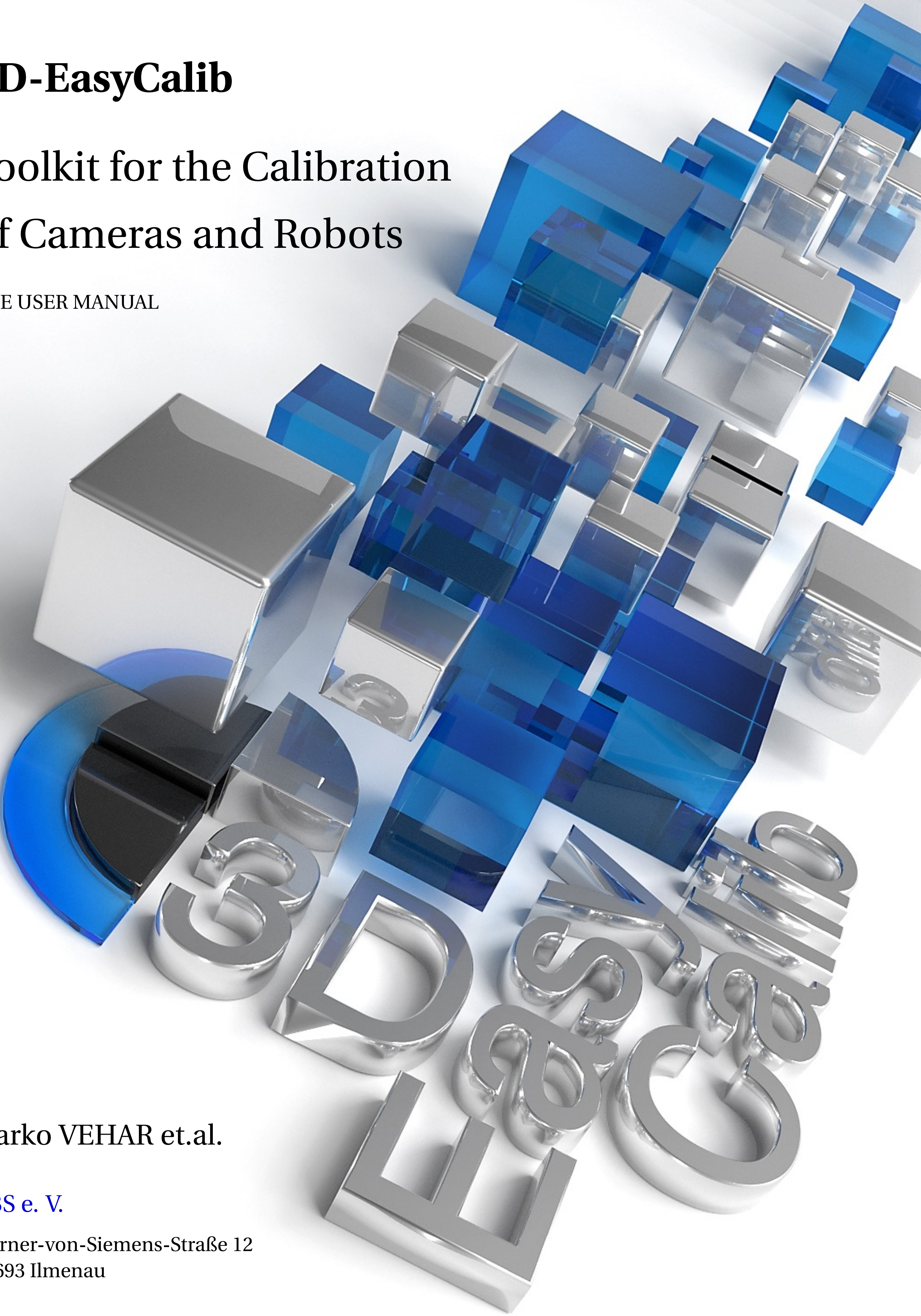
3D-EasyCalib

Toolkit for the Calibration of Cameras and Robots

THE USER MANUAL



Darko VEHAR et.al.

ZBS e. V.

Werner-von-Siemens-Straße 12
98693 Ilmenau

Legal Notice

Werner-von-Siemens-Straße 12, 98693 Ilmenau, Germany
info@zbs-ilmenau.de
www.zbs-ilmenau.de

ZBS Publication Series

ISSN: 1432-3346

Contents

CHAPTER

1 Introduction to Camera Calibration

A camera creates planar images of the three-dimensional physical world by using perspective projection. This technique was first studied and understood in detail during the Renaissance by artists and mathematicians. The painters of that time (see fig. 1.1) derived the geometric construction of perspective projection and applied it to their paintings to create a sense of depth and realism. When photography was invented in the 19th century, the measurement of scenes from perspective images also became interesting for scientific purposes. Since that time, photogrammetry, which addresses this question, has continually evolved. In the same century, mathematicians defined projective geometry, a branch of mathematics that studies properties of geometric figures preserved under projections. This field also deals with points at infinity, which are points where parallel lines appear to meet in a perspective drawing.

Projective geometry provides the theoretical foundation for perspective projection and photogrammetry. However, in practice, it is necessary to account for the specific characteristics and distortions of each camera. This is the subject of camera calibration, which we will discuss in the subsequent sections.

1.1 History of camera calibration

The first methods and procedures for practical camera calibration were motivated by aerial photogrammetry during WWI [7]. The goal was to estimate distances according to the stereoscopic principle from aerial photos and to map the Earth's surface. Since the ideal mathematical camera model differed greatly from a real camera, particularly due to lens distortion, significant effort was initially put into developing mathematical models for lens distortion. The manufacturing tolerances of the time meant the

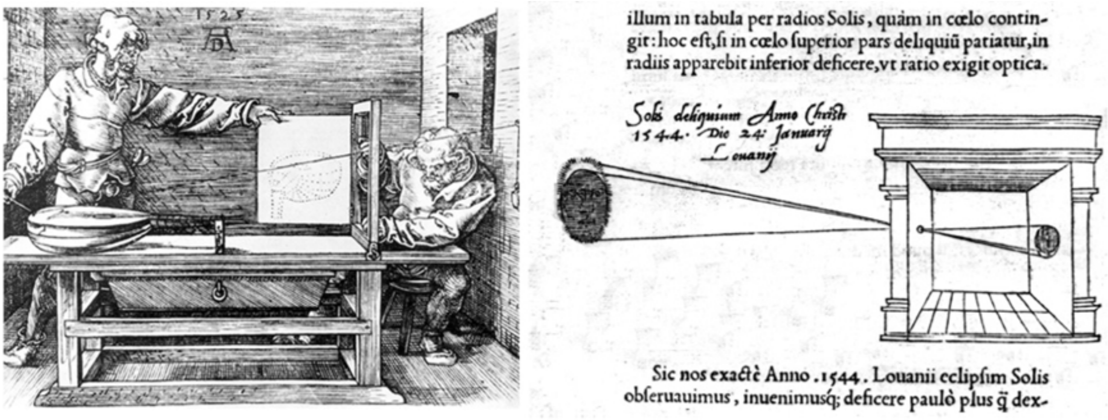


Fig. 1.1: Albrecht Dürer, 1525, Underweysung der Messung (left). The first illustration of a pinhole camera, the so-called Camera Obscura by Reinerus Gemma-Frisius, 1544 (right).

parameters provided by manufacturers for focal length and radial distortion were not precise enough. A correction for decentering distortion was also necessary due to the imprecise alignment of lens elements along the optical axis caused by the manufacturing process.

In 1956, Brown [5] summarized and evaluated the scientific advances in camera lens modeling up to that point. He combined and experimentally investigated the model for radial and tangential distortion presented by Conrady in 1919 [8]. This lens distortion model is still valid today and is explained in **??**. Brown also laid the foundation for modern bundle adjustment for determining the camera parameters in [4].

Determining camera parameters was quite time-consuming before the invention of computers. Originally, star images were used to manually calculate the image formation parameters. This method was replaced by the rotation collimator, based on the goniometer principle, because it was less complex. Again, it was Brown who presented a much simpler method: the so-called plumb-line calibration [6]. Brown described a mathematical model for determining lens distortion parameters based on images of straight lines. This technique proved very useful and was used to calibrate cameras for various applications - from microscopic imaging to aerial photography.

With the advent of the digital era, the optimization of parameters from calibration on computers significantly accelerated. In 1986, Tsai presented a robust, two-step method for determining the intrinsic and extrinsic parameters from a 3D target. This method became popular almost ten years later when Willson [22] provided the first freely available implementation. At this time, ZBS e. V. began working with camera calibration for the first time and has been expanding its competencies in this field ever since. The origins of the 3D-EasyCalib toolkit trace back to the Matlab implementation by Bouguet [2]. He adopted the method presented by Zhang [23], which uses multiple planar targets and optimizes the parameters with a Levenberg-Marquardt gradient descent method. Bouguet recognized that the precision of the determined coordinates of calibration points played a decisive role in effective calibration. The visualization of these errors and the ability to correct them manually set his toolkit apart from the rest. Since there is no reliable method to automatically remove the incorrectly recognized control points, it is important to visualize them to enable manual corrections. His methods were later ported to C++ in the open-source library OpenCV [3]. The methods implemented back then are still used today in almost unchanged form.

1.2 What is camera calibration used for?

Geometric camera calibration is used for various applications that require accurate measurements or 3D reconstruction from images, such as computer vision, robotics, augmented reality, medical imaging, and photogrammetry. Some use-cases of geometric camera calibration are:

- In computer vision, camera calibration can help to correct lens distortion, perform image rectification, and estimate the relative pose and scale of multiple cameras.
- In robotics, camera calibration can enable robots to perceive their environment and perform tasks such as navigation, manipulation, and obstacle avoidance.
- In augmented reality, camera calibration can allow the overlay of virtual objects on real-world scenes with consistent perspective and occlusion.
- In medical imaging, camera calibration can facilitate the registration of images from different modalities, such as X-ray, MRI, and ultrasound.
- In photogrammetry, camera calibration can improve the accuracy and quality of 3D models generated from aerial or terrestrial images.

1.3 What distinguishes a good camera calibration?

Calibration target should be designed based on the working range of the camera and the image resolution. It should also be precisely manufactured.

Image sequence: A sufficient number of images of the target, captured at different distances and orientations to the camera, are required. The camera should move as little as possible while the picture is being taken, and the lens settings should remain unchanged (particularly, auto-focus should be switched off).

The camera parameters to be optimized should be selected based on the camera and optics used. Avoid using too many free parameters, as they may reduce calibration errors but not guarantee an accurate computed model.

Outlier detection and elimination: Ideal methods for camera calibration include all calibration points in the optimization. Since the coordinates of the control points can be detected imprecisely or incorrectly in real images, e.g., due to occlusion, shadows, camera noise or movement during capture, it is necessary to remove the outliers.

CHAPTER

2 Pinhole Camera Model

The pinhole camera model defines the transformation of three-dimensional Euclidean space onto a two-dimensional image plane based on the principle of central projection. It is used to approximate real cameras. In a pinhole camera, a point in space is imaged into a picture point collinearly by a ray passing through the projection center of the camera.

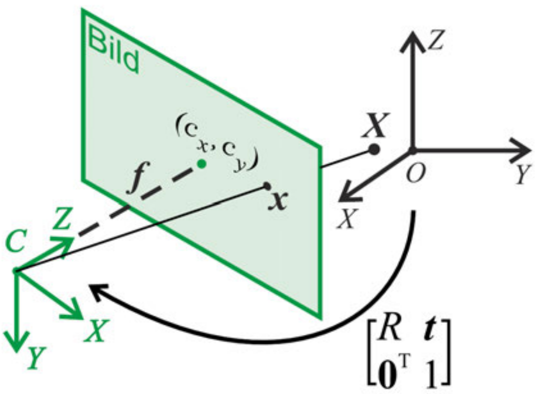


Fig. 2.1: The transformation components of a 3D space to a 2D plane, as implemented with a pinhole camera, consist of the image plane, point C (which does not lie on the image plane), the optical center, and the image distance f (the distance between point C and the image plane). The optical axis is the line passing through C and perpendicular to the image plane. It intersects the image plane at the principal point (c_x, c_y). The mapping **x** of a point in space **X** is the intersection of the optical ray (C,X) with the image plane.

The model can be derived using Figure 2.1: If we equate the 3D coordinate system of the world with that of the camera, with the origin at point C, then this projection is compactly described, according to [12], with the following equation.

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X_C} \tag{2.1}$$

The homogeneous coordinate vectors of the image point and the 3D point are denoted by $\mathbf{x} = u\cdot(x, y, 1)^T$ and $\mathbf{X}_C = (X, Y, Z, 1)^T$ respectively. They are uniquely determined up to a scaling factor. Even if the scaling factor of $\mathbf{X}_C$ is chosen to unity, the scaling factor u of $\mathbf{x}$ has to be considered explicitly. The true image coordinates x, y are obtained by dividing $\mathbf{x}$ by its third component.

The index C denotes that the 3D point $\mathbf{X}_C$ is defined in the camera coordinate system. The notation $[I|\mathbf{0}]$ stands for a 3×4 block matrix, which consists of the 3×3 identity matrix I and the column vector $\mathbf{0} = (0,\ 0,\ 0)^T$. The 3×3 matrix K

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

is referred to as the camera matrix or the calibration matrix of the pinhole camera model. It consists of the intrinsic parameters: focal length f (the distance of the image plane from the camera center), and the principal point (c_x, c_y).

If the point $\mathbf{X}$ is defined in a 3D world, and the orientation and position of the camera in this world are described with the rotation matrix R and the translation $\mathbf{t}$, then $\mathbf{X}$ must first be affinely transformed into the camera coordinate system

$$\mathbf{X}_C = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X} \tag{2.3}$$

and only then projected onto the image plane according to the equation (2.1). In summary, the well-known algebraic relationship of the pinhole camera model results

$$\mathbf{x} = K\,[R|\mathbf{t}]\,\mathbf{X}. \tag{2.4}$$

The fully expanded form of the equation is

$$u \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & \gamma & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.5}$$

with the scaling factor u mentioned above.

Note The extrinsic parameters R and $\mathbf{t}$ in equation (2.3) represent the transformation from the world coordinate system with origin O to the camera coordinate system with origin C. The translation vector $\mathbf{t}$ points from C to O, $\mathbf{t} = \overrightarrow{CO}$. The inverse transformation, from camera to world coordinates, is:

$$\mathbf{X} = \begin{bmatrix} R^T & -R^T\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_C \tag{2.6}$$

2.1 Pinhole Model Parameters

Principal Point

The principal point (c_x, c_y) is the intersection of the optical axis (axis Z_C as shown in Figure: Pinhole Camera Model) and the image plane. It is defined in the right-hand Cartesian image coordinate system, with the origin situated at the top-left corner of the image.

Note 1 The principal point is typically located near the center of the image. Exceptions can occur, for instance, when the image is asymmetrically cropped. The principal point also deviates from the center when the lens is shifted, a situation that is particularly common with DMD projectors.

Note 2 The principal point of the image is expressed in image coordinates. The X-axis of the image coordinate system runs parallel to the image rows and points from left to right. The Y-axis runs parallel to the image columns and points from top to bottom. The origin of the image coordinate system is situated at the top-left corner of the image.

Focal Length

The focal length f describes the distance of the image plane from the camera's center. For practical reasons, it is divided into two so-called effective focal lengths f_x and f_y. These correspond to the focal length f multiplied by the horizontal or vertical scaling. This value roughly corresponds to the actual focal length of the camera lens divided by the size of a sensor element (pixel). The unit is one pixel (1 px). For example, for a camera with a Sony IMX174 chip with a pixel size of $5.86\mu m$ and a lens with an 8 mm focal length, the effective image distance is $f_x = f_y = 8\,\text{mm}/0.00586\frac{mm}{px} = 1365.2px$.

Note 1 For digital cameras (CCD, CMOS), the effective focal lengths f_x, f_y should always be equal, $f_x = f_y$, because the sensor elements of such cameras are square. The only plausible exceptions for f_x and f_y to be different are as follows:

- Different scaling of the image in the horizontal or vertical direction, caused either by the anamorphic lenses or by subsequent image processing in which the image was not shrunk or stretched uniformly.
- Non-square pixels of the sensor. This occurs particularly with photographic films or recording devices with an analogue frame grabber.

2.2 Lens Distortion Model

So far, it has been assumed that the linear pinhole camera model is an exact model of the image acquisition process. In this model, the world point, the corresponding image point, and the optical center are collinear, and the objects in the scene are projected onto the image plane following the principle of central projection. This model is valid under the assumption that the image plane (sensor, film) is flat, the pixels (sensor elements) are arranged in a regular grid, and the optical elements of the lens show no distortion. The first two assumptions are true for digital cameras, but the last is not. Especially with wide-angle cameras, it is important to pay attention to lens distortion.

The Brown-Conrady model [5] is an established method for distortion correction. The model takes into account the so-called radial and tangential distortion. The first type is based on the lens's curvature. The tangential distortion models the thin prism distortion. With radial distortion, the image is scaled radially symmetrically to the center of distortion (principal point) and with tangential, it is shifted perpendicularly to the radially distorted point. The rectified coordinates are:

$$x_u = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2x^2) + 2p_2 xy \tag{2.7}$$

$$y_u = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_2(r^2 + 2y^2) + 2p_1 xy \tag{2.8}$$

where the radius $r = \sqrt{x^2 + y^2}$ defines the distance from the distortion center to the distorted point (x, y). The radial coefficients k_1, k_2, and k_3, as well as the tangential coefficients p_1 and p_2, are part of the internal camera parameters.

This correction step is carried out in normalized camera coordinates $x := X_C/Z_C$, $y := Y_C/Z_C$, i.e. it is inserted between K and $[R|\mathbf{t}]$ in eq. (2.4). Hence the distortion coefficients are independent from the image resolution.

Note Tangential distortion (p_1, p_2) is negligible with today's digital cameras. These parameters played a significant role nearly a hundred years ago due to poorer manufacturing tolerances. It was not always the case that the film or film plate was perpendicular to the optical axis of the lens. Such image errors were corrected based on the model of the thin prism. The parameters for tangential distortion $(p_1,\ p_2)$ should therefore only be taken into account in plausible cases.

CHAPTER

3 Calibration Target

A key prerequisite for accurate calibration is a precisely manufactured calibration target. A classic target provides several spatial (calibration) points with a known geometric position in space (3D target) or on a plane (2D or planar target). The calibration points and their surroundings on the target are designed so they can be reliably and automatically detected and measured based on their geometry and surrounding contrasts in the actual camera image. Ideally, targets should be easily producible in various sizes and universally applicable for multiple calibration tasks. A planar calibration target with a chessboard pattern, as illustrated in Figure 3.1, satisfies these criteria. The calibration points are positioned at the corners of the squares.

3D-EasyCalib's embedded calibration methods utilize images of such planar calibration targets. The settings for accurate detection of the target and precise measurement of the calibration points are as follows:

3.1 Grid of the Checkerboard Calibration Target

Inner Corners represent the number of inner corners in a checkerboard pattern. In Figure 3.1, these corners are marked with green dots. The first number (here 9) defines the x-axis of the target, and the second (here 8) defines the y-axis. The target's coordinate origin is marked with 0.

To unambiguously determine the orientation of the calibration target, the numbers of corners in the X and Y directions must have different parities. I.e., if the number of corners in the X-direction is odd, the number of corners in the Y-direction must be

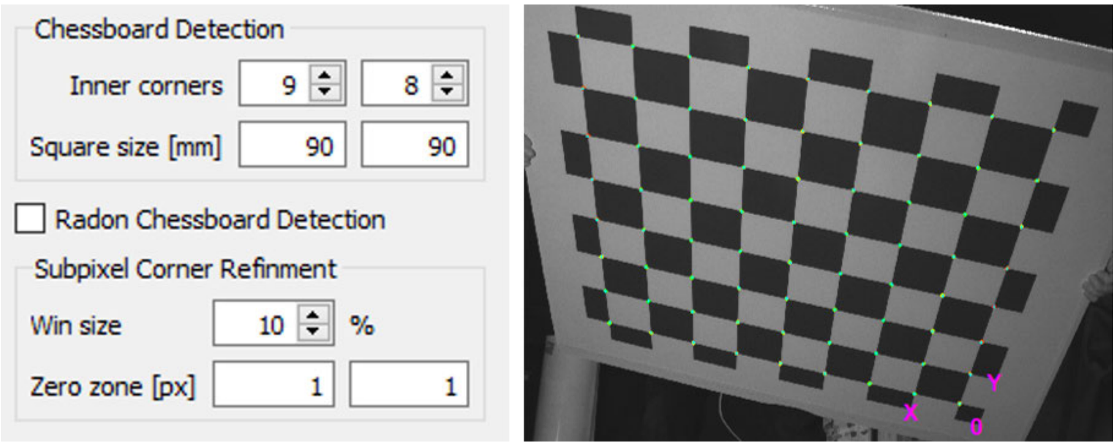


Fig. 3.1: Settings for detecting the 9 x 8 calibration target on the right and measuring the points with sub-pixel accuracy. The calibration points are marked in green, and the coordinate axes on the target are in magenta.

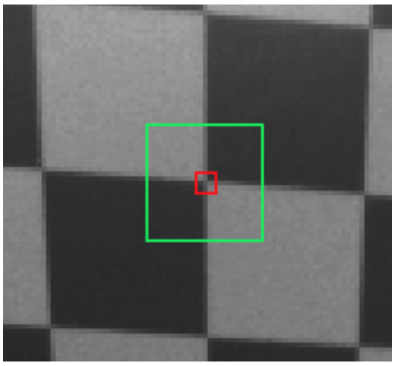
even and vice versa. This ensures that the target has exactly one axis of symmetry, and its orientation can be determined unambiguously.

Square Size [mm] denotes the size of a square on the target. This information is necessary for generating point coordinates of the calibration target. **Note:** Only the ratio (in the case of non-square fields) is crucial for the calculation of the intrinsic camera parameters - the absolute size of the squares is irrelevant. However, an accurate value of the square size is necessary for the estimation of the extrinsic parameters.

3.2 Measuring the Target Points with Sub-pixel Precision

There are two methods available for sub-pixel precise estimation of calibration points (subpixel corner refinement in Fig. 3.1): The **Radon Chessboard Detection** according to Duda [9], and the classic approach according to Förstner [10]. The Radon detector delivers superior results on noisy images and does not require any parameter setting.

If **Radon Chessboard Detection** is not selected, the control points are calculated using the classic approach. This method calculates the subpixel-precise position of the corners within a window and has the following settings:



Win Size is the window size (green) in % around the calibration point. The numerical percentage is defined as the smallest horizontal or vertical distance to the nearest calibration point.

Zero Zone is the area around the calibration point that is excluded from the calculation, with units in pixels. When entering -1, -1, all points within the search window are considered.

3.3 Considerations When Capturing Calibration Targets with a Camera

Precise calibration requires a precise and rigid calibration target. The optimal size of the target and the number of inner corners depend on the camera and the application scenario (size of the workspace). Upon request, we can design and manufacture an ideal calibration target for your application.

When capturing a calibration target with a camera, consider the following:

- The target should be fully visible in the image.
- In each shot, the target's orientation and position relative to the camera should vary.
- Vary the target's orientation around all three spatial axes.
- Capture the target at varying distances from the camera to cover the entire working area.
- The calibration points of all images should be distributed over the entire image.
- The camera lens's focus and focal length should remain unchanged during the recording.
- The number of images for a calibration should be at least 10.

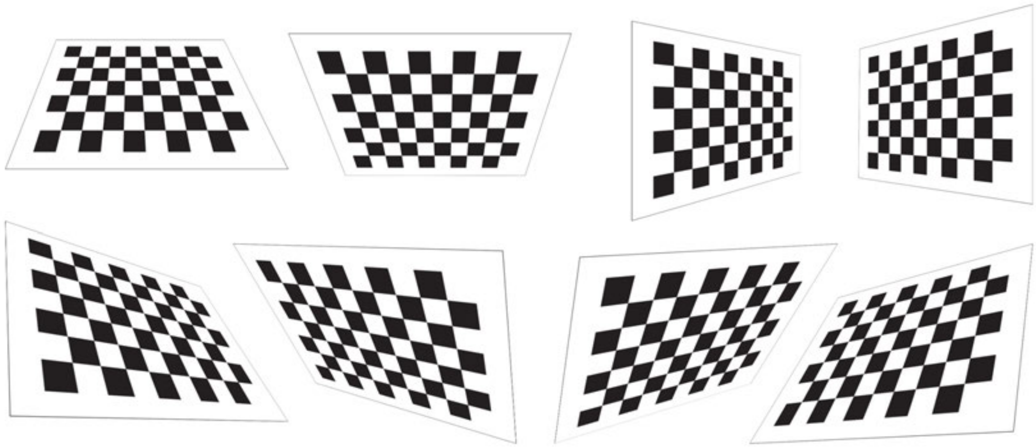


Fig. 3.2: Recommended target poses for calibration. Initially, capture the target as close as possible. Similar target poses can then be captured at larger camera-target distances.

Comment on the Selection of the Geometric Shape for the Calibration Points

Most targets utilize either circles or checkerboard patterns. With a distortion-free camera lens, it is irrelevant which type of features are used on a calibration target. Both the centers of the circles and the corners of the chess pattern can be determined with equal accuracy in a perspectively distorted image. However, if the image is severely distorted due to the lenses, then circles will be asymmetrically distorted, and determining the coordinates of the original center becomes non-trivial. This has been theoretically and experimentally proven in [17]. Therefore, we recommend using a checkerboard pattern for calibration instead of targets with circles.

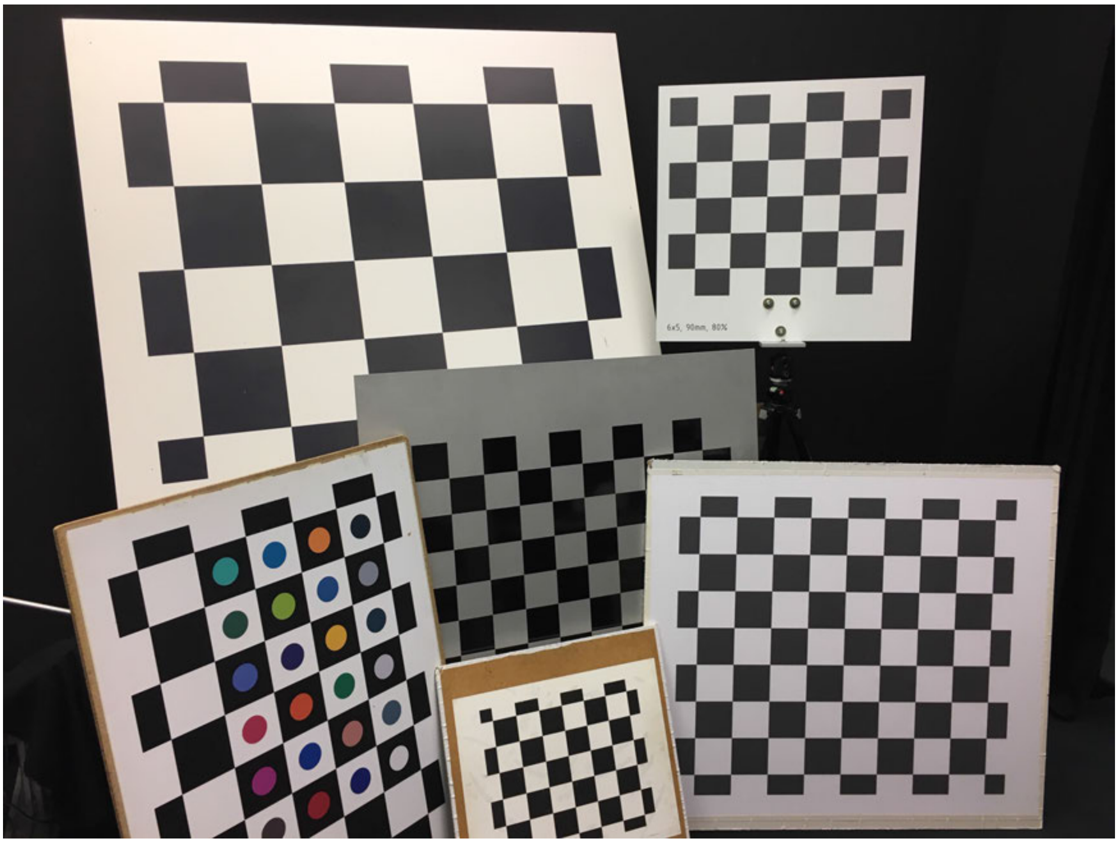


Fig. 3.3: Calibration targets for various purposes

CHAPTER

4 User interface of the 3D-EasyCalib

The main window of the program, as depicted in Figure 4.1, is divided into three areas: on the left side the added components (cameras, coordinate systems) are displayed as a list (2). The corresponding coordinate systems for the components listed on the left are visualized in the middle (3). The actions – the calibration procedures – are listed on the right (4). Projects can be saved and loaded via the File menu item (1).

(1) The File menu contains the following project-related routines:

New Project deletes all added components and creates a new project.

Load Project loads a saved project from the YML file. The specification of the format is described in section Structure of the project file.

Save Project exports the project as YML and Json file. Here all cameras in the "Objects" field are exported with respect to the selected coordinate system. This extrinsic data transforms the coordinates of the respective camera into the selected world coordinate system. Any camera in the "Objects" list can be used as the world.

(2) The cameras can be added with the button [Add Component] and the already added cameras can be deleted with the bin button. The intrinsic data for a camera can be loaded with the folder button. The presentation of the intrinsic and extrinsic parameters is described in detail after the two following sections.

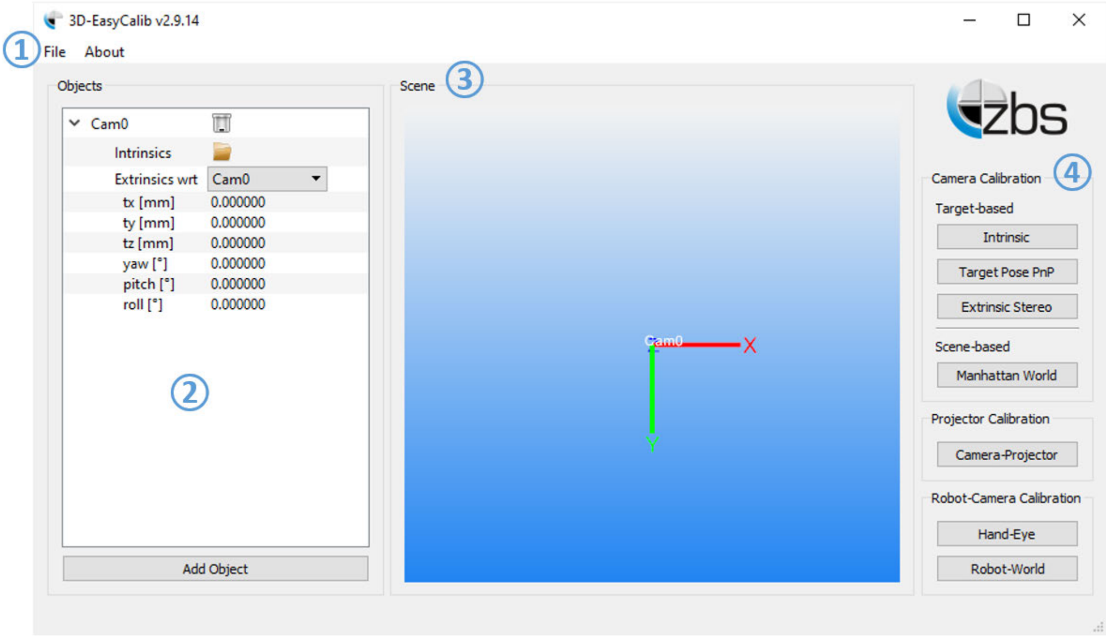


Fig. 4.1: Main components of the user interface

③ **Scene** In the middle of the main window, the components are symbolically displayed in three dimensions with the coordinate axes and a label. Navigation: left mouse rotates the view, middle mouse moves the scene parallel to the camera and the mouse scroll enlarges / reduces (zoom in / zoom out) the view. Keyboard key R resets the zoom scale.

④ Intrinsic/extrinsic calibration methods are divided into three categories: camera, projector and robot camera calibration. The camera calibration contains target-based methods for [Intrinsic] camera calibration, for determining the extrinsic parameters of a camera or the target [Target Pose PnP] and extrinsic calibration of a camera pair [Extrinsic Stereo]. Scene-based extrinsic calibration is included in the Manhattan World method. The camera-projector-calibration [Camera Projector] determines the intrinsic parameters of a projector as well as its position and orientation in relation to a camera. The robot-to-camera calibration contains two methods: the so-called [Hand-Eye] and tool-flange respectively [Robot-World] calibration.

Each procedure opens in a new window. The calibrated parameters are automatically transferred to the main window.

4.1 Structure of the project file

The YAML file (see image 4.2) contains the date of the saved calibration (ISODate) after the header (%YAML:1.0). After that come the sensor names (Sensors) and the world (World) to which all listed components are extrinsically calibrated. In other words, all coordinate systems have the same reference - the world coordinate system. The intrinsic (K_i, kc_i) and the extrinsic (R_i, T_i) parameters follow for all sensors. The i represents the order of the sensor in the sensors list. The elements of the camera matrix K_i are stored row by row as in the equation 2.2 and the order of the distortion parameters is [k_1, k_2, p_1, p_2, k_3] (see **??** lens distortion, page **??**). The R_i contains the 3×3-rotation matrix and the T_i the translation vector.

4.2 Representation of intrinsic parameters

The intrinsic parameters are shown in the object list (see fig. 4.3, left). The parameters of the camera matrix, the effective focal lengths f_x, f_y, and the principal point (c_x, c_y) are listed. These parameters are either generated from a method contained in 3D-EasyCalib or loaded from a YAML file. The file format (see Fig. 4.3) is defined as follows:

The camera matrix K is shown in one line in the data field of the object K. The order of the distortion parameters in the data field of the kc is [k1, k2, p1, p2, k3]. The decimal separator is the point and the numbers can also be written in exponential form.

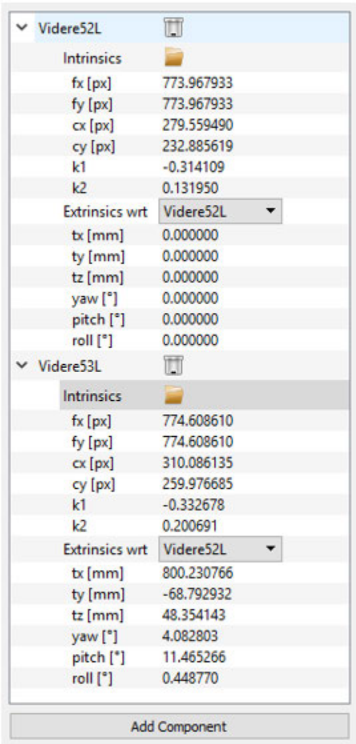
Every time an intrinsic calibration is performed, such a file with the name "Intr.yml" is always created in the same folder as the contained image data as a backup and can be loaded here.

The numerical values in the GUI can be changed as desired with a double-click. Lens distortion parameters are shown only if they are non-zero.

4.3 Representation of extrinsic parameters

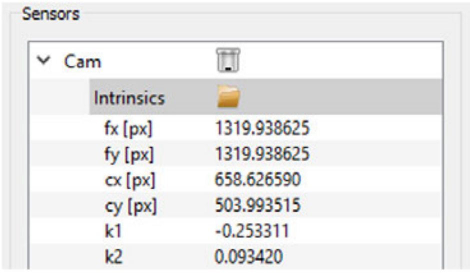
The camera coordinate system is always defined as in Fig. 2.1: Z-axis is on the optical axis and points into the image, X-axis is parallel to the image rows and Y-axis is parallel to the image columns. The distance between the coordinate center and the image plane is the focal length f.

The orientation and position of a camera in any coordinate system is described with the rotation matrix R and the translation $\mathbf{t}$. Here, R is the rotation matrix of the camera with the origin in C and is with respect to a world W. Vector $\mathbf{t}$ represents the




```
%YAML:1.0
---
ISODate: "2018-11-27T09:37:37"
Sensors:
   - Videre52L
   - Videre53L
World:
   - Videre53L
K0: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 7.7396793300000002e+02, 0., 2.7955948999999998e+02, 0.,
       7.7396793300000002e+02, 2.3288561899999999e+02, 0., 0., 1. ]
kc0: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -3.1410900000000003e-01, 1.3195000000000001e-01, 0., 0., 0. ]
R0: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 9.7755822426612149e-01, -6.9642942711658939e-02,
       1.9882097147969027e-01, 6.9777329061873700e-02,
       9.9754244890776556e-01, 6.3393198443462079e-03,
       -1.9877384767282194e-01, 7.6761421012456936e-03,
       9.8001532351478227e-01 ]
T0: !!opencv-matrix
   rows: 3
   cols: 1
   dt: d
   data: [ -7.9667692618343563e+02, 1.2479371725929736e+01,
       1.1220521201887109e+02 ]
K1: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 7.7460861000000000e+02, 0., 3.1008613500000001e+02, 0.,
       7.7460861000000000e+02, 2.5997668499999997e+02, 0., 0., 1. ]
kc1: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -3.3267799999999997e-01, 2.0069100000000001e-01, 0., 0., 0. ]
R1: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 1., 0., 0., 0., 1., 0., 0., 0., 1. ]
T1: !!opencv-matrix
   rows: 3
   cols: 1
   dt: d
   data: [ 0., 0., 0. ]
```


Fig. 4.2: Example project file for two cameras. On the left is the representation in the program and on the right is the content of the project file in YAML format.




```
%YAML:1.0
---
K: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 1.3199386251652179e+03, 0., 6.5862659004789816e+02, 0.,
       1.3199386251652179e+03, 5.0399351487326260e+02, 0., 0., 1. ]
kc: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -2.5331134076491058e-01, 9.3419898166071255e-02, 0., 0., 0. ]
```


Fig. 4.3: Representation of the intrinsic parameters, the camera matrix K and the distortion parameters in the GUI (left). The parameters can be loaded by clicking on the folder symbol. Intrinsic parameter YAML file specification (right).

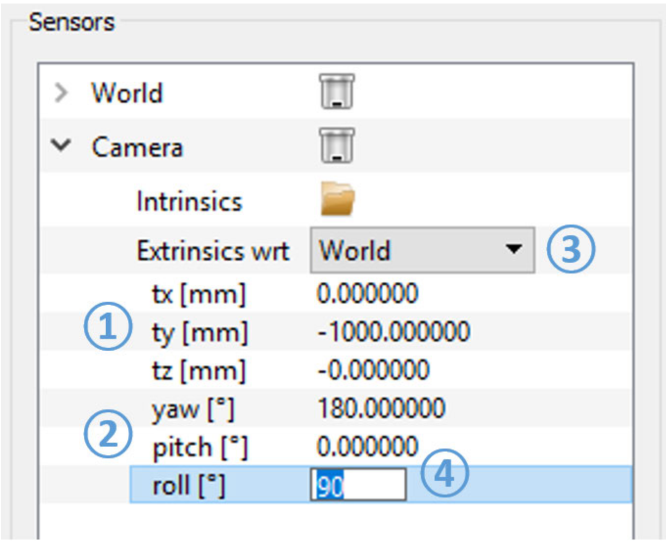


Fig. 4.4: Representation of the extrinsic parameters, the translation vector **t** and the rotation R with Euler angles.

vector from C to the origin of the world coordinate system. In other words: point **X** is transformed from the world to the camera coordinate system.

$$\mathbf{X}_C = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}$$

The extrinsic parameters shown in the GUI (see Fig. 4.4) have the following meaning:

(1) The translation vector **t** is given with the components tx, ty and tz (in millimeters) and the rotation R as yaw, pitch and roll angles.

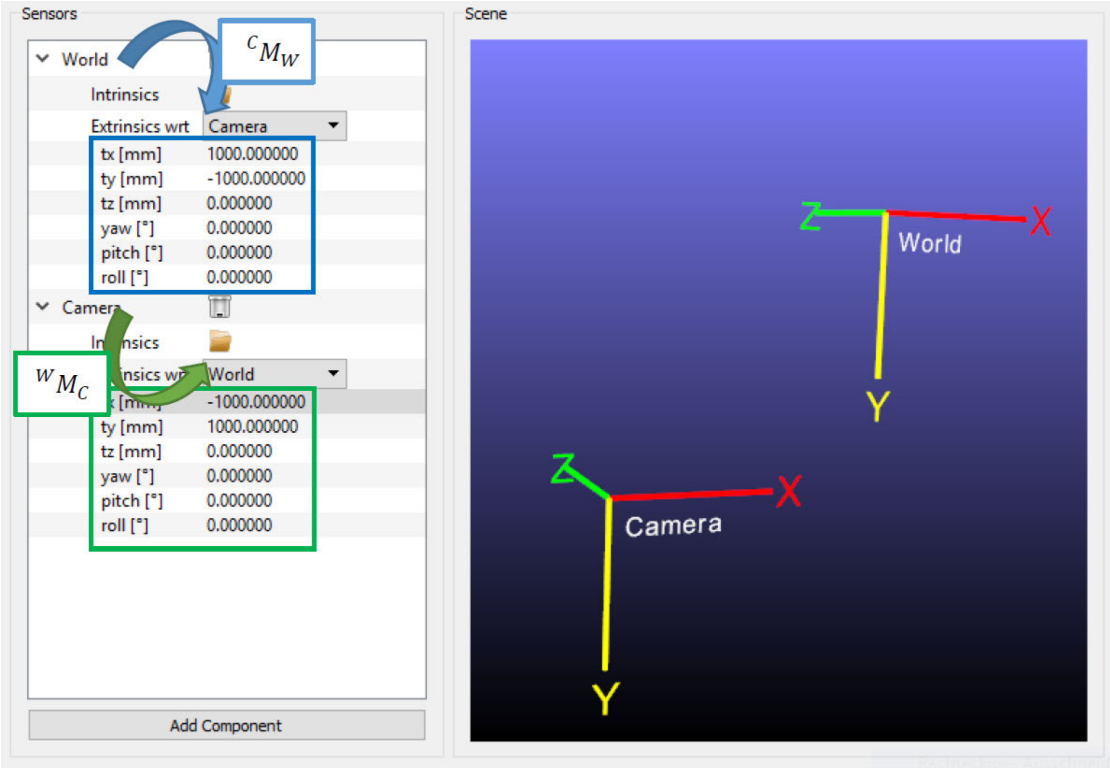


Fig. 4.5: The extrinsic parameters of the **Camera** are listed with respect to (w.r.t.) to the **World**.

(2) Euler's angles are defined according to the $z - y' - x''$ convention: yaw is the rotation around the z-axis of the reference coordinate system (here world coordinate system) by the angle yaw in degrees anticlockwise. Similarly, pitch is rotation about the y-axis and roll is rotation about the x-axis of the reference coordinate system. The order of rotations is yaw, pitch, and then roll. That is, rotation is first performed around the z-axis, then around the y-axis and finally around the x-axis.

(3) The reference coordinate system can be selected in the combo box.

(4) The values of the extrinsic parameters can be changed by double-clicking on the corresponding field and using the keyboard.

4.4 Example of a camera-to-world transformation

The following example is intended to illustrate the transformation and derive the representation of a three-dimensional point in different coordinate systems.

In the 3D-EasyCalib, the relation of a camera to the other cameras (or to a world) can be read as shown in the Figure 4.5: The extrinsic parameters of the **Camera** are defined with respect to the **World**. Once the points have been captured in the camera coordinate system, it is possible to transform them into the selected world coordinate system using the extrinsic parameters.

The following notation should be used to illustrate the world-to-camera or camera-to-world transformation for a point. Let R be written as ${}^{C}R_{W}$ and $\mathbf{t}$ as ${}^{C}\mathbf{t}_{W}$. Here the indices are to be read as "from W to C" (see also Figure 4.5). The transformation

$$ {}^{C}M_{W} = \begin{bmatrix} {}^{C}R_{W} & {}^{C}\mathbf{t}_{W} \\ \mathbf{0}^{T} & 1 \end{bmatrix} \tag{4.1} $$

transforms the point ${}^{W}\mathbf{P} = (X, Y, Z, 1)^{T}$ defined in the world coordinate system into the camera coordinate system:

$$ {}^{C}\mathbf{P} = {}^{C}M_{W} \cdot {}^{W}\mathbf{P} \tag{4.2} $$

Remark The notation is chosen because of the following easy-to-remember rule: M transforms the point "from W to C" and the points to the left and right of a transformation M must have the same subscript and superscript (here C or W):

$$ {}^{C}\mathbf{P} = {}^{C}M_{W} \cdot {}^{W}\mathbf{P} \tag{4.3} $$

This notation is also helpful for a chain transformation over several coordinate systems: two consecutive transformations must have the same subscript or superscript (here O):

$$ {}^{C}\mathbf{P} = {}^{C}M_{O} \cdot {}^{O}M_{W} \cdot {}^{W}\mathbf{P} \tag{4.4} $$

4.5 Supported image formats

3D EasyCalib supports the most common image formats such as PNG, JPEG, TIFF and BMP. All formats can be read both as 8-bit and 16-bit. The full list of supported image types is:

- Windows Bitmaps (BMP, DIB)

- JPEG files (jpeg, jpg, jpe)
- JPEG 2000 files (jp2)
- Portable Network Graphics (png)
- WebP (webp)
- Portable image format (pbm, pgm, ppm)
- Sun rasters (sr, ras)
- TIFF files (tiff, tif)

If the desired image format is not listed in the image loading dialog, please select "All files" in the combo box.

Note It is recommended to save the captured images in a lossless compressed image format such as Portable Network Graphics (PNG).

CHAPTER

5 Intrinsic Camera Calibration

The aim of an intrinsic camera calibration is to estimate the parameters that define the internal geometry and optical characteristics of a camera. These parameters include the focal length, the principal point and the lens distortion coefficients. By knowing these parameters, one can correct for the effects of lens distortion and project a 3D point in the world coordinate system onto a 2D point in the image plane. Intrinsic camera calibration is essential for many computer vision applications, such as 3D reconstruction, augmented reality, and object recognition.

The intrinsic parameters of a camera describe how the camera maps the 3D world coordinates to the 2D image coordinates. These parameters can be estimated by using the equation $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$, where $\mathbf{x}$ is the image point, $\mathbf{X}$ is the world point, K is the matrix of intrinsic parameters, and $[R|\mathbf{t}]$ is the matrix of extrinsic parameters that represent the rotation and translation of the camera. To calibrate the camera using this equation, a calibration target (chapter 3) with known geometry and dimensions is

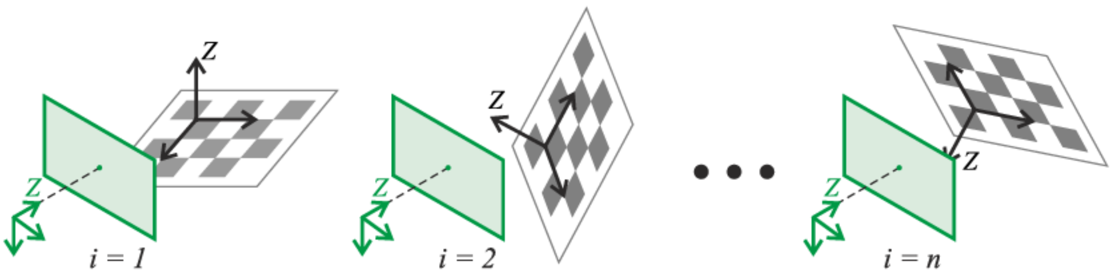


Fig. 5.1: The calibration data set with a planar target consists of several images of the target in different poses.

required. The target must be captured in different poses and orientations to obtain enough equations to solve for the unknown parameters.

The camera matrix K is estimated from images of a planar calibration pattern whose known world coordinates $\mathbf{X}_i$ and corresponding image coordinates $\mathbf{x}_i$ are used to solve the equation $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$ in homogeneous coordinates. Zhang's method [23] has been widely adopted in the community, especially due to the availability of free implementations, such as the user-friendly Matlab toolbox of Bouguet [2] and its port to OpenCV [3].

According to this approach, the target poses are estimated with the homography between the target and image plane in the first step. After that, n images of the calibration target (see fig. 5.1) and m known coordinates of calibration points $\mathbf{X}_j$ on the target and their mapping

$$\begin{aligned} \mathbf{x}_{1j} &= K[R_1|\mathbf{t}_1]\mathbf{X}_j, \\ \mathbf{x}_{2j} &= K[R_2|\mathbf{t}_2]\mathbf{X}_j, \\ &\dots \\ \mathbf{x}_{nj} &= K[R_n|\mathbf{t}_n]\mathbf{X}_j, \end{aligned} \tag{5.1}$$

the following functional

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{x}_{ij} - \mathbf{x}(K, R_i, \mathbf{t}_i, \mathbf{X}_j)\|^2 \tag{5.2}$$

is minimized.

5.1 Reprojection error

The Euclidean distances contained in the (5.2) between the calibration points projected according to the model $\mathbf{x}(K, R_i, \mathbf{t}_i, \mathbf{X}_j)$ and the points $\mathbf{x}_{ij}$ measured in the image are called *reprojection errors*. This error metric, given for each calibration point and as a total error (RMS) for each calibration target pose, is an indication of the quality of a calibration and is displayed accordingly in the toolkit, see Figure 5.4. Since the calibration points can be detected imprecisely or incorrectly, e.g. due to occlusion, shadows, camera noise or movement during image acquisition, potential outliers are color-coded. These points can then be manually removed from the calibration data set.

The GUI for intrinsic camera calibration (see Figure 5.2) consists of the following elements:

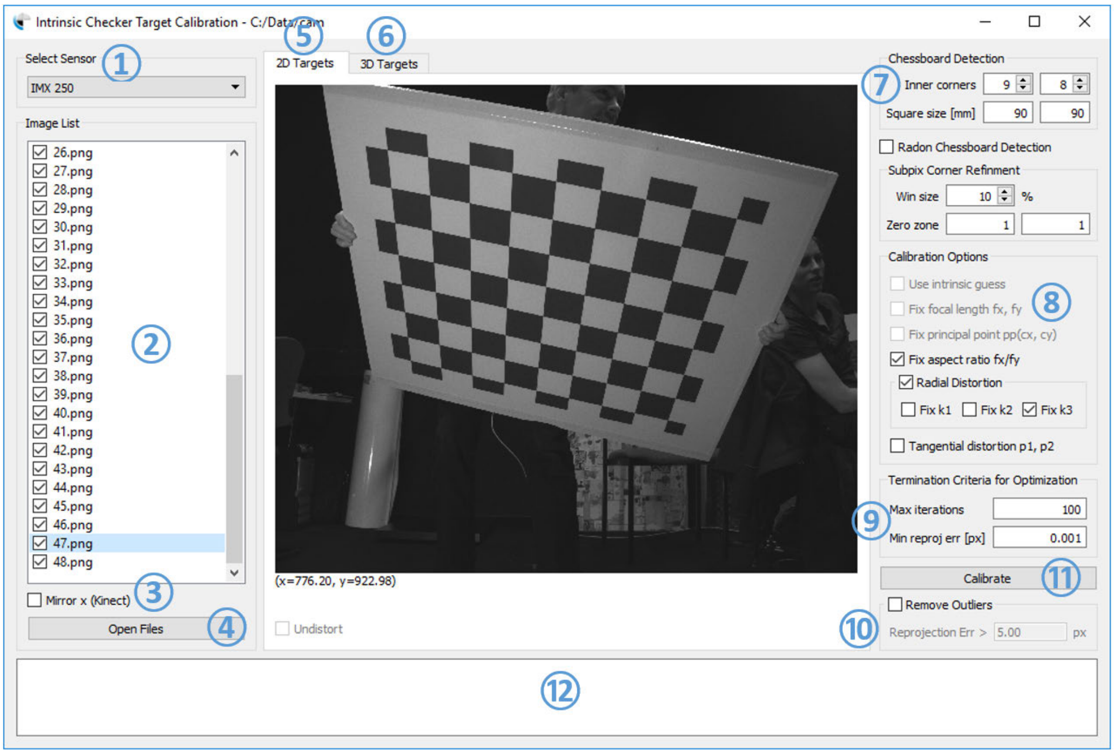


Fig. 5.2: The user interface includes the input (left), setting parameters (right), visualization of the target and the calibration points (middle) and logging in the log window (below).

5.2 Input

① **Select Sensor:** The camera to be calibrated should be selected in the combo box. The components added in the main window (chapter 4) can be selected here.

② **Image List:** The loaded images are shown with their file names in the image list. A selected image is displayed in the center of the window. Individual images can be excluded from the calibration with the check box in front of the file name.

③ **Mirror x (Kinect)**: This setting flips the images horizontally. This is necessary for some cameras that output the image mirrored, such as the Kinect2. If the image remains unchanged, it is defined in a left-handed coordinate system. Extrinsic calibration of the cameras with mixed left-handed and right-handed coordinate systems will result in invalid parameters. The 3D-EasyCalib always uses the right-handed coordinate systems.

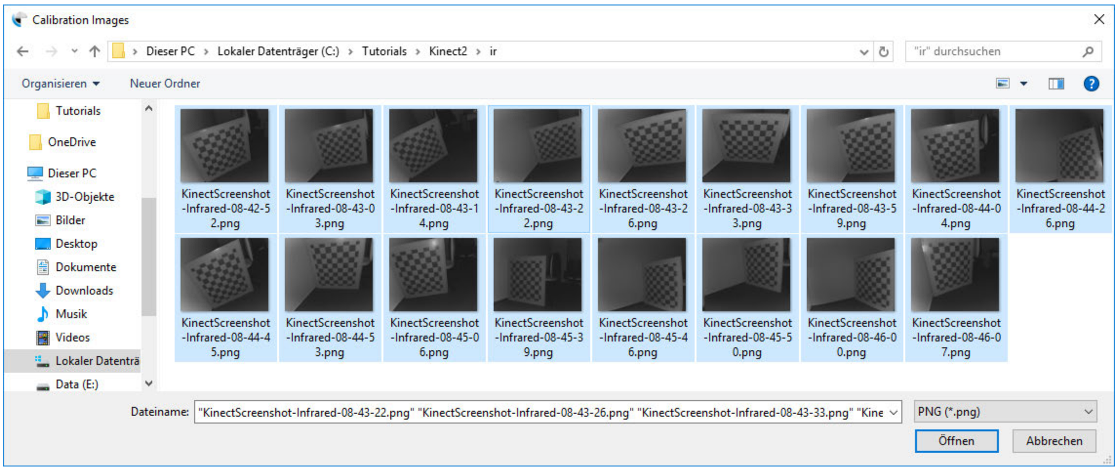


Fig. 5.3: Example images for an intrinsic calibration. Images are loaded into the program by selecting them.

④ **[Open Files]:** Opens the Load Image dialog to load the images of the calibration target. The supported image formats are listed in the section 4.5.

5.3 Visualization

The displayed images can be enlarged / reduced with the mouse wheel and rotated in the 3D view with the left mouse button.

⑤ **2D Targets:** The selected image is shown in this area. The image coordinates of the mouse pointer are shown below the image. If the distortion parameters are present, the image can be displayed undistorted with the **[x] undistort** check box. After the calibration, the reprojection errors for the target are listed: RMS over all points on the target (left) as well as minimum (middle) and maximum (right) error of the target (see Fig. 5.4 below).

⑥ **3D Targets:** After calibration, the calibration targets and the camera are displayed in 3D. The color of the target corresponds to the reprojection error regarding the min. and max. error across all calibration targets. The X and Y axis of the target in the 3D view are shown with a thicker line.

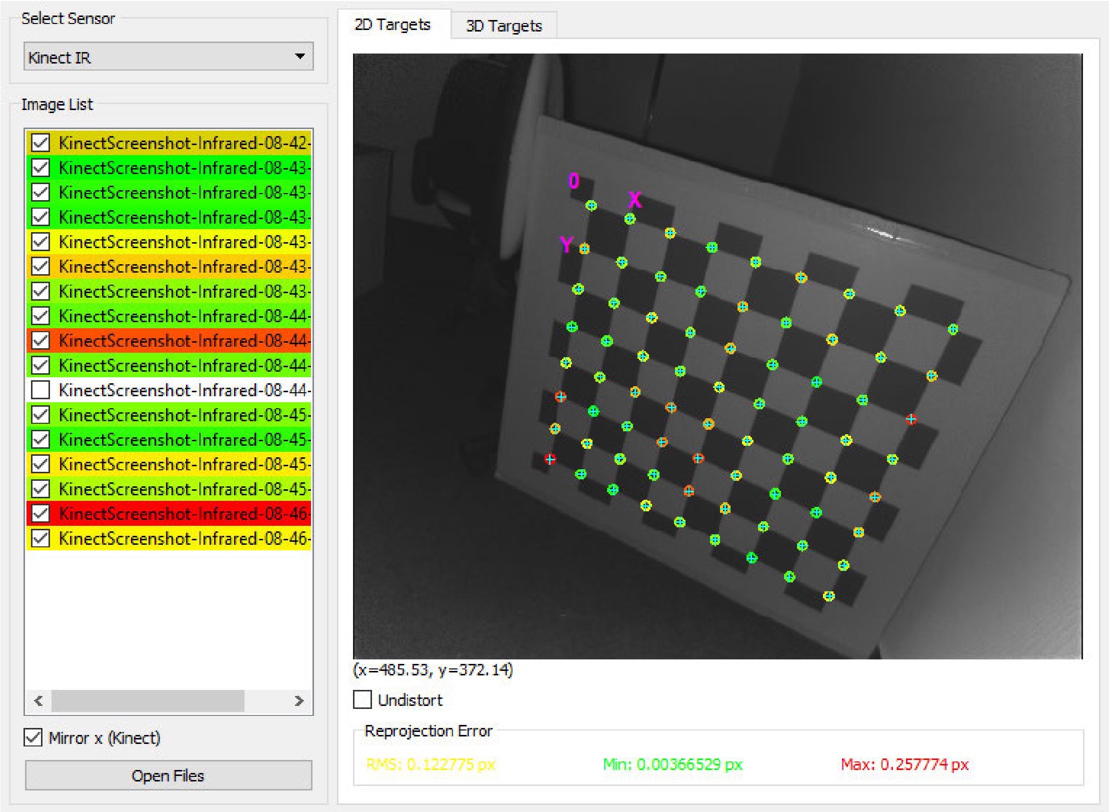


Fig. 5.4: The list of calibration images can be seen on the left. These are colored from green (min. error) to red (max. error) according to the total reprojection error of a target. On the right is the image of a selected target pose with detected calibration points (circles) and points calculated according to the model (crosses). The color of the circles represents the reprojection error (section 5.1). The image in the GUI can be zoomed in/out with the mouse wheel.

5.4 Settings

7. **Chessboard Detection:** The parameters for the recognition of the calibration target are described chapter 3.

8. **Calibration Options:** The options with "Fix", such as **Fix focal length** mean that the parameters in question should not be optimized - they remain constant (fixed). If the setting is deselected, the applicable parameters are taken into account during optimization.

   - **Use intrinsic guess**: Use internal camera parameters as initial values for optimization. This option is enabled after a calibration has already been carried out. It is also enabled when the parameters for the selected camera have been loaded in the main window.

- **Fix focal lenght fx, fy**: The effective focal length remain unchanged (fixed) during the optimization and are equal to the initial values.
- **Fix principal point pp(cx, cy)**: If activated, the principal point remains unchanged (same as the initial value) during the optimization.
- **Fix aspect ratio fx/fy**: If activated, the ratio of the focal lengths remains fixed during the optimization.
- **Radial Distortion**: If checked, the radial distortion is estimated, otherwise all radial distortion parameters are set equal to zero and fixed. The theory is explained in the **??**.
- **Fix k1, Fix k2, Fix k3**: Fixes (checked) or optimizes (unchecked) the individual radial coefficients.
- **Tangential distortion p1, p2**: If activated, the tangential coefficients are estimated during optimization, otherwise they are neglected.

(9) **Termination Criteria for Optimization:** If the reprojection error for all target images changes by less than **epsilon** in two consecutive iteration steps or if the number of iterations (**Max Iterations**) is exceeded, the optimization is finished.

(10) **Remove Outliers:** After the calibration has been carried out once, the individual calibration points on the target can be excluded from the optimization with the criterion "**Reprojection Err >**". Outlier elimination takes place in two steps: after calibration, outliers that are larger than the set error are marked and then removed during recalibration.

(11) **Calibrate:** The nonlinear minimization problem (5.2) is solved with the Levenberg-Marquard algorithm by minimizing the reprojection error (see section 5.1).

5.5 Output

After the calibration, the calculated intrinsic parameters are transferred to the main window (section 4.2), saved as **intr.yml** in the folder of the loaded calibration images and output in the log window.

(12) **Log-Window:** Serves to log the events during the calibration process (see fig. 5.5). Successfully detected targets are represented with a green number, those that

```
Detecting chessboards:
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Calibration in progress ...
Calibration completed with rms reprojection error: 0.114604
Intrinsics:
fx 367.7249 +/- 0.2522
fy 367.7249 +/- 0.2522
cx 251.6726 +/- 0.1075
cy 204.3253 +/- 0.1302
Distortion coefficients:
k1 0.0834 +/- 0.0011
k2 -0.2042 +/- 0.0019
Intrinsics saved to C:/Tutorials/Kinect2/ir/Intr.yml
```

Fig. 5.5: The events during the calibration process are logged in the log window. A red number under the **Decoding cbesshoards** means a failed target detection. After the optimization is completed, the intrinsic parameters are output. The deviations (+/-) correspond to three times the standard deviation 3σ.

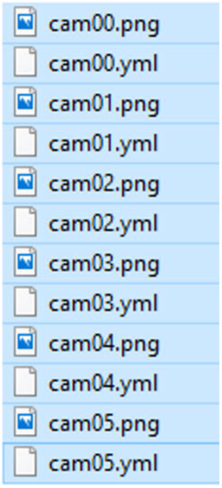
were not recognized with a red number and the deselected targets with a gray number. After the calibration, the RMS of the reprojection error as well as the optimized intrinsic parameters are output. The deviations (+/-) correspond to three times the standard deviation 3σ. The intrinsic parameters are transferred to the main window and saved to the “Intr.yml” file. This YML file can be used in the main window to load the intrinsic parameters.

5.6 Use of own calibration points or a 3D target

If you are using a custom target (2D or 3D) and the calibration points have been estimated using your own method, these points can also be loaded into 3D-EasyCalib and used for the calibration.

In addition to each image, there should also be a file with the same prefix as the image itself and with the file extension "yml" (see Figure 5.6 on the left). The content for an example YML file is given in the Figure 5.6 on the right and consists of two data sets, "corners" and "obj". The image positions of the calibration points ($\mathbf{x}$ in the $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$) are listed in "corners", where x- and y-coordinates should be written one after the other. Data set "obj" contains the X, Y and Z coordinates of the corresponding world points ($\mathbf{X}$ in the $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$), with units in millimeter. The corresponding points are color coded (cyan, magenta, orange) for the first three datasets in the Figure 5.6. The "rows" contain the number of control points (here 30) and "dt" the data type, namely two-dimensional (2f) and three-dimensional (3f) points with coordinates as float numbers.

If a 3D target is used, meaning the Z-coordinate in the "obj" points is not zero, then initial values for the focal length f and the principal point must be entered in the main window. For the focal length, for example, the focal length of the lens divided by the




```
%YAML:1.0
---
corners: !!opencv-matrix
   rows: 30
   cols: 1
   dt: "2f"
   data: [ 2.76297119e+02, 1.46180212e+03, 2.91102478e+02,
       1.21454382e+03, 3.04257599e+02, 9.95825989e+02, 3.16484894e+02,
       8.01452087e+02, 3.27687408e+02, 6.28003540e+02, 3.37811249e+02,
       4.71902100e+02, 5.26938477e+02, 1.44395435e+03, 5.26174500e+02,
       1.19801941e+03, 5.25727661e+02, 9.79980774e+02, 5.25601868e+02,
       7.86432068e+02, 5.25740723e+02, 6.13521667e+02, 5.25862488e+02,
       4.58380035e+02, 7.77427368e+02, 1.42616223e+03, 7.61643311e+02,
       1.18142297e+03, 7.47361572e+02, 9.64187988e+02, 7.34912476e+02,
       7.71445435e+02, 7.23666504e+02, 5.99495728e+02, 7.14069702e+02,
       4.45431366e+02, 1.02769104e+03, 1.40756982e+03, 9.96760803e+02,
       1.16416382e+03, 9.68962952e+02, 9.48342651e+02, 9.44198059e+02,
       7.56694275e+02, 9.21959351e+02, 5.85559143e+02, 9.01866516e+02,
       4.32524963e+02, 1.27697461e+03, 1.38816992e+03, 1.23112585e+03,
       1.14655542e+03, 1.19001501e+03, 9.32611511e+02, 1.15270496e+03,
       7.42447632e+02, 1.11935254e+03, 5.72307617e+02, 1.08976538e+03,
       4.19871368e+02 ]
obj: !!opencv-matrix
   rows: 30
   cols: 1
   dt: "3f"
   data: [ 0., 0., 0., 15., 0., 0., 30., 0., 0., 45., 0., 0., 60., 0.,
       0., 75., 0., 0., 0., 15., 0., 15., 15., 0., 30., 15., 0., 45.,
       15., 0., 60., 15., 0., 75., 15., 0., 0., 30., 0., 15., 30., 0.,
       30., 30., 0., 45., 30., 0., 60., 30., 0., 75., 30., 0., 0., 45.,
       0., 15., 45., 0., 30., 45., 0., 45., 45., 0., 60., 45., 0., 75.,
       45., 0., 0., 60., 0., 15., 60., 0., 30., 60., 0., 45., 60., 0.,
       60., 60., 0., 75., 60., 0. ]
```


Fig. 5.6: Example of an image input with pre-detected calibration points. The data set "corners" contains the calibration points in image coordinates (x, y) and the data set "obj" contains the coordinates on the calibration target (X, Y, Z).

size of a pixel on the camera chip could be used. The image center can be entered as the principal point. In addition, the "Use intrinsic guess" option must be selected in the "Calibration Options".

CHAPTER

6 Extrinsic Stereo Calibration

The extrinsic stereo calibration is a process that aims to find the geometric relationship between two cameras capturing images of the same scene from different viewpoints. This relationship can be described by the relative position **t** and orientation R of camera C_1 with respect to camera C_2, which define the transformation that maps a point in the coordinate system of camera C_1 to the corresponding point in the coordinate system of camera C_2. Figure 6.1 illustrates this concept with an example of two cameras and their coordinate systems.

Analogous to the calculation of target poses for intrinsic camera calibration (see chapter 5), the target is simultaneously captured by both cameras and its pose is

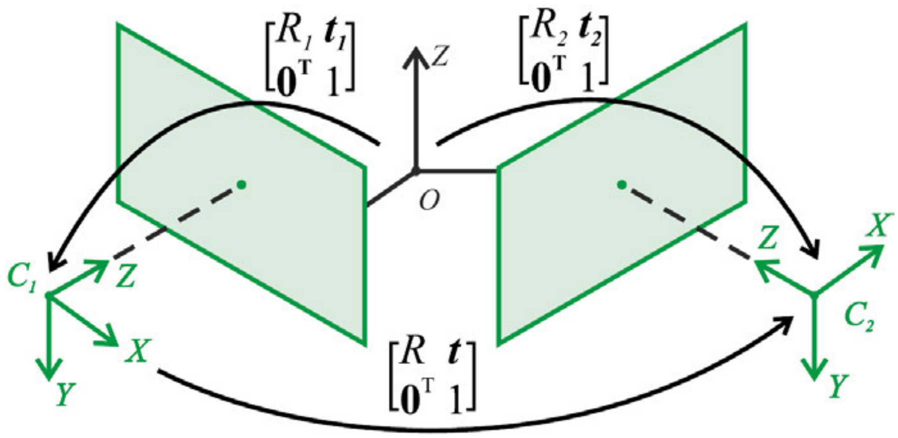


Fig. 6.1: Relative position and orientation of camera C_1 with respect to camera C_2

calculated from the views of both cameras, as well as their relative poses to each other (see section 6.1). The resulting extrinsic parameters R and $\mathbf{t}$ transform points from the coordinate system of the first camera (X_1) to the coordinate system of the second camera (X_2).

$$\mathbf{X_2} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X_1} \tag{6.1}$$

The nonlinear optimization is similar to the intrinsic camera calibration and the parameters R and $\mathbf{t}$ are optimized in such a way that the sum of the reprojection errors in both images becomes minimal.

Requirements for a stereo calibration:

- The position and orientation of the two cameras remain unchanged during acquisition of the calibration target.
- The calibration target is captured completely and at the same time by the two cameras.

Note: Ideally, the cameras should already have been intrinsically calibrated, so that only the rotation and translation are determined during this process. If the intrinsic parameters are unknown, they must be estimated and optimized prior to the stereo calibration.

Note: Additional methods can be used to estimate the relative position and orientation of two cameras, depending on available information and assumptions. One common method is to use point correspondences between images captured by the two cameras, and then compute the fundamental matrix or the essential matrix that relates them. However, both matrices have multiple possible solutions, and additional constraints or methods are needed to select the correct one. One can use other features or information, such as lines, planes, or vanishing points, to estimate the relative pose of the cameras, as an alternative to chessboard targets.

6.1 Extrinsic parameter R and t

If a world point $\mathbf{X}$ on the target is viewed in the first and then in the second camera coordinate system, the transformed points are (see Figure 6.1):

$$\mathbf{X}_1 = \begin{bmatrix} R_1 & \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X} \tag{6.2}$$

$$\mathbf{X}_2 = \begin{bmatrix} R_2 & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X} \tag{6.3}$$

The desired transformation (6.1) can be derived by solving the equation (6.2) for **X** and inserting it into (6.3):

$$\begin{aligned} \mathbf{X}_2 &= \begin{bmatrix} R_2 & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} R_2 & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} R_1^T & -R_1^T \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 &= \begin{bmatrix} R_2 R_1^T & \mathbf{t}_2 - R_2 R_1^T \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_1 \end{aligned} \tag{6.4}$$

The searched for parameters, rotation R and translation $\mathbf{t}_1 = \overrightarrow{C_1 C_2}$ are thus:

$$\begin{aligned} R &= R_2 R_1^T \\ \mathbf{t} &= \mathbf{t}_2 - R\mathbf{t}_1 \end{aligned} \tag{6.5}$$

6.2 Input

The GUI for extrinsic camera calibration (see Fig. 6.2) is similar to the GUI for internal calibration, with the difference that the input and the visualization have been expanded to include the second camera. The setting parameters for the optimization are extended by the intrinsic parameters of the second camera. The user interface consists of the following elements:

① **Select Sensors:** The cameras to be calibrated can be selected in the combo box. The components added in the main window can be selected here.

② **Image List:** The loaded images are shown with their file names in the image list. When an image is selected, the corresponding image pair is then displayed in the center of the window. With the check box in front of the file name, individual images can be excluded from the calibration. Deselecting a check box in one column will also ignore the associated image in the second column during calibration.

After calibration, each entry in the list is color-coded according to the reprojection error. Red color corresponds to the maximum and green to the minimum error.

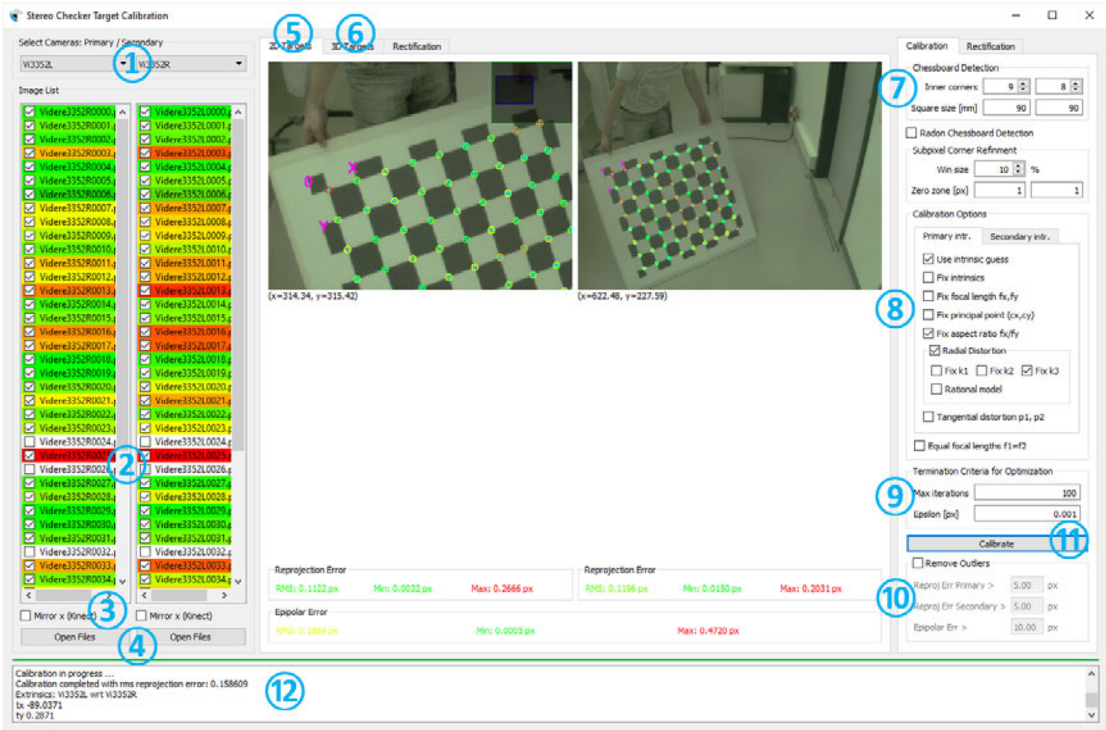


Fig. 6.2: User interface to calibrate two cameras. See text for a detailed description of the user interface.

③ **Mirror x (Kinect):** This setting flips the images horizontally. This is necessary for some cameras that output the image mirrored, such as the Kinect2. If the image remains unchanged, it is defined in a left-handed coordinate system. Extrinsic calibration of the cameras with mixed left-handed and right-handed coordinate systems will result in invalid parameters. The 3D-EasyCalib always uses the right-handed coordinate systems.

④ **[Open Files]:** The images for the respective camera are loaded with the [Open Files] button. The images should load in the same order for both cameras. The supported picture formats are listed in section 4.5.

6.3 Visualization

⑤ **2D Targets:** The selected image pair is shown in this area. The image coordinates of the mouse pointer are shown under the respective image. After the calibration, the reprojection errors for the target are listed under the respective image (see

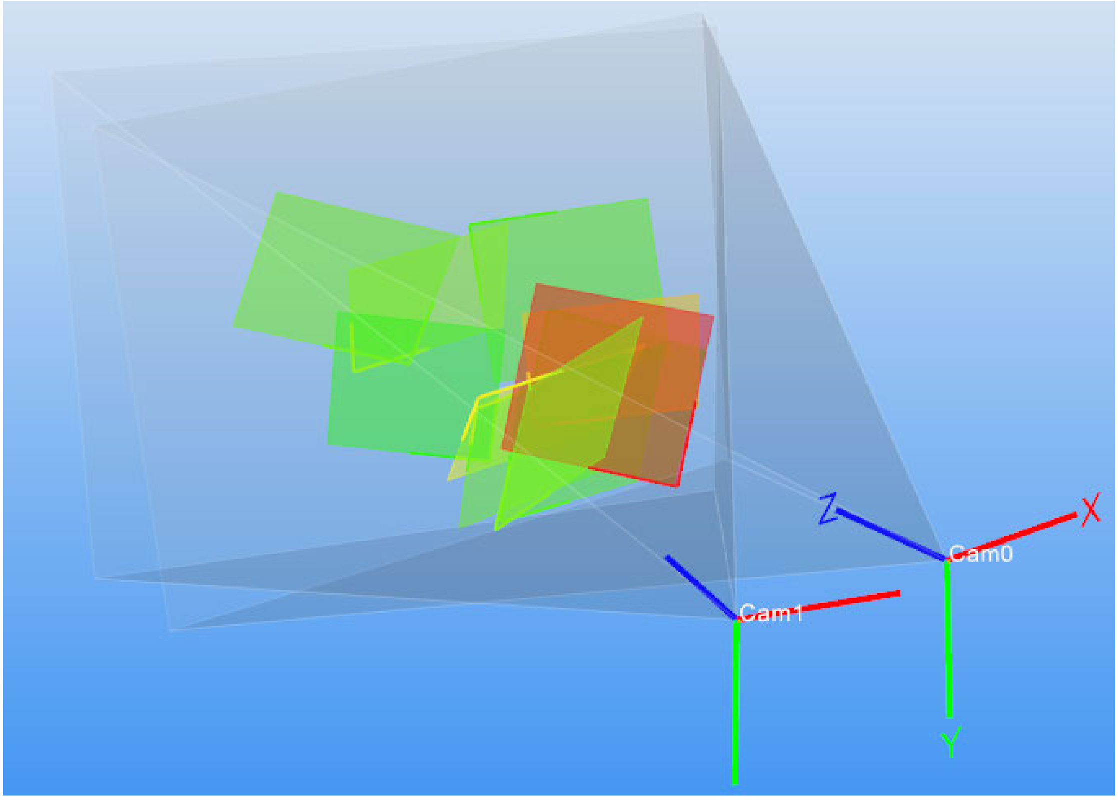


Fig. 6.3: 3D representation of the two cameras and the calibration targets. Green means a small error and red means a big error.

Fig. 6.2): RMS over all corners on the target (left) as well as minimum (middle) and maximum (right).) error of the target.

Below that, the epipolar error is output across all calibration points on the two images. Epipolar error is the smallest distance between a calibration point in one image and the associated epipolar line in the second image. In particular, these errors represent the effects of the optimized position and orientation of the cameras with respect to one another.

6 **3D Targets:** After calibration, the calibration targets and the cameras are displayed in three dimensions (see Figure 6.3). The color of the target corresponds to the size of the reprojection error averaged over the two cameras. The X and Y axis of the target in the 3D view are illustrated with a thicker line.

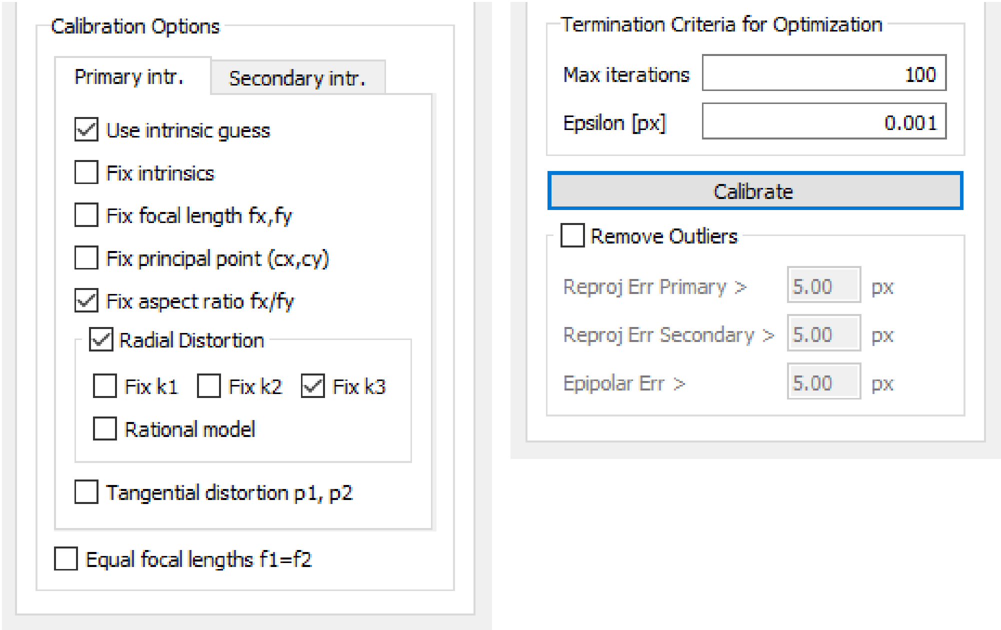


Fig. 6.4: Settings for the stereo calibration

6.4 Settings

⑦ **Chessboard Detection:** The parameters for the detection of the calibration standard are described in the chapter 3.

⑧ **Calibration Options:** The settings are separate for the primary (**Primary Intr.**) and secondary camera (**Secondary Intr.**) and can be configured independently of each other.

The following settings are available for parameter optimization. The options with **Fix**, such as **Fix focal length**, mean that the relevant parameters are not optimized - they remain constant (fixed). If the setting is deselected, the applicable parameters are taken into account during optimization.

- **Use intrinsic guess:** use known intrinsic parameters as initial values for the optimization. This option is enabled after a calibration has already been carried out or when the parameters for the corresponding camera have been loaded in the main window.
  If **use intrinsic guess** is not activated, the intrinsic calibration (chapter 5) is first carried out for the applicable camera.
- **Fix intrinsics:** All intrinsic parameters remain unchanged and are equal to the initial values. This option is only available if **use intrinsic guess** is

enabled. If the cameras were calibrated precisely, then this option should be selected.

- **Fix focal length fx, fx:** The effective focal lengths remain unchanged (fixed) during optimization and are equal to the initial values.
- **Fix principal point (cx,cy):** If activated, the principal point remains unchanged (same as the initial value) during the optimization.
- **Fix aspect ratio fx/fy:** If activated, the ratio of the image widths remains fixed during the optimization.
- **Same focal lengths f1=f2:** Only to be used if the two cameras and lenses are identical.
- **Radial Distortion**: If selected, the radial distortion is estimated, otherwise the radial distortion parameters are all set to zero and fixed. The theory and suggestions for choosing the parameters are described in **??**.
- **Fix k1, Fix k2, Fix k3:** fixes the individual radial coefficients (checked) or optimizes them (unchecked).
- **Tangential distortion p1, p2:** If activated, the tangential coefficients are estimated during optimization, otherwise not. In general, this option is not recommended for today's digital cameras.

(9) **Termination Criteria for Optimization:** If the reprojection error for all target images changes by less than **epsilon** in two consecutive iteration steps or if the number of iterations (**Max Iterations**) is exceeded, the optimization is finished.

(10) **Remove Outliers**: After calibration, the individual calibration points on the target can be excluded from optimization with the criteria **Reprojection Err Primary** > and **Reprojection Err Secondary** >. "Primary" and "Secondary" designate the first and second cameras listed in the **Select Sensors** area. The value in **Epipolar Err** > removes the corresponding point pairs that exceed the criterion. Outlier removal is implemented in two steps: after calibration, the outliers that meet this condition are marked and then removed during recalibration.

(11) **Calibrate** runs the optimization. The parameters are optimized in such a way that the sum of the reprojection errors in the corresponding image pairs is minimal.

6.5 Output

After the calibration, the computed extrinsic (6.5) and, if applicable, intrinsic parameters are transferred to the main window (section 4.2), saved in a file **stereo.yml** in the folder of the loaded calibration images and output in the log window.

(12) **Log Window:** is used to log events during the calibration process. Successfully detected targets are represented with a green number, those that were not recognized with a red number and the deselected targets with a gray number. After calibration, the RMS of the reprojection error and the optimized extrinsic parameters are output.

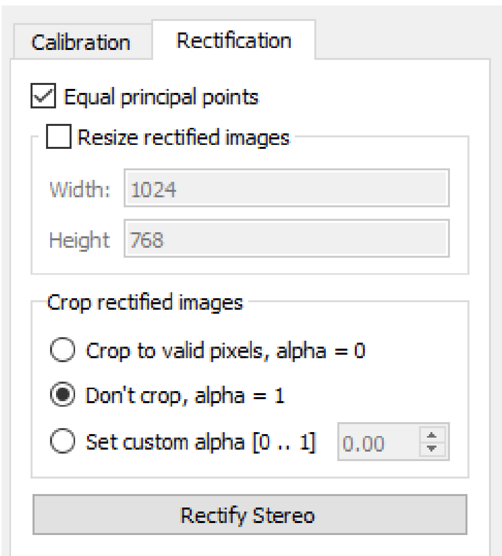
Note: The intrinsic parameters, including the distortion parameters, can be optimized independently of the other camera. This is necessary for a pair of cameras with different optics, image resolution or modality (e.g. thermal and visual cameras). This functionality is usually not available with alternative programs for stereo calibration.

6.6 Stereo Rectification

In a stereo correspondence analysis, the goal is to find matching points or regions in a pair of images taken by two cameras from different viewpoints. This task can be simplified if the search for similar structures is restricted to the same rows (or columns) of the images. This condition is met in a standard stereo configuration, where the two cameras have the same intrinsic parameters, and are aligned such that their optical axes are parallel and their image planes are coplanar. In this case, the epipolar lines, which are the lines of intersection of the image planes with the plane defined by a scene point and the two camera centers, coincide with the image rows (or columns). However, in practice, the cameras may have different intrinsic parameters or orientations, which makes the epipolar lines skewed. To overcome this problem, one can apply a geometric transformation to the images, using the known internal camera parameters and the relative rotation and translation between the cameras, to make them appear as if they were taken with the standard stereo configuration. This transformation, which is realized by a perspective projection that preserves straight lines, is called rectification. Rectification makes the stereo correspondence analysis easier and faster, since it reduces the search space from two dimensions to one dimension.

In 3D-EasyCalib™, analogous to the method of Bouquet [2], the respective image is first projected back onto the focal plane, rotated accordingly and projected again onto the common image plane with a camera matrix that is the same for both images.

Fig. 6.5: Result of image rectification and rectification with drawn epipolar lines (below) of the input images (above).



Equal principal points sets the two principal points of the rectified images to a common point. This simplifies the 3D calculation from the disparity image, for example with the function **reprojectImageTo3D** from OpenCV.

Resize rectified images Scales the image widths of the rectified images so that the rectified images have a different image size.

Crop rectified images Parameter alpha defines how much of the rectified (non-rectified) image is still visible. Alpha equal to 1 means that all pixels of the original images from the cameras are preserved in the rectified images (see example Figure 6.5). Alpha = 0 means that the warped images are enlarged and shifted so that only valid pixels are visible (no black areas after transformation).

CHAPTER

7 Camera Projector Calibration

A camera projects a 3D scene onto a 2D image plane, while a projector maps a 2D planar pattern onto a 3D space. Therefore, projectors are often referred to as, and modeled like, inverse cameras. The mathematical theory underlying image formation remains the same for both a camera and an inverse camera (projector). Thus, the same calibration algorithms can be applied. The primary challenge in projector calibration lies in accurately pinpointing the calibration points in a projector image.

This problem can be elegantly addressed with 3D-EasyCalib™ using a camera that ...

CHAPTER

8

Robot/World and Tool/Flange Calibration

In a camera-to-robot calibration, the relative position and orientation of a stationary camera with respect to a robot coordinate system are determined. For this purpose, the calibration target should be attached to the end effector of the robot. Then, the target should be moved to different poses with the robot and captured with the camera. The poses of the target in both the camera and the robot coordinate systems are used to estimate the transformation between them.

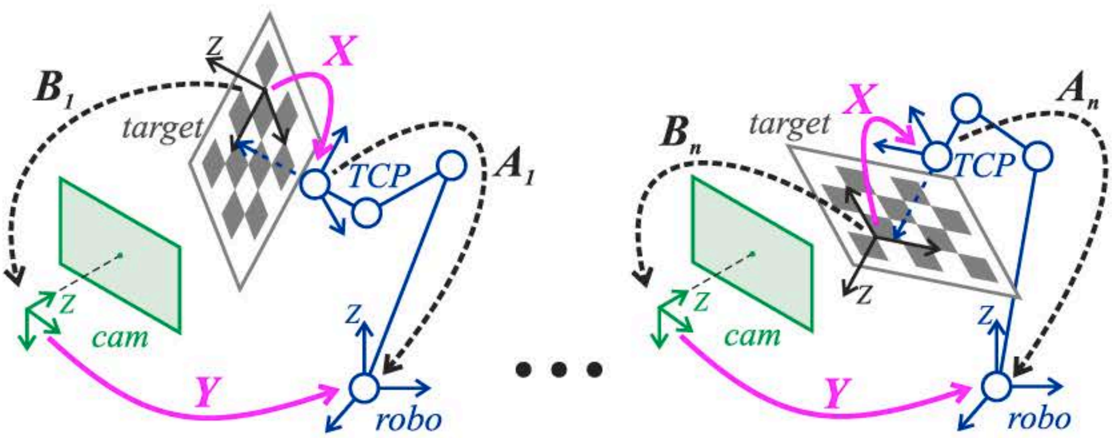


Fig. 8.1: Illustration of the transformations in the robot/world, tool/flange problem.

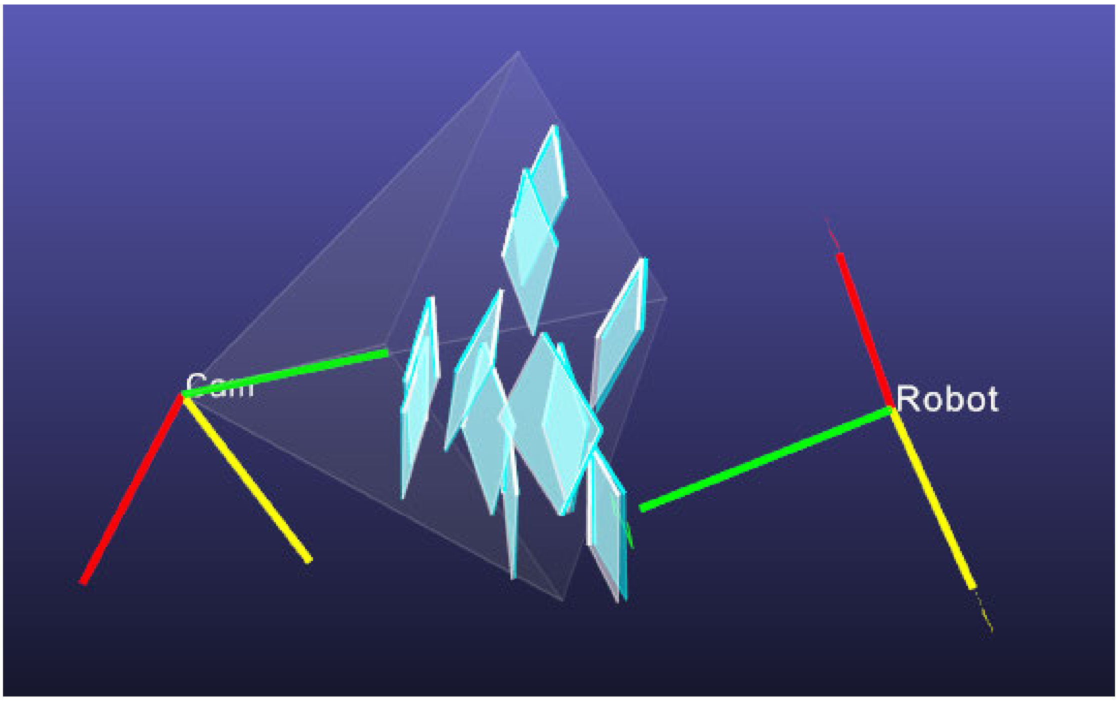


Fig. 8.6: 3D view of the robot, camera coordinate systems and calibration targets after calibration. The cyan targets represent the target poses detected by the camera, which were transformed into the robot's coordinate system (R_Y, t_Y). The white targets were first transformed with the TCP data and then with the offset to the TCP (R_X, t_X). This is the graphical representation of the 3D error defined above. Ideally, the white and cyan rectangles overlap perfectly.

CHAPTER

9 Hand/Eye Calibration

The problem of hand-eye calibration is to estimate the rigid transformation (rotation and translation) between a camera mounted on a robot actuator and the actuator itself. This transformation describes the relative position and orientation of the camera and the tool coordinate systems, which are attached to the camera and the actuator respectively. The transformation sought is determined

The hand-eye calibration problem is important for applications that require precise alignment or coordination between the camera and the tool, such as robot manipulation, augmented reality, or medical surgery.

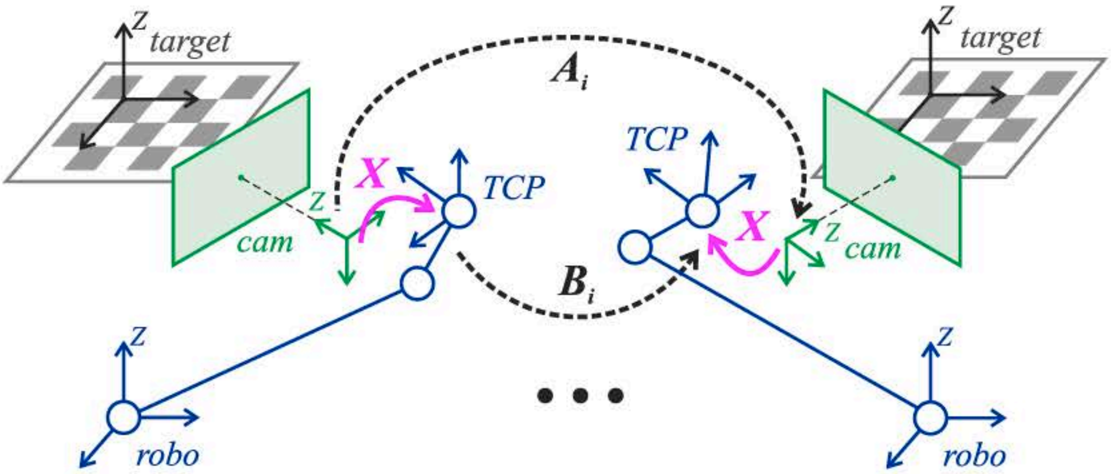


Fig. 9.1: Illustration of the transformations in the hand/eye problem.

Bibliography

A complete documentation is part of a licensed programme version.

We will be happy to make you an attractive offer on request by e-mail 3d-easycalib@zbs-ilmenau.de !