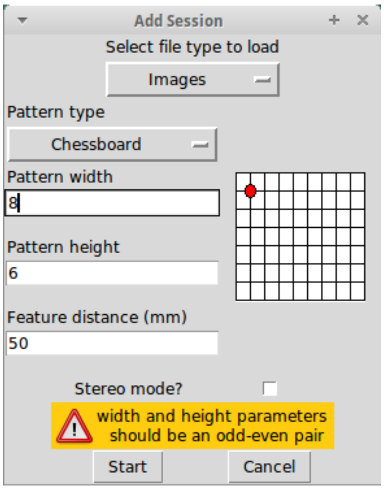
After launching the MRT Camera Calibration Toolbox, only the add session button is available, which launches the pop-up window for setting the configurations of a new session.

For chessboard pattern type, the width and height values are defined by the number of row and columns of inner corners of the chessboard, while for the asymmetric grid pattern type, the width and height values are the number of circles per row and column.

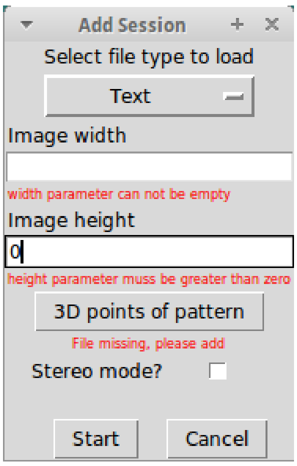
In case that one or more of the parameters is empty or has an invalid value (pattern width and height must be greater than one and feature distance must be greater than zero) when clicking yes button, the corresponding error will be shown in the pop-up window, as shown in Figure 14a and 14c. The width and height should be an even-odd pair. In case it is not, a warning will inform the event in the pop-up window, as shown in Figure 14b.



(a) Entry parameters error notification, image session



(b) Entry parameters warning notification, image session



(c) Entry parameters error notification, text session

Figure 14: Errors and warning notifications when adding session

When the stereo mode checkbox is selected, the interface will be adapted (see Figure 15) for the two camera parameters visualization, otherwise, the interface will remain as the initial one (see Figure 2).

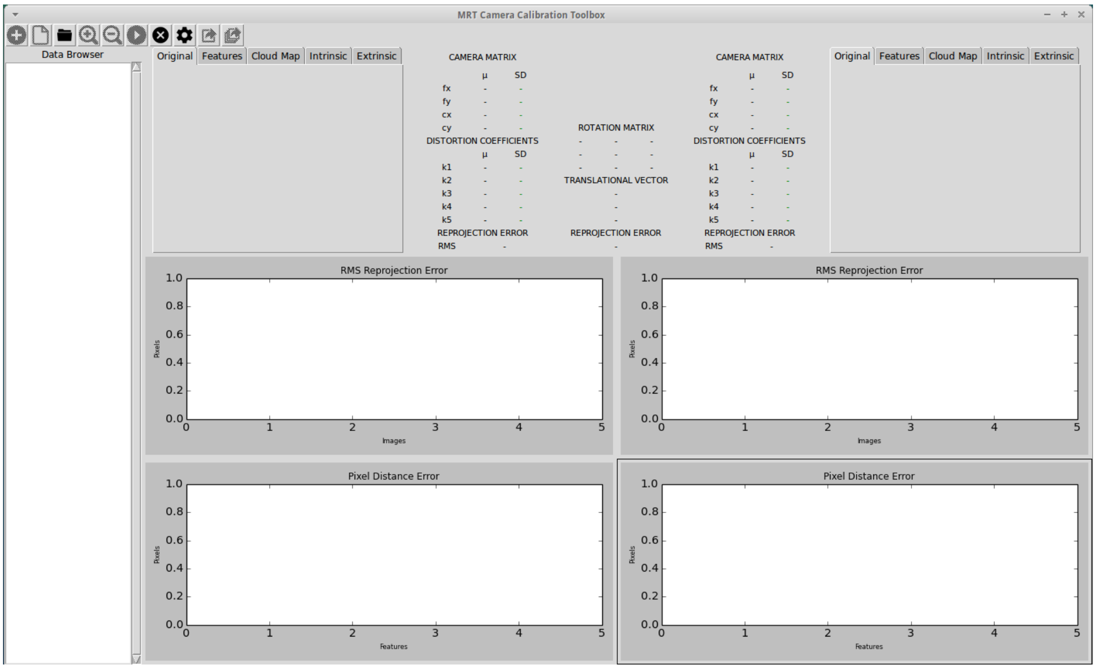


Figure 15: Interface of MRT Camera Calibration Toolbox for stereo mode

5.2 Add images

After successfully adding the session, the add session button is disabled until the session is deleted. The add image per folder button is enabled, which will ask for one folder for the single-camera mode and two folders for the stereo mode. In case the session is in single mode, the add image per file button will also be enabled.

A pop-up with a progress bar will show information about the imported files (see Figure 16). In case that the file selection contains repeated or rejected files, a pop-up window will resume all failures.

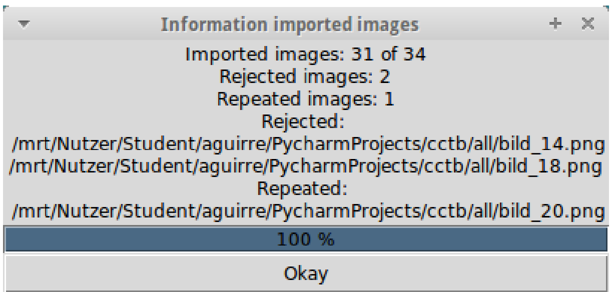


Figure 16: Importing pop-up window

Images are rejected either because the pattern features of the image do not match those configured when adding the session, or the pattern feature detection algorithm failed for the image (e.g. low contrast).

5.3 Image interaction

When at least one file (or one pair of files for the stereo mode) is successfully imported, the zoom-in and zoom out buttons are enabled. The selection of the files is made by clicking its name in the data browser (left side of the interface) as shown in Figure 17.

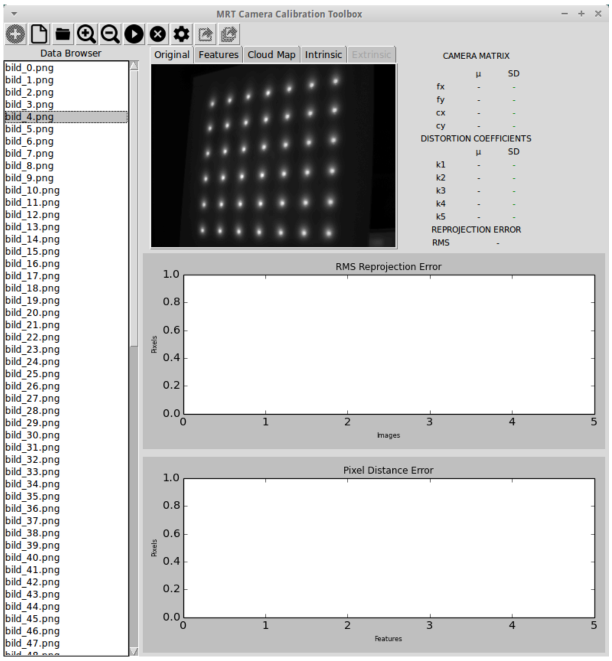
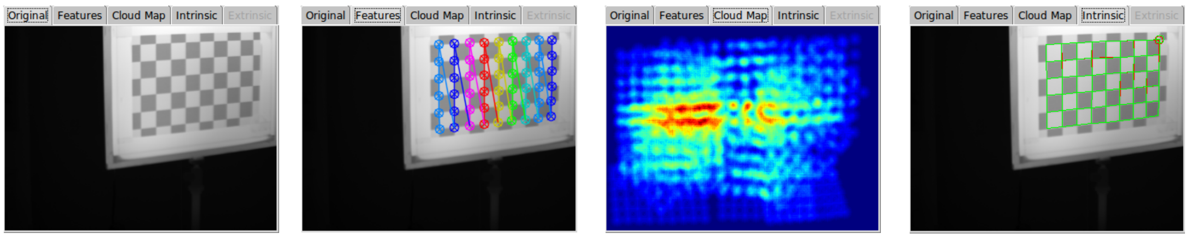


Figure 17: Example of selection of image 'bild_4.png' in a single mode session with an asymmetric grid pattern type

For each image, there are different tabs showing pictures that represent different features of the pattern and the calibration results (see Figure 18). The tabs are Original, Features, Cloud Map, Intrinsic and Extrinsic (only for stereo mode).



(a) Original image

(b) Image with detected features

(c) Cloud map of all the different poses

(d) Intrinsic parameters representation

Figure 18: Tabs for a single pose in a session with a chessboard pattern type and multiple loaded images

Original shows the loaded normalized image, *Features* will show the detected points for the pattern type, *Cloud Map* represents the feature points distribution on the camera screen for all different poses, *Intrinsic* compares the original image (green lines) with its projection (red lines) using the calculated intrinsic parameters after the calibration, and *Extrinsic* compares the original image (green lines) with its projection (red lines) calculated from the other camera using the calculated intrinsics and extrinsics after the calibration. In case that the calibration hasn't been run yet, the two last tabs represent only the original pattern lines.

5.4 Calibrate

When at least one image (or one pair of images for the stereo mode) is successfully imported, the configure parameters for the calibration button and run calibration button are also enabled, which will launch the calibration pop-up window shown in Figure 9.

When for the calibration pop-up window *Load from file* is chosen, the necessary buttons for the file selection will be generated as follows: for the single mode, only one button for the intrinsics, and for stereo mode, two buttons for the intrinsics of each camera and an additional button for the transformations between the two cameras (see Figure 10). The extrinsic file is optional, when no loaded, the program calculate the extrinsics before calculating the error, otherwise, the program calculates directly the error using only the loaded intrinsics and extrinsic parameters.

After selected the corresponding file through a file dialog, the chosen file name is shown below the selector button, and the information for the parameters is updated, as shown in Figure 19. When the *Calibrate* button is clicked and the necessary files are not loaded, the situation will be notified as shown in Figure 20c.

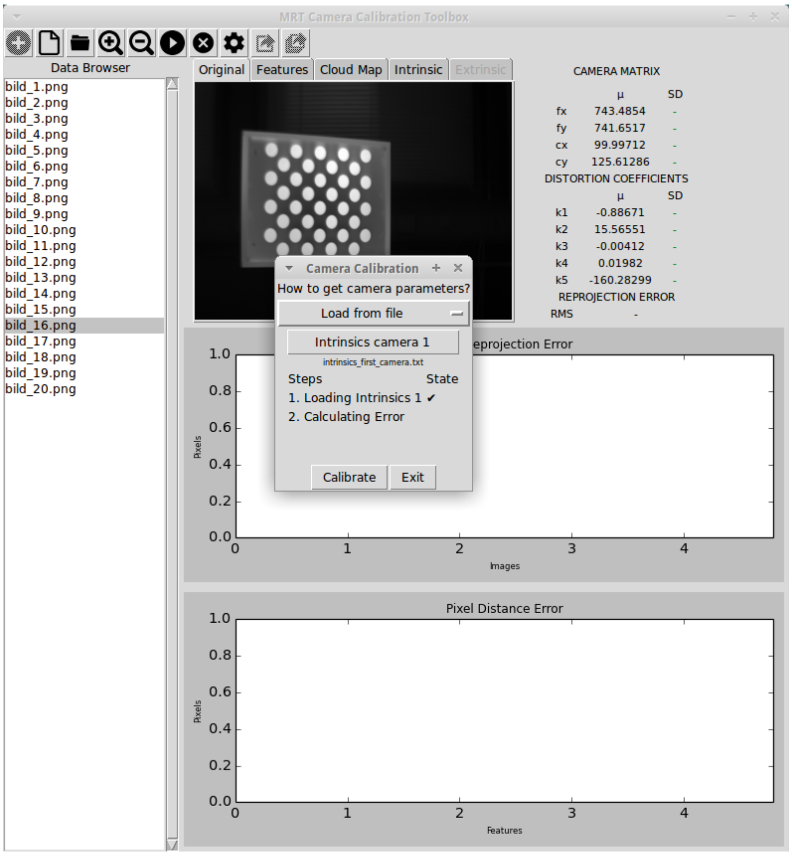
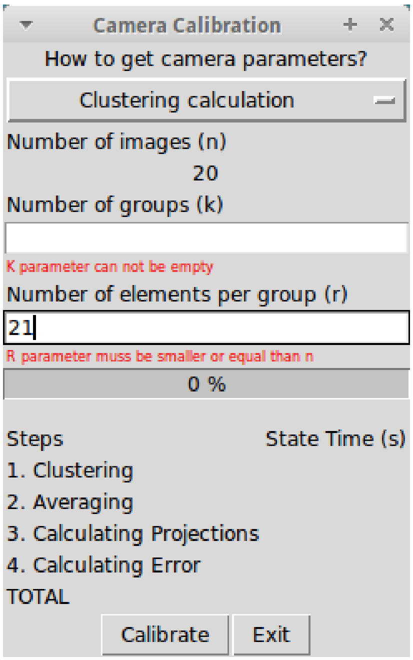
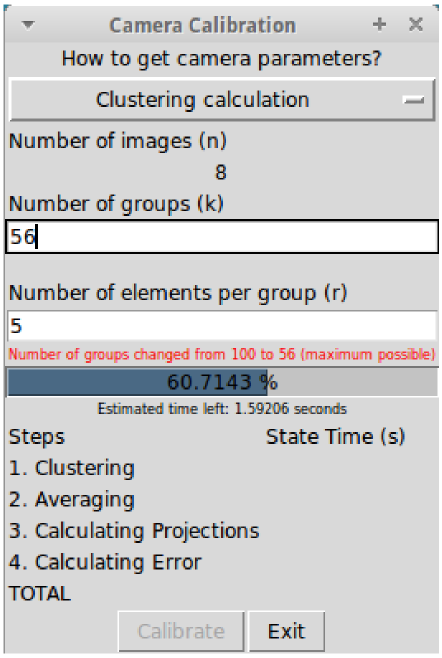


Figure 19: Loading parameters from 'intrinsics_first_camera_all.txt' text file

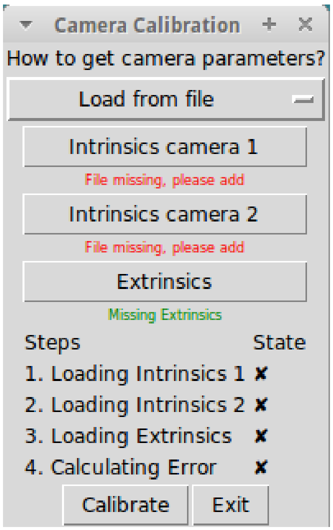
When for the calibration pop-up window *Clustering calculation* is selected, it is necessary to set the number of groups k and its number of elements r (see Section 4.3.6). In case that the parameters are empty or null, the pop-up window will show an error message. The error notification is also given when r is greater than the total number of images, as shown in Figure 20a. When k is greater than the maximum possible, the calibration will run but with the correction of these parameters, as Figure 20b shows.



(a) Calibration setting parameters error notification



(b) Correction of the number of groups according to the maximum possible



(c) Calibration files selection error notification

Figure 20: Error notifications when calibrating

When the calibration is made through clustering, the final value of each intrinsic parameter (f_x, f_y, c_x and c_y for the camera matrix and k_1, k_2, k_3, k_4, k_5 for the distortion coefficients) is the mean of all the calculated parameters. The standard deviation of each parameter is also given.

For the stereo mode, the final value of the translation vector is the simple mean of all the calculated translation vectors, while for the final value of the rotational matrix, the mean of all the calculated rotational matrices is performed as described in section 2.2.6.

After the calibration succeeds, the *RMS reprojection Error* and *Pixel Distance Error* charts are updated according to the calculated parameters, as shown in Figure 21.

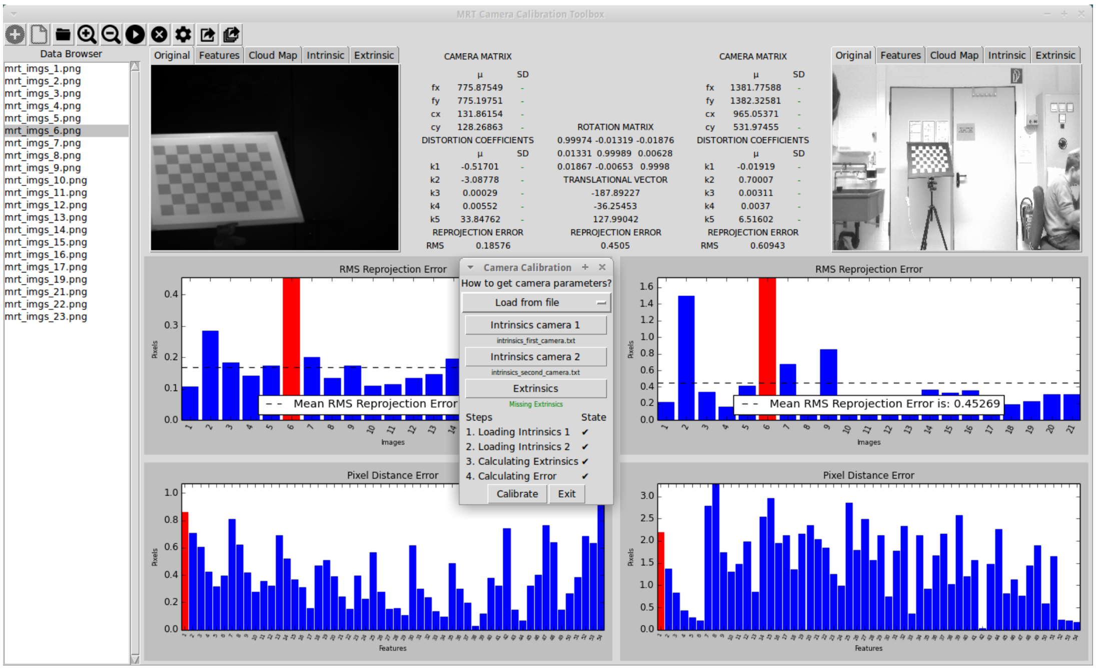


Figure 21: Interface update after a successful calibration ('Load from file') for a stereo mode session for a chessboard pattern type

5.5 Calibration error bar interaction

The *RMS reprojection Error* chart shows the RMS error for each pose. A black dashed line represents the mean of the RMS error for all the poses per camera, for which the legend below shows its value. The red color in the bar represents the currently selected image, for example, in Figure 21, the sixth image (mrt_imgs_6.png) is selected. Click event over a bar will change the active image in the data browser and so the tabs images are updated.

The *Pixel Distance Error* chart shows, for each detected feature, the pixel distance error between the original image and its reprojection using the intrinsics (for the single mode) or using the extrinsics and view through the other camera (for the stereo mode). The red color in the bar represents the currently selected feature, which for the Intrinsic and Extrinsic tab is highlighted with a green circle, for example, in Figure 21, the feature in the left down corner is selected.

In case that for the intrinsics and extrinsics parameters found with the calibration the error is too high (Due to a division, where the denominator is near to zero (around $10^{-}323$) so that python returns infinity), the calibration won't succeed and it will necessary to perform another one with more suitable parameters.

5.6 Exporting intrinsics and extrinsics

After successful calibration, the "export parameters of the perspective projection" button is enabled. The format of the exported files is shown in Listing 5 and 6.

5.7 Exporting intrinsics and extrinsics of the calibration with clustering

After successful calibration, the "export parameters of the perspective projection" button is enabled. A folder is selected through a dialog box to export all the necessary text files. The file *samples.txt* in Listing 7 is created and shows the used files for the calibration. Then, single text files for each intrinsic parameter are generated (f_x, f_y, c_x and c_y for the camera matrix and k_1, k_2, k_3, k_4, k_5 for the distortion coefficients), where each row contains the result for each iteration. Listing 8 shows an example of an exported text file for the parameter f_x.

Listing 7: Clustering calibration file *samples.txt* indicating the used files for each calibration, for a calibration with 5 groups with 3 elements each one

```
[[/bild_2.png,/bild_5.png,/bild_13.png]]
[[/bild_16.png,/bild_22.png,/bild_23.png]]
[[/bild_1.png,/bild_5.png,/bild_15.png]]
[[/bild_0.png,/bild_4.png,/bild_6.png]]
```

Listing 8: Clustering calibration file *fx_cam_1.txt* for the parameter for a calibration with 5 groups

```
781.947756237
777.321520848
778.194542271
776.612838795
780.472309719
```

Additionally, in the same selected folder, for the extrinsic parameters two text files are generated: a text file for the rotation matrices, where each row is the rotation matrix for each iteration group (see Listing 9), and a text file for the translation vectors, where each row is the translation vector for each iteration group (see Listing 10).

Listing 9: Clustering calibration file *rotational.txt* for the rotational matrices, for a calibration with 5 groups

```
0.9911 -0.0124 -0.0145 0.01354 0.9991 0.0050 0.0128 -0.0034 0.9976
0.9910 -0.0115 -0.0178 0.01381 0.9991 0.0047 0.0191 -0.0038 0.9956
0.9943 -0.0198 -0.0191 0.01359 0.9999 0.0033 0.0187 -0.0036 0.9914
0.9989 -0.0123 -0.0172 0.01374 0.9999 0.0031 0.0191 -0.0032 0.9998
0.9981 -0.0132 -0.0173 0.01348 0.9990 0.0032 0.0151 -0.0033 0.9989
```

Listing 10: Clustering calibration file *translation.txt* for the translation vector, for a calibration with 5 groups

```
-187.7350813198 -36.4105178909 128.8631907191
-187.1206821819 -36.3591566868 130.7532856474
-186.7042789011 -35.7234713184 128.7539183155
-184.1398024863 -34.5389392056 121.0859793060
-188.3206539881 -35.1704466259 129.9356747528
```

5.8 Delete images

An image can be deleted by selecting it by click and then pressing the Del key. In case a calibration has already been run, the error charts *RMS reprojection Error* and *Pixel Distance Error* will be updated and its not update status will be shown in the interface, as shown in Figure 22.

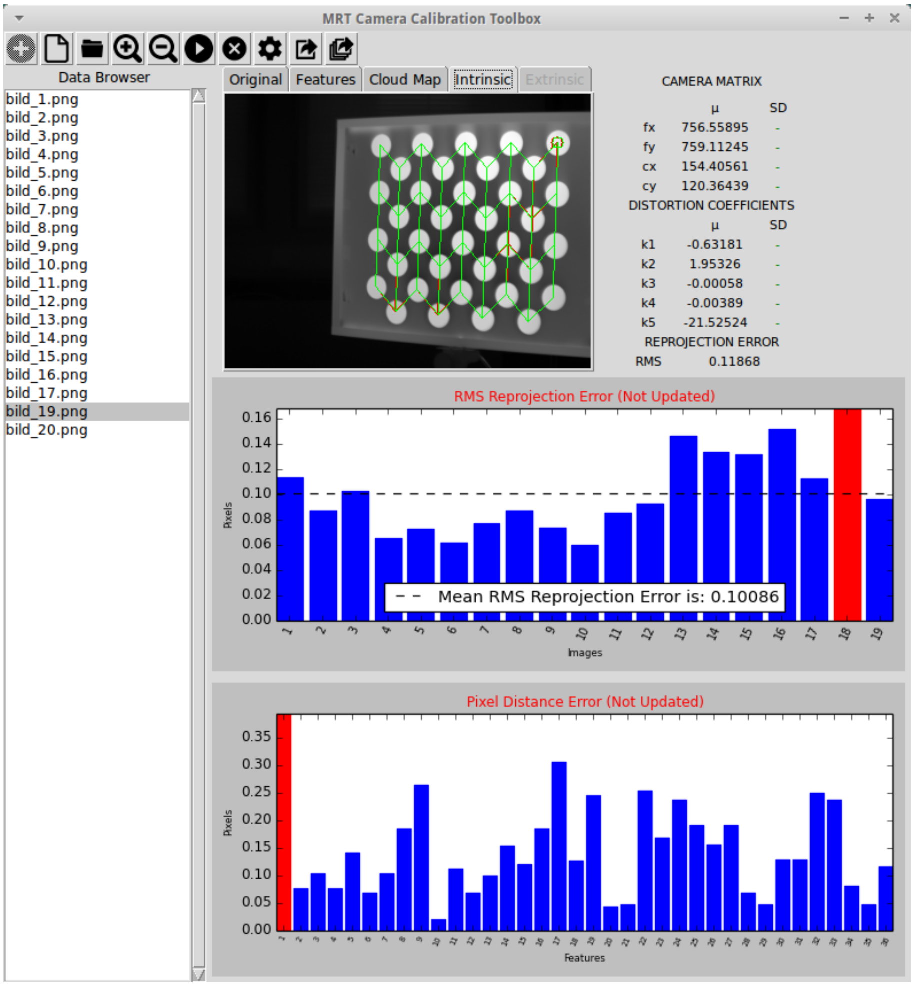


Figure 22: Notification of not updated status for the RMS error charts in a single mode session for an assymetric grid pattern type

5.9 Delete all session

After pressing the "delete session" button and press 'Yes' in the confirmation pop-up window shown in Figure 11, all the pattern configurations, mode, images, error chart, and calculated parameters are erased. Now it's time to start again!

Appendix A Icons licenses

icon type	link	license type
Delete	https://www.iconfinder.com/icons/309090/close_cross_delete_icon	Creative Commons (Attribution-Share Alike 3.0 Unported)
Export	https://www.iconfinder.com/icons/216190/export_icon	Creative Commons (Attribution-Share Alike 3.0 Unported)
Export All	https://www.iconfinder.com/icons/216190/export_icon	Creative Commons (Attribution-Share Alike 3.0 Unported)
Folder	https://www.iconfinder.com/icons/211608/folder_icon	MIT License
Open	https://www.iconfinder.com/icons/227587/file_icon	Creative Commons (Attribution-Share Alike 3.0 Unported)
Play	https://www.iconfinder.com/icons/326581/circle_fill_play_icon	Free for commercial use
Plus	https://www.iconfinder.com/icons/183316/add_plus_icon	Free for commercial use
Settings	https://www.iconfinder.com/icons/185095/settings_streamline_icon	Free for commercial use (Include link to authors website)
Zoom Less	https://www.iconfinder.com/icons/106233/out_zoom_icon	Free for commercial use
Zoom More	https://www.iconfinder.com/icons/106237/in_zoom_icon	Free for commercial use

References

[1] OpenCV dev team, „Camera Calibration and 3D Reconstruction," https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, 2019.

[2] F. L. Markley, Y. Cheng, J. L. Crassidis und Y. Oshman, „Averaging quaternions," *Journal of Guidance, Control, and Dynamics*, Band 30, Nr. 4, S. 1193–1197, 2007.

[3] Christoph Hagen, „(Correctly) averaging quaternions," https://github.com/christophhagen/averaging-quaternions/blob/master/averageQuaternions.py, 2019.