

3D-EasyCalib

Toolkit zur Kalibrierung von Kameras und Robotern

BEDIENUNGSANLEITUNG - DEMO

Programmversion 3.1.1

ZBS e. V.

Werner-von-Siemens-Straße 12
98693 Ilmenau



Impressum

©2024–2026 Zentrum für Bild- und Signalverarbeitung e. V.

Werner-von-Siemens-Straße 12, D-98693 Ilmenau

info@zbs-ilmenau.de

www.zbs-ilmenau.de

Schriftenreihe des ZBS

ISSN: 1432-3346

Die Urheber übernehmen für fehlerhafte Angaben und deren unmittelbare und mittelbare Folgen keine juristische Verantwortung oder Haftung.



Inhaltsverzeichnis

Inhaltsverzeichnis	3
1 Einleitung	5
1.1 Geschichte der Kamerakalibrierung	5
1.2 Was ist der Zweck der Kamerakalibrierung?	7
1.3 Was zeichnet eine gute Kamerakalibrierung aus?	7
2 Theorie des Lochkameramodells	9
2.1 Parameter des Lochkameramodells	11
2.2 Modell der Linsenverzeichnung	12
3 Kalibriertarget	15
3.1 Raster des Schachbrett-Kalibriernormalis	15
3.2 Hinweise für Aufnahmen von Kalibriertargets	19
4 Benutzeroberfläche des 3D-EasyCalib	23
4.1 Hauptfenster	23
4.2 Aufbau der Projektdatei	27
4.3 Darstellung der intrinsischen Parameter in 3D-EasyCalib	27
4.4 Darstellung der extrinsischen Parameter in 3D-EasyCalib	29
4.5 Bezugssystem und Formelnotation	30
4.6 Unterstützte Bildformate	32
5 Intrinsische Kamerakalibrierung	33
5.1 Grundlagen	33

5.2	Reprojektionsfehler	34
5.3	Eingabe	35
5.4	Visualisierung	36
5.5	Einstellungen und Aktionen	37
5.6	Ausgabe	39
5.7	Verwendung eigener Passpunkte	39
6	Extrinsische Stereo-Kalibrierung	43
6.1	Grundlagen	43
6.2	Eingabe	45
6.3	Visualisierung	46
6.4	Einstellungen und Aktionen	47
6.5	Ausgabe	48
6.6	Stereo-Rektifizierung	49
6.7	Verwendung eigener Passpunkte	52
7	Projektor-Kalibrierung	53
7.1	Grundlagen	53
7.2	Erzeugung des Datensatzes	56
7.3	Eingabe	57
7.4	Einstellungen und Aktionen	59
7.5	Ausgabe	60
8	Robot/World- und Tool/Flange-Kalibrierung	63
8.1	Grundlagen	63
8.2	Erzeugung des Datensatzes	66
8.3	Eingabe	67
8.4	Einstellungen und Aktionen	70
8.5	Ausgabe und Visualisierung	71
9	Roboterwerkzeug zu Kamera Kalibrierung (Hand/Eye)	73
9.1	Grundlagen	73
9.2	Eingabe	76
9.3	Einstellungen und Aktionen	78
9.4	Ausgabe und Visualisierung	79
	Literaturverzeichnis	81

Einleitung

Eine Kamera erzeugt planare Bilder der physikalischen Welt, die wir als dreidimensional empfinden. Die Geometrie der Entstehung eines Bildes wurde erst in der Renaissance genauer untersucht und verstanden. Die Maler der damaligen Zeit (siehe [Abb. 1.1](#)) leiteten die geometrische Konstruktion der perspektivischen Projektion ab und wandten sie bei ihren Gemälden an. Als im 19. Jahrhundert die Photographie erfunden wurde, wurde auch die Vermessung der Szene aus den perspektivischen Abbildungen interessant. Seit dieser Zeit hat sich die Photogrammetrie, die sich mit dieser Frage beschäftigt, ständig weiterentwickelt. Im gleichen Jahrhundert definierten Mathematiker die projektive Geometrie, die sich dem Umgang mit Punkten im Unendlichen zuwandte und perspektivische Projektionen behandelte.

1.1 Geschichte der Kamerakalibrierung

Die ersten Methoden und Verfahren für die praktische Kamerakalibrierung wurden durch die Luftbildphotogrammetrie während des Ersten Weltkriegs angeregt [7]. Die Motivation war, Entfernungen aus Luftbildern nach dem stereoskopischen Prinzip zu schätzen und die Erdoberfläche zu kartieren. Da sich das ideale mathematische Kameramodell insbesondere durch die Linsenverzeichnung stark von der realen Kamera unterscheidet, wurde zunächst viel Aufwand in die Entwicklung der mathematischen Modelle für die Linsenverzeichnung gesteckt. Aufgrund der damaligen Fertigungstoleranzen waren die von den Herstellern angegebenen Parameter für Brennweite und radiale Verzeichnung nicht genau genug. Außerdem musste die dezentrierende Verzeichnung korrigiert werden, die durch eine unsachgemäße Zentrierung der Linsenelemente entlang der optischen Achse verursacht wurde.

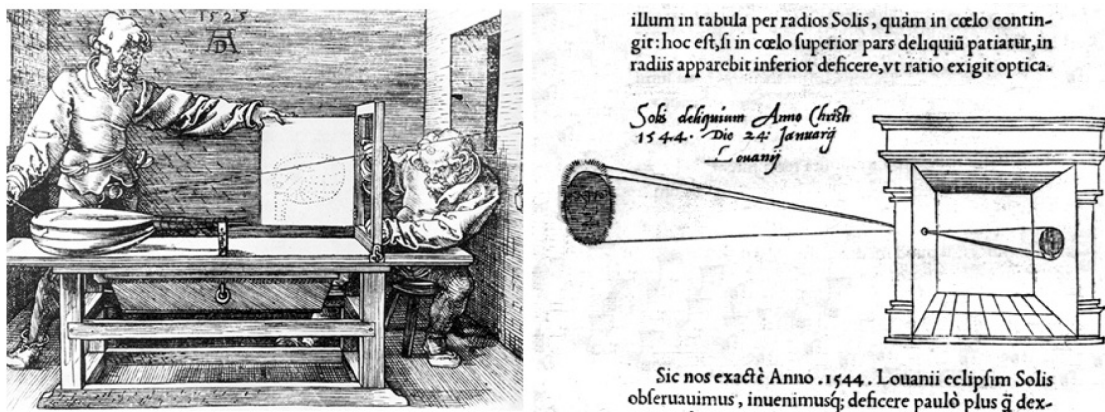


Abb. 1.1: Albrecht Dürer, 1525, Unterweysung der Messung (links). Die erste Illustration einer Lochkamera, der sog. Camera Obscura von Reinerus Gemma-Frisius, 1544 (rechts).

1956 fasste Brown [5] wissenschaftliche Fortschritte auf dem Gebiet der Modellierung von Kameraobjektiven zusammen und bewertete sie. Er kombinierte die bereits bekannte radiale Verzerrung und die von Conrady [9] neu eingeführte tangentielle Verzerrung und untersuchte das vorgeschlagene Modell experimentell. Dieses Linsenverzeichnungsmodell ist bis heute gültig und wird in [Abschnitt 2.2](#) erläutert. Brown war es auch, der in [4] den Grundstein für eine modernes Bündelausgleich zur Bestimmung der Kameraparameter legte.

Der Prozess der Bestimmung von Kameraparametern war ziemlich zeitaufwändig, bevor Computer erfunden wurden. Ursprünglich wurden die Sternbilder verwendet, um die Bildentstehungsparameter manuell zu berechnen. Dieses Verfahren wurde aufgrund des geringeren Aufwands durch den rotierenden Kollimator nach dem Goniometerprinzip ersetzt. Wieder war es Brown, der eine viel einfachere Methode vorstellte: die sogenannte Plumb-Line-Kalibrierung [6]. Brown beschrieb ein mathematisches Modell zur Bestimmung von Linsenverzerrungsparametern basierend auf den Bildern von geraden Linien. Diese Technik hat sich als sehr nützlich erwiesen und wurde verwendet, um die Kameras für verschiedene Anwendungen zu kalibrieren – von der Erfassung auf mikroskopischer Ebene bis hin zu Luftbildern.

Mit dem Einzug des digitalen Zeitalters wurde die Optimierung der Parameter aus der Kalibrierung am Computer deutlich beschleunigt. 1986 präsentierte Tsai ein robustes, zweistufiges Verfahren zur Bestimmung der intrinsischen und extrinsischen Parameter mithilfe eines 3D-Targets. Es wurde erst zehn Jahre später populär, als Willson [19] die erste frei verfügbare Implementierung bereitstellte. Zu diesem Zeitpunkt hat sich ZBS e. V. erstmals mit der Kamerakalibrierung befasst und seitdem

die Kompetenzen auf diesem Gebiet kontinuierlich ausgebaut. Die Wurzeln des 3D-EasyCalib-Toolkits gehen auf die Matlab-Implementierung von Bouguet [2] zurück. Er nahm die von Zhang [20] vorgestellte Methode, die mehrere planare Targets verwendet und die Parameter mit einer Gradientenabstiegsmethode nach Levenberg-Marquardt optimiert. Bouguet hat erkannt, dass insbesondere die Genauigkeit der ermittelten Kalibrierpunkte eine entscheidende Rolle für eine präzise Kalibrierung spielt. Die Visualisierung dieser Fehler und die Möglichkeit, sie manuell zu korrigieren, unterscheidet das Toolkit von anderen. Da es bisher keine zuverlässige Methode zur automatischen Entfernung falsch erkannter Passpunkte gibt, ist es wichtig, diese zu visualisieren, um selbst Korrekturen vornehmen zu können. Dessen Methoden wurden dann in C++ in die Bibliothek *OpenCV* [3] portiert. Die damals eingesetzten Methoden werden heute noch in nahezu unveränderter Form verwendet.

1.2 Was ist der Zweck der Kamerakalibrierung?

Der Einsatz von multimodalen Mehrkamerasystemen nimmt im Bereich der Robotik allgemein und der Mensch-Roboter-Kooperation (MRK) kontinuierlich zu. In diesen Anwendungen müssen vielfach 2D- und 3D-Sensordaten geometrisch registriert und fusioniert sowie darin Elemente der Aufnahmeanordnung und Szenenobjekte korrekt im Raum verortet werden. Dabei spielen Methoden zur geometrischen Kalibrierung von datenerfassenden Systemen, wie Kameras, und von im Raum agierenden Objekten, wie Robotern, eine herausragende Rolle. Gegenstand der geometrischen Kalibrierung ist die Bestimmung der (intrinsischen) Abbildungseigenschaften von Kameras sowie der geometrischen Lage und Orientierung von Kameras bzw. Roboter in Weltkoordinaten durch extrinsische Parameter.

1.3 Was zeichnet eine gute Kamerakalibrierung aus?

Das Kalibriertarget sollte passend zum Arbeitsbereich der Kamera und zur Bildauflösung ausgelegt werden. Außerdem sollte es präzise gefertigt sein.

Bildaufnahme. Eine ausreichende Anzahl aufgenommener Bilder des Targets mit unterschiedlichen Abständen und Ausrichtungen zur Kamera ist erforderlich. Die Kamera sollte sich während der Aufnahme möglichst wenig bewegen und die Objekteneinstellungen sollten unverändert bleiben (insbesondere der Autofokus sollte ausgeschaltet sein).

Die korrekte Wahl der zu optimierenden Kameraparameter hängt von der verwendeten Kamera und Optik ab. Obwohl viele freie Parameter zu kleineren Kalibrie-

rungsfehlern führen können, garantiert das nicht die Genauigkeit des berechneten Modells.

Ausreißerkennung und -beseitigung Im Idealfall lassen sich alle Kalibrierpunkte in die Optimierung einbeziehen. Da jedoch die Koordinaten der Passpunkte in realen Aufnahmen, z.B. durch Okklusion, Schatten, Kamerarauschen oder Bewegung während der Aufnahme unpräzise oder falsch detektiert werden können, ist es notwendig, Ausreißer zu identifizieren und gegebenenfalls zu entfernen.

Theorie des Lochkameramodells

Das Lochkameramodell definiert die Transformation des dreidimensionalen euklidischen Raums auf eine zweidimensionale Bildebene nach dem Prinzip der Zentralprojektion und wird zur Approximation realer Kameras benutzt. In einer Lochkamera erfolgt die Abbildung eines Raumpunktes in einen Bildpunkt kollinear durch einen Strahl, der durch das Projektionszentrum der Kamera geht.

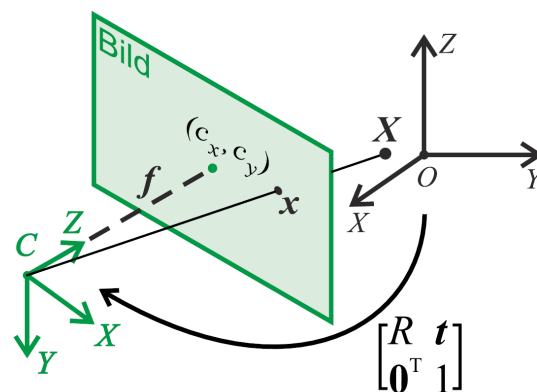


Abb. 2.1: Komponenten des Lochkameramodells, bestehend aus der Bildebene, dem Projektionszentrum C und der Bildweite f , dem Abstand zwischen C und der Bildebene. Die optische Achse ist die Linie, die durch C verläuft und senkrecht zur Bildebene steht. Sie schneidet die Bildebene im Hauptpunkt (c_x, c_y) . Das Bild $\mathbf{x}$ eines Punktes im Raum $\mathbf{X}$ ist der Schnittpunkt des optischen Strahls $(C, \mathbf{X})$ mit der Bildebene.

Das Modell lässt sich anhand der [Abb. 2.1](#) wie folgt herleiten. Wenn zunächst das 3D-Koordinatensystem der Welt gleich dem der Kamera mit dem Ursprung im Punkt C gesetzt wird, dann ist bei der Verwendung einer Notation gemäß [12] diese Projektion kompakt mit der Gleichung

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_C \quad (2.1)$$

beschrieben. Mit $\mathbf{x} = u \cdot (x, y, 1)^T$ und $\mathbf{X}_C = (X, Y, Z, 1)^T$ werden die homogenen Koordinatenvektoren des Bildpunktes bzw. des Raumpunktes bezeichnet. Diese sind bis auf einen Skalierungsfaktor eindeutig bestimmt. Auch wenn der Skalierungsfaktor für $\mathbf{X}_C$ gleich 1 gewählt wird, ist für $\mathbf{x}$ ein Skalierungsfaktor u explizit zu berücksichtigen. Die echten Bildkoordinaten x, y erhält man aus der Division von $\mathbf{x}$ durch dessen dritte Komponente.

Der Index C verdeutlicht, dass der Raumpunkt $\mathbf{X}_C$ im Kamerakoordinatensystem definiert ist. Die Schreibweise $[I|\mathbf{0}]$ steht für eine 3×4 -Blockmatrix, die aus der 3×3 -Einheitsmatrix I und dem rechtsstehenden Spaltenvektor $\mathbf{0} = (0, 0, 0)^T$ besteht. Die 3×3 -Matrix K

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

wird hier als Kameramatrix oder auch Kalibrierungsmatrix des Lochkameramodells bezeichnet. Sie besteht aus den intrinsischen Parametern Bildweite f (Abstand der Bildebene vom Kamerazentrum) und Bildhauptpunkt (c_x, c_y) .

Jetzt sei der Raumpunkt $\mathbf{X}$ in einer 3D-Welt definiert, und die Orientierung sowie die Lage der Kamera in dieser Welt seien allgemein mit der Rotationsmatrix R und der Translation $\mathbf{t}$ beschrieben. Dann ist $\mathbf{X}$ zunächst affin in das Kamera-Koordinatensystem gemäß

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

zu transformieren und anschließend nach (2.1) auf die Bildebene zu projizieren. Insgesamt erhält man den bekannten algebraischen Zusammenhang des Lochkameramodells

$$\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}. \quad (2.4)$$

Die ausgeschriebene Formel lautet:

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

mit dem o.g. Skalierungsfaktor u .

Bemerkung. Die extrinsischen Parameter R und $\mathbf{t}$ in der Gleichung (2.3) stellen die Transformation vom Weltkoordinatensystem mit dem Ursprung O in das Kamerakoordinatensystem mit dem Ursprung C dar. Der Translationsvektor $\mathbf{t}$ zeigt vom C zu O , $\mathbf{t} = \overrightarrow{CO}$. Die inverse Transformation, von Kamera- in die Weltkoordinaten, lautet:

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

2.1 Parameter des Lochkameramodells

Bildhauptpunkt

Der Bildhauptpunkt (c_x, c_y) ist der Schnittpunkt der optischen Achse (Z-Achse des Kamera-Koordinatensystems mit der Bildebene, siehe Abb. 2.1. Er ist im rechtshändigen kartesischen Bild-Koordinatensystem mit dem Ursprung in der linken oberen Ecke des Bildes definiert.

Bemerkung 1. Der Bildhauptpunkt befindet sich im Normallfall in der Nähe der Bildmitte. Das gilt nicht, wenn zum Beispiel das Bild asymmetrisch abgeschnitten wird. Auch bei einer Linsenverstellung (engl. lens shift), die insbesondere bei DMD-Projektoren sehr häufig auftritt, wird der Hauptpunkt von der Mitte verschoben.

Bemerkung 2. Der Bildhauptpunkt ist in Bildkoordinaten angegeben. Die X-Achse des Bildkoordinatensystems ist parallel zu den Bildzeilen und zeigt von links nach rechts und die Y-Achse ist parallel zu den Bildspalten und zeigt von oben nach unten. Der Ursprung des Bildkoordinatensystems ist die linke obere Ecke des Bildes.

Bildweite

Die Bildweite f bezeichnet den Abstand der Bildebene vom Kamerazentrum. Aus praktischen Gründen wird sie in zwei sogenannte effektive Bildweiten f_x und f_y aufgeteilt. Sie entsprechen der Bildweite f multipliziert mit der horizontalen beziehungsweise vertikalen Skalierung. Dieser Wert entspricht ungefähr der realen Brennweite des Kameraobjektivs, gemessen als Vielfaches des Pixelabstandes. Die Einheit ist ein Pixel (1 px). Zum Beispiel beträgt für eine Kamera mit dem Chip Sony IMX174 mit der Pixelgröße von $5,86 \mu\text{m}$ und einem Objektiv mit 8 mm Brennweite die effektive Bildweite

$$f_x = f_y = \frac{8 \text{ mm}}{0,00586 \frac{\text{mm}}{\text{px}}} = 1365,2 \text{ px.}$$

Bemerkung. Für Digitalkameras (CCD, CMOS) mit quadratischen Pixeln sollten die effektiven Bildweiten f_x, f_y bei der Parameteroptimierung stets gleichgesetzt werden. Nur in folgenden plausiblen Fällen sind f_x und f_y ausnahmsweise unterschiedlich anzusetzen:

- bei unterschiedlicher Skalierung des Bildes in horizontaler und vertikalen Richtung, verursacht entweder durch ein anamorphotisches Kameraobjektiv oder durch eine nachträgliche Bildverarbeitung, in der das Bild nicht uniform gestaucht bzw. gestreckt wird;
- bei nicht-quadratischen Pixeln des Sensors. Dies tritt insbesondere für Filme bzw. Filmplatten oder Aufnahmegeräte mit einem analogen Framegrabber zu.

2.2 Modell der Linsenverzeichnung

Bisher wurde angenommen, dass das lineare Lochkameramodell den Bildaufnahme-prozesses exakt beschreibt. In diesem Modell sind der Raumpunkt, der korrespondierende Bildpunkt und das optische Zentrum kollinear, und die Objekte der Szene werden mittels Zentralprojektion auf die Bildebene projiziert. Dieses Modell gilt unter der Annahme, dass die Bildebene (Sensor, Film) eben ist, die Pixel (Sensorelemente) in einem regelmäßigen Gitter angeordnet sind und das abbildende System keine Verzeichnung aufweist. Die ersten beiden Annahmen sind für digitale Kameras erfüllt, die letzte jedoch nicht. Insbesondere bei Kameras mit Weitwinkelobjektiv ist die Verzeichnung zu beachten.

In der digitalen Bildverarbeitung hat sich ein kombinierter Ansatz aus radialer und tangentialer Verzeichnung bewährt [5]. Die radiale Verzeichnung berücksichtigt die geometrisch-optischen Eigenschaften eines ideal zentrierten optischen Systems. Die tangentiale Verzeichnung modelliert Symmetriefehler durch den Effekt eines dünnen Prismas. Unter Berücksichtigung späterer Arbeiten [8] ergibt sich das folgende sehr allgemeine Verzeichnungsmodell:

$$\begin{aligned} x_u &= x \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2x^2) + 2p_2 xy \\ y_u &= y \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_2(r^2 + 2y^2) + 2p_1 xy \end{aligned} \quad (2.7)$$

mit den unverzerrten (korrigierten) Bildkoordinaten x_u, y_u und den realen Bildkoordinaten x, y bzw. $r = \sqrt{x^2 + y^2}$. Alle Koordinaten werden vom Bildhauptpunkt aus gemessen. Die radialen Koeffizienten $k_1, \dots, k_6$ sowie die tangentialen Koeffizienten p_1, p_2 gehören zu den inneren Kameraparametern.

Diese Korrekturvorschrift wird mit den normierten Kamerakoordinaten $x := X_C/Z_C$, $y := Y_C/Z_C$ durchgeführt, d.h. sie wird in Gleichung (2.4) zwischen K und $[R|t]$ „eingeschoben“. Damit gelten die Verzeichnungskoeffizienten unabhängig von der Bildauflösung.

Bemerkung. Die tangentielle Verzerrung (p_1, p_2) ist bei heutigen digitalen Kameras vernachlässigbar. Diese Parameter haben vor knapp hundert Jahren aufgrund von höheren Herstellungstoleranzen eine bedeutende Rolle gespielt. So war es nicht immer gegeben, dass der Film beziehungsweise die Filmplatte senkrecht zur optischen Achse des Objektivs stand. Daher wurde dieser Bildfehler nach dem Modell des dünnen Prismas korrigiert. Die Parameter für die tangentielle Verzeichnung (p_1, p_2) sollten daher nur in plausiblen Fällen bei der Kalibrierung berücksichtigt werden.

Kalibriertarget

Die Voraussetzung für gute Kalibrierungsergebnisse ist ein präzise gefertigtes Kalibriernormal, auch Kalibriertarget genannt. Ein klassisches Target stellt mehrere Raum- (Pass-)Punkte mit bekannter geometrischer Lage im Raum (3D-Target) oder auf einer Ebene (2D- bzw. planares Target) bereit. Die Gestaltung der Passpunkte und ihres Umfeldes auf dem Target werden so gewählt, dass sie anhand ihrer Geometrie und ihres Umgebungskontrastes im realen Kamerabild robust, eindeutig und automatisch zu detektieren und zu vermessen sind. Aus praktischer Sicht sind Normale, die einfach und in unterschiedlichen Größen herstellbar sind und universell für viele Kalibrieraufgaben verwendet werden können, zu bevorzugen. Ein planares Kalibriertarget mit einem Schachbrettmuster erfüllt diese Anforderungen. Die Passpunkte sind an den Ecken der Quadrate lokalisiert.

Die in 3D-EasyCalib enthaltenen Kalibriermethoden verwenden Aufnahmen eines solchen planaren Kalibriertargets. Die Einstellungen für die Erkennung des Targets und die präzise Vermessung der Kalibrierpunkte werden nachfolgend beschrieben:

3.1 Raster des Schachbrett-Kalibriernormals

Einfaches Schachbrett

Im einfachsten Fall besteht das Kalibriermuster nur aus abwechselnd schwarzen und weißen quadratischen Feldern ohne zusätzliche Merkmale. Nur die inneren Ecken werden detektiert und zur Kalibrierung verwendet. Deshalb können die äußeren Felder verkleinert oder abgerundet werden. Zur eindeutigen Identifikation der Eckpunkte muss die Feldanzahl horizontal und vertikal von unterschiedlicher

Parität (gerade / ungerade) sein. Folgende Parameter, die im Hauptfenster einstellbar sind, definieren das einfache Schachbrett:

Num Squares X: Anzahl der Felder in horizontaler Richtung. Die äußeren (eventuell verkleinerten) Felder werden mitgezählt.

Num Squares Y: Anzahl der Felder in vertikaler Richtung. Die äußeren (eventuell verkleinerten) Felder werden mitgezählt.

Square Width [mm]: Breite eines inneren Feldes in mm.

Square Height [mm]: Höhe eines inneren Feldes in mm, wenn von der quadratischen Form abgewichen wird.

Für die Berechnung der intrinsischen Kameraparameter ist nur das Verhältnis der Feldkantenlängen (bei nicht quadratischen Feldern) von Bedeutung, die absolute Größe eines Feldes ist irrelevant. Andererseits ist für die Berechnung der extrinsischen Parameter eine präzise Angabe für die Feldgröße unbedingt erforderlich.

Schachbrett mit Positionsmarkierungen

Das einfache Schachbrett hat den Nachteil, dass es von der Kamera vollständig abgebildet werden muss. Nur so können die gefundenen Ecken eindeutig den Zeilen und Spalten des Musters zugeordnet werden. Wenn diese Bedingung wegen der Größe des Schachbretts nicht erfüllbar ist, kann man durch Markierungen eine bestimmte Ecke als Koordinatenursprung kennzeichnen. Konkret werden drei Kreismarkierungen in benachbarte Felder gesetzt, so dass sie die Richtungsvektoren eines ebenen Koordinatensystems aufspannen. Damit wird eindeutig eine Ecke des Musters als Koordinatenursprung festgelegt. [Abb. 3.1](#) zeigt Beispiele für diese Art Target.

Spalten- und Zeilenzahl des Musters sind frei wählbar, bezüglich Parität besteht keine Einschränkung. Die Größe der Kreismarkierungen ist variabel. Ein Tool der Bibliothek *OpenCV* legt den Radius der Kreise auf 17% der Kantenlänge eines Feldes fest. Zur Kalibrierung muss nur ein Teil des Schachbretts im Bild sichtbar sein, aber mindestens die Felder mit den Kreismarkierungen müssen abgebildet werden. Folgende Parameter sind zusätzlich zu den Parametern des einfachen Schachbretts im Hauptfenster einzugeben:

Use Position Markers: Das Kontrollkästchen ist zu aktivieren, damit die weiteren Parameter verfügbar sind.

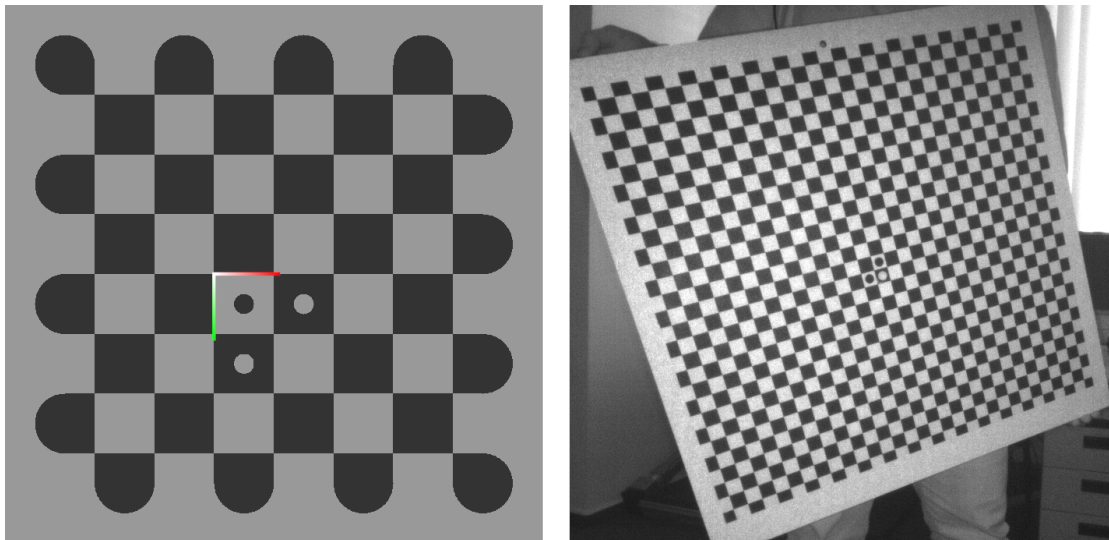


Abb. 3.1: Targets mit Positionsmarkierungen. Links simuliertes Target der Größe 8x8 mit Markierung bei X=3, Y=4. Der farbige Winkel symbolisiert das Koordinatensystem und gehört nicht zum Muster. Die Rundungen der äußeren Felder haben keine tiefere Bedeutung für die Kalibrierung. Rechts reales Target während eines Kalibrierprozesses.

Marker Base X: Spaltenindex der Ecke, die den Koordinatenursprung bilden soll, von links mit Index 1 beginnend.

Marker Base Y: Zeilenindex der Ecke, die den Koordinatenursprung bilden soll, von oben mit Index 1 beginnend.

ChArUco-Target

Dieser Target-Typ besteht aus einem Schachbrettmuster, bei dem jedes helle Feld einen speziellen Binärcode enthält, einen sogenannten ArUco-Marker. Die Abkürzung steht für „Augmented Reality University of Cordoba“ und bezeichnet eine Methode zur Erzeugung und Erkennung von speziellen Folgen quadratischer Binärmuster [11]. Ein Marker aus $n \times n$ Bits kann theoretisch 2^{n^2} verschiedene Muster bilden, die aber nicht alle in einer solchen Folge enthalten sind. Aus Gründen der Fehlertoleranz und schnellen Detektierbarkeit wird nur eine Teilfolge der möglichen Muster verwendet. Jede zulässige Teilfolge wird als „Dictionary“ bezeichnet.

Schachbrettmuster mit ArUco-Markern („ChArUco“) können mit der Bibliothek *OpenCV* ab Version 4.7.0 erzeugt werden. Sie werden durch die Klasse `cv::aruco::CharucoBoard` repräsentiert. Jeder ArUco-Marker ist von einem hellen Rand umgeben

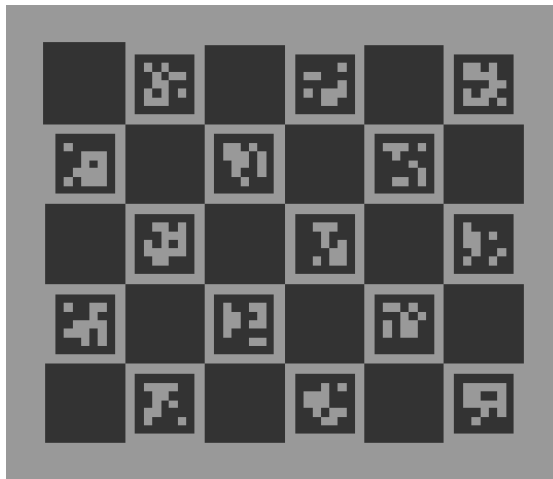


Abb. 3.2: Simuliertes ChArUco-Target aus 6 x 5 Feldern und Markern der Größe 5 x 5 Bits, basierend auf dem „Dictionary“ DICT_5X5_50

und dadurch sicher lokalisierbar. [Abb. 3.2](#) zeigt ein Beispiel. Folgende Parameter, die im Hauptfenster einstellbar sind, definieren das ChArUco-Target:

Dictionary: Hier ist das „Dictionary“ auszuwählen, mit dem das ChArUco-Target erzeugt wurde. Die Bezeichnungen stimmen mit den Aufzählungswerten in `cv::aruco::PredefinedDictionaryType` der *OpenCV*-Bibliothek überein.

Num Squares X: Anzahl der Felder in horizontaler Richtung.

Num Squares Y: Anzahl der Felder in vertikaler Richtung.

Square Size [mm]: Breite und Höhe eines Feldes in mm. Es sind nur quadratische Felder zugelassen.

ArUco Size [mm]: Breite und Höhe eines Markers in mm. Muss kleiner als „Square Size“ sein, um einen hellen Rand zu erzeugen.

ChArUco Legacy: Definiert eine Einschränkung zur Farbe des linken, oberen Feldes, die nur für *OpenCV*-Versionen älter als 4.6.0 notwendig ist.

3.2 Hinweise für Aufnahmen von Kalibriertargets

Die Voraussetzung für eine präzise Kalibrierung ist ein präzises, biegefestes Kalibriertarget. Die optimale Größe des Targets sowie die Anzahl der inneren Ecken hängen von der Kamera und dem Anwendungsszenario (Größe des Arbeitsraumes) ab. Auf Anfrage entwerfen und fertigen wir für Ihren Anwendungsfall das ideale Kalibrier-normal. Bei den Kameraaufnahmen eines Kalibriertargets sollte auf folgende Punkte geachtet werden:

- Ein Schachbrettmuster ohne zusätzliche Codierung muss vollständig abgebildet sein.
- In jeder Aufnahme sollte das Target bezüglich der Kamera unterschiedlich orientiert und positioniert sein.
- Die Orientierung des Targets sollte um alle drei Raum-Achsen variiert werden.
- Das Target sollte in unterschiedlichen Abständen zur Kamera erfasst werden, so dass möglichst der ganze Arbeitsraum abgedeckt wird.
- Die Kalibrierpunkte der gesamten Bildserie sollten über das ganze Bild verteilt sein.
- Fokus und Brennweite des Kameraobjektivs dürfen sich während der Aufnahmen nicht ändern.
- Die Anzahl der Bilder für eine Kalibrierung sollte mindestens 10 betragen.

Bemerkung zur Wahl der geometrischen Form für das Kalibriertarget

Die meisten Targets verwenden entweder Kreise oder Schachbrettmuster. Bei einem verzerrungsfreien Kameraobjektiv ist es irrelevant, welche Art von Merkmalen auf einem Kalibriertarget verwendet wird. Sowohl die Mittelpunkte der Kreise als auch die Ecken des Schachmusters können in einem perspektivisch verzerrten Bild mit gleicher Genauigkeit bestimmt werden. Wenn das Bild jedoch aufgrund der Objektiveigenschaften stark verzeichnet wird, werden Kreise asymmetrisch verzerrt und die Koordinaten des ursprünglichen Zentrums sind nicht trivial zu bestimmen. Das wurde auch in [15] theoretisch und experimentell nachgewiesen. Daher wird für die Kalibrierung ein Schachbrettmuster anstelle eines Targets mit Kreisen empfohlen.

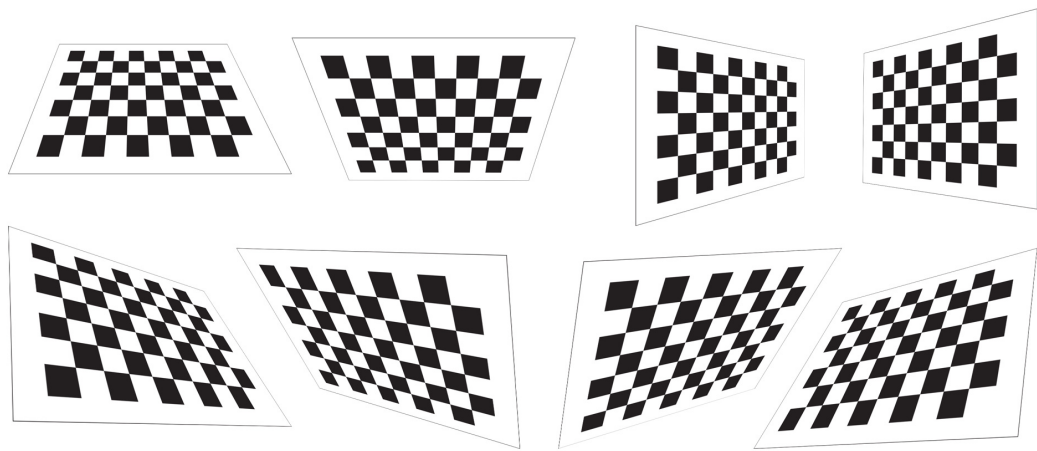


Abb. 3.3: Empfohlene Targetposen für die Kalibrierung. Zuerst sollte das Target möglichst nah vor der Kamera aufgenommen werden. Ähnliche Targetposen sind dann noch bei größeren Kamera-Target-Abständen aufzunehmen.

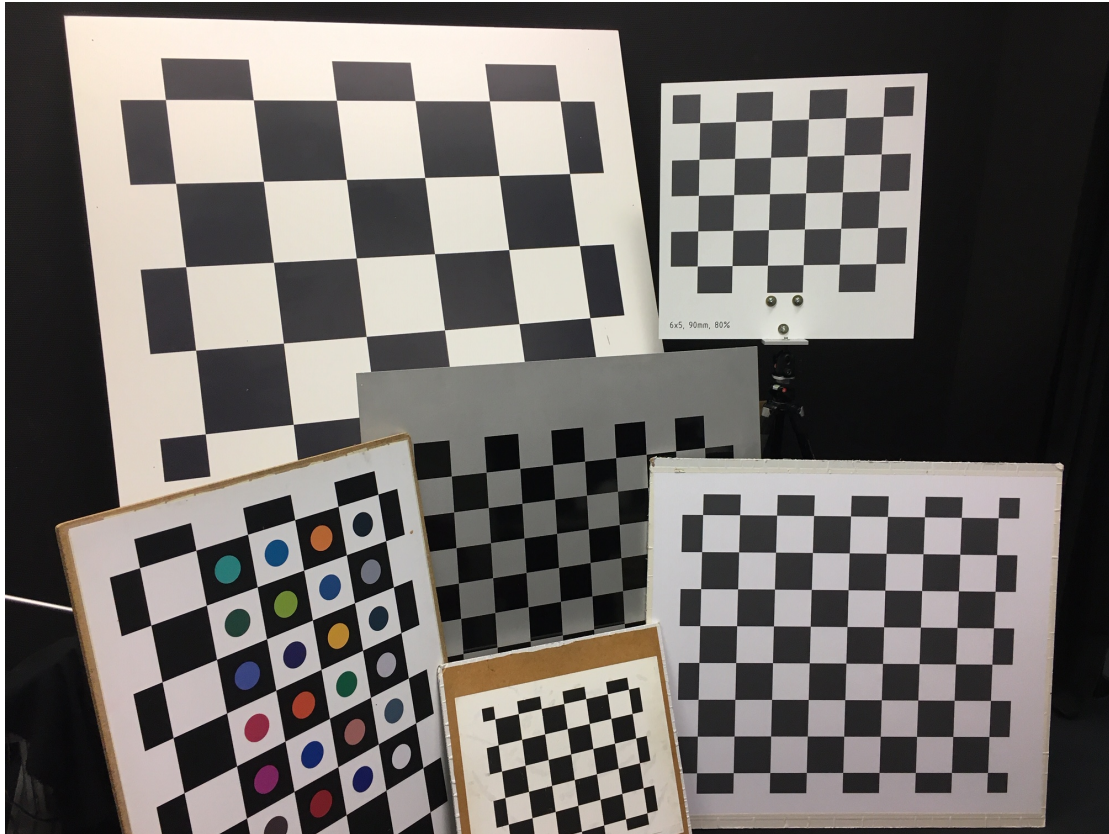


Abb. 3.4: Kalibriertargets für unterschiedliche Einsatzzwecke

Benutzeroberfläche des 3D-EasyCalib

Hinweis

Die nachfolgend gezeigten Screenshots der Benutzeroberfläche können in Abhängigkeit von der Programmversion und den Einstellungen des Betriebssystems geringfügig von der realen Darstellung abweichen.

4.1 Hauptfenster

Das Hauptfenster des Programms ist in drei Bereiche aufgeteilt: auf der linken Seite sind die hinzugefügten Komponenten (Kameras, Koordinatensysteme) als Liste dargestellt (3). Der mittlere Teil enthält Ausgaben in visueller und Textform (4, 5). Rechts werden das Kalibriertarget und die Aktionen definiert (6). Die Details der Benutzeroberfläche werden nachfolgend entsprechend ihrer Nummerierung in [Abb. 4.1](#) beschrieben.

① Das File-Menü beinhaltet die Aktionen:

New Project löscht alle hinzugefügten Komponenten und erstellt ein neues Projekt.

Open Project lädt ein gespeichertes Projekt aus einer YML-Datei. Die Spezifikation des Formats ist im [Abschnitt 4.2](#) beschrieben.

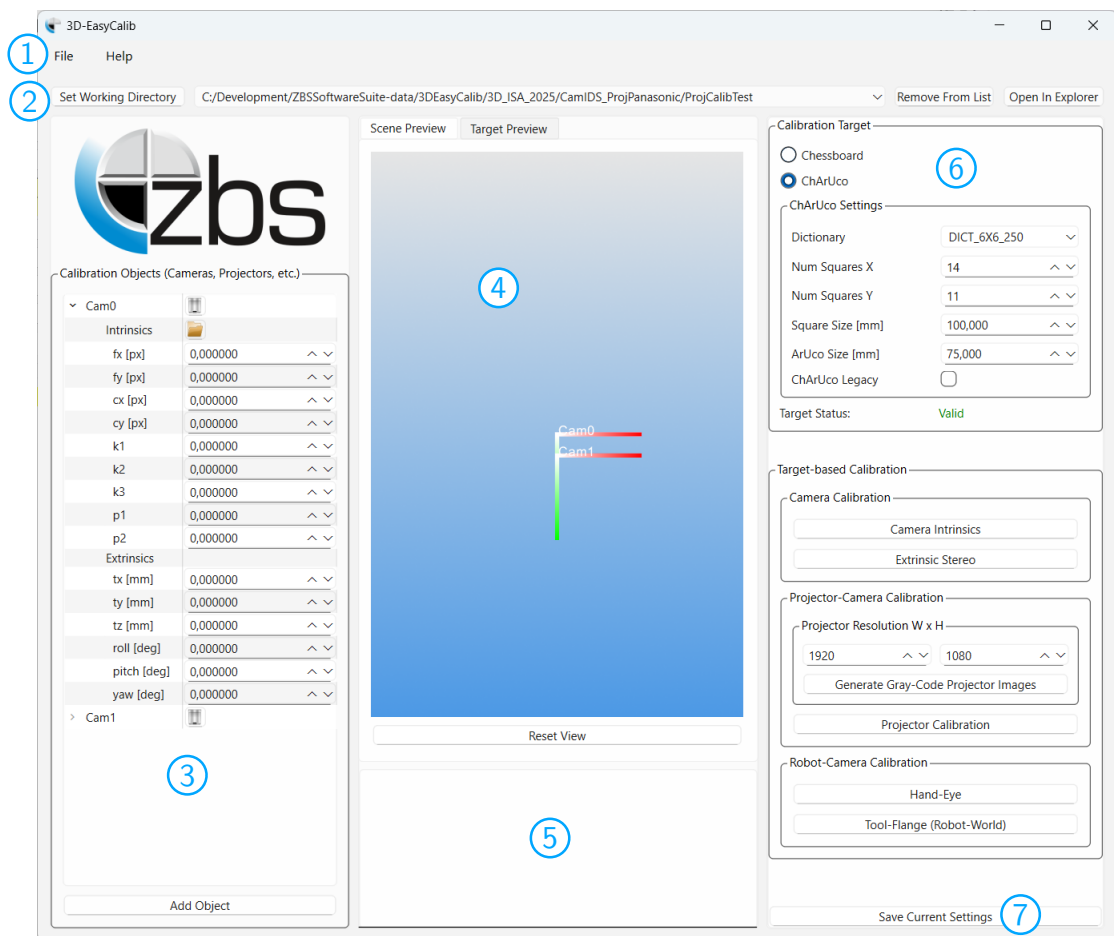


Abb. 4.1: Hauptkomponenten der Benutzeroberfläche

Save Project exportiert das Projekt als YML- und Json-Datei. Hier wird der Inhalt des linken Bereichs des Hauptfensters gespeichert, bestehend aus den intrinsischen und extrinsischen Parametern aller Kameras im aktuellen Kalibrierzustand. Vor der Speicherung wird abgefragt, welche Kamera als Referenzobjekt für die extrinsischen Parameter dient. Alle Parameter werden automatisch auf dieses Referenzobjekt umgerechnet.

Open Settings öffnet die Konfigurationsdatei „3D-EasyCalib.ini“ zum Bearbeiten. Änderungen in der Datei werden erst beim nächsten Programmstart wirksam. Fehlende Einträge werden mit ihren Standardeinstellungen ergänzt. Der Eintrag

[General]


```
...  
UseDarkTheme=true
```

bewirkt einen dunklen Hintergrund in allen Fenstern. Wird `true` durch `false` ersetzt, entsteht ein heller Hintergrund.

Warnung

Andere Einstellungen sollten nur nach Rücksprache mit dem Programm-entwickler geändert werden.

- ② Mit der Schaltfläche [SET WORKING DIRECTORY] wird das Arbeitsverzeichnis festgelegt. Dieses Verzeichnis wird von den Datei-Dialogfenstern zuerst angeboten, und automatisch generierte Dateien werden in diesem Verzeichnis gespeichert.
- ③ In der *Objektliste* sind die an der Kalibrierung beteiligten Objekte (Kameras, Projektoren u. a.) mit ihren Parametern aufgelistet. Damit wird der aktuelle Kalibrierzustand vollständig beschrieben. Zu Beginn wird die benötigte Anzahl von Objekten mittels [ADD OBJECT] hinzugefügt. Bereits hinzugefügte Objekte lassen sich mit dem Papierkorb-Button löschen. Die intrinsischen Daten für eine Kamera können mit dem Ordner-Button geladen werden. Die Darstellung der intrinsischen bzw. extrinsischen Parameter ist in den Abschnitten 4.2ff. detailliert beschrieben. Nach erfolgreicher Kalibrierung werden die berechneten Parameter automatisch in die Objektliste übertragen.
- ④ Die Ansicht in der Mitte des Hauptfensters lässt sich mit den Tab-Reitern umschalten. Bedienung: Linke Maustaste dreht die Ansicht, mittlere Maustaste verschiebt die Szene parallel zur Kamera, das Mausrad vergrößert oder verkleinert die Ansicht. Die Schaltfläche [RESET VIEW] unterhalb der Ansicht stellt die Standard-Perspektive wieder her.

Scene Preview. Die beteiligten Objekte werden symbolisch mit ihren Koordinatenachsen und einer Beschriftung dreidimensional dargestellt.

Target Preview. Das rechts im Bereich (6) definierte Kalibriertarget wird maßstäblich exakt gezeichnet, einschließlich der optionalen Positionsmarkierungen oder der ChArUco-Codes.

- ⑤ In diesem Log-Fenster werden allgemeine Verarbeitungsschritte dokumentiert. Detaillierte Ausgaben findet man in den verfahrensspezifischen Fenstern ([Kapitel 5](#) und folgende).

- ⑥ **Calibration Target.** Im oberen Teil des rechten Hauptfensterbereichs werden die Parameter des Kalibriertargets vorgegeben. Es kann zwischen dem einfachen Schachbrett („Chessboard“), dem Schachbrett mit Positionsmarkierungen und dem ChArUco-Target gewählt werden. Die Parameter sind in [Abschnitt 3.1](#) erläutert.

Diese Einstellungen sind nur für den eingebauten Eckendetektor relevant. Sie werden ignoriert, wenn der Nutzer die Punktkorrespondenzen zwischen Target und Kamera oder Projektor vorgibt. Das genaue Vorgehen wird in [Abschnitt 5.7](#) beschrieben.

Target-based Calibration. Mit den Schaltflächen im unteren Teil wird die gewünschte Kalibriermethode aktiviert. Es öffnet sich jeweils ein für die Methode spezifisches Fenster. Dessen Verwendung wird in den nachfolgenden Kapiteln für jedes Verfahren erläutert.

[CAMERA INTRINSICS], intrinsische Kalibrierung einer einzelnen Kamera, [Kapitel 5](#).

[EXTRINSIC STEREO], extrinsische Stereo-Kalibrierung, [Kapitel 6](#).

[PROJECTOR CALIBRATION], Kalibrierung eines Video-Projektors bezüglich einer Kamera, [Kapitel 7](#). Mit der Schaltfläche [GENERATE GRAY-CODE...] wird eine Folge von PNG-Bildern mit Binärmustern im Arbeitsverzeichnis erzeugt. Die Bilder werden für die eingegebenen Werte der Projektor-Auflösung (Breite x Höhe in Pixeln) optimiert. Die Projektionen dieser Bilder auf ein Schachbrettmuster bilden die Grundlage der Projektor-Kalibrierung.

[HAND-EYE], Bestimmung von Lage und Orientierung einer Kamera an einem Roboterwerkzeug zum Roboterwerkzeug selbst, [Kapitel 9](#).

[TOOL-FLANGE (ROBOT-WORLD)], Registrierung einer stationären Kamera zu einem Roboter, [Kapitel 8](#).

- ⑦ Mit der Schaltfläche [SAVE CURRENT SETTINGS] wird eine Datei „MainWindowSettings.ini“ in das aktuelle Working Directory geschrieben. Diese Datei enthält die Einstellungen in den Gruppen „Calibration Target“ und „Projector-Camera Calibration“. Wird ein Working Directory mit einer vorhandenen „MainWindowSettings.ini“ gewählt, werden die entsprechenden Einstellungen automatisch geladen.

Hinweis

Das Hauptfenster wird NICHT gesperrt, wenn ein spezifisches Fenster für eine Kalibriermethode geöffnet ist. Änderungen im Hauptfenster (z. B. in der Objektliste) sind nachträglich möglich und werden in ein geöffnetes Kalibrierfenster übernommen.

4.2 Aufbau der Projektdattei

Eine Projektdattei im YAML-Format (siehe [Abb. 4.2](#)) enthält die intrinsischen und extrinsischen Parameter eines Kalibrierprozesses. Nachfolgend wird die Syntax beschrieben, obwohl eine manuelle Bearbeitung der Projektdatteen selten erforderlich ist.

Nach dem Header (%YAML:1.0) folgen das Datum (ISODate), die Objektnamen (Sensors) und das Bezugsobjekt (World), zu dem alle anderen Objekte extrinsisch kalibriert sind. Mit anderen Worten, alle Koordinatensysteme haben die gleiche Referenz – das Weltkoordinatensystem. Darunter stehen die intrinsischen (K_i, kc_i) sowie die extrinsischen (R_i, T_i) Parameter für alle Sensoren. Das i gibt die Position des Sensors in der Sensorliste an. Die Elemente der Kameramatrix K_i sind zeilenweise wie in (2.2) gespeichert. Die Reihenfolge der Verzeichnungskoeffizienten lautet

$$kc_i = [k_1, k_2, p_1, p_2, k_3]_i$$

oder

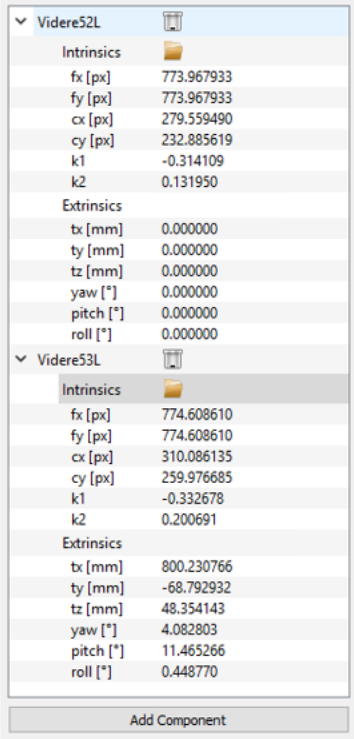
$$kc_i = [k_1, k_2, p_1, p_2, k_3, k_4, k_5, k_6, 0, 0, 0, 0, 0, 0]_i.$$

Wenn $k_4 = k_5 = k_6 = 0$ gilt, siehe (2.7), ist die erste Darstellung ausreichend. Ansonsten ist die zweite Darstellung zu verwenden. Die 6 Nullen am Enden sind erforderlich, haben aber zur Zeit keine Bedeutung.

R_i enthält die Rotationsmatrix vom Format 3x3 und T_i den Translationsvektor.

4.3 Darstellung der intrinsischen Parameter in 3D-EasyCalib

Die intrinsischen Parameter sind in der Objektliste (siehe [Abb. 4.3](#) links) dargestellt. Angezeigt werden die Elemente der Kameramatrix – die Bildweiten f_x, f_y und der Bildhauptpunkt (c_x, c_y) – sowie die Verzeichnungskoeffizienten. Diese Parameter werden entweder durch eine in 3D-EasyCalib enthaltene Methode erzeugt oder aus einer YAML-Datei mit folgendem Inhalt geladen:



The screenshot shows the 'Add Component' dialog box in the 3D-EasyCalib software. It lists two cameras: Videre52L and Videre53L. For each camera, intrinsic parameters (fx, fy, cx, cy, k1, k2) and extrinsic parameters (tx, ty, tz, yaw, pitch, roll) are displayed. The 'Add Component' button is at the bottom.

Parameter	Videre52L	Videre53L
fx [px]	773.967933	774.608610
fy [px]	773.967933	774.608610
cx [px]	279.559490	310.086135
cy [px]	232.885619	259.976685
k1	-0.314109	-0.332678
k2	0.131950	0.200691
tx [mm]	0.000000	800.230766
ty [mm]	0.000000	-68.792932
tz [mm]	0.000000	48.354143
yaw [°]	0.000000	4.082803
pitch [°]	0.000000	11.465266
roll [°]	0.000000	0.448770

```
%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. ]
```

Abb. 4.2: Beispiel einer Projektkonfiguration für zwei Kameras. Links ist die Darstellung im Programm und rechts der Inhalt der Projektdatei im YAML-Format.

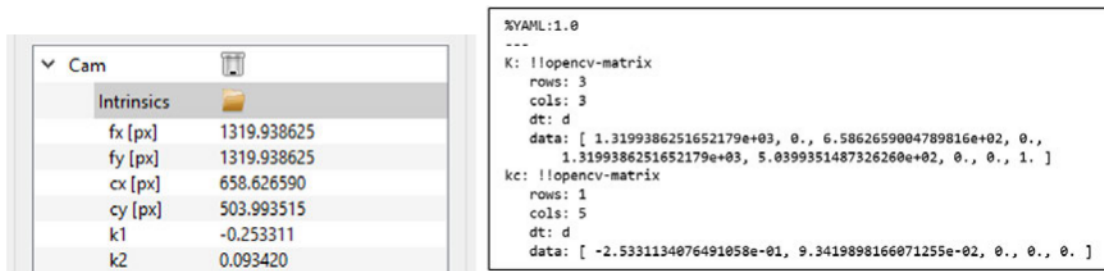


Abb. 4.3: Links: Darstellung der intrinsischen Parameter (Kameramatrix und Verzeichnungskoeffizienten) in der GUI. Mit Klick auf das Ordner-Symbol können die Parameter geladen werden. Rechts: Spezifikation der YAML-Datei für intrinsische Parameter.

- Ein Feld K mit der Kameramatrix im Format 3x3,
- ein Feld kc mit den Verzeichnungskoeffizienten.

Die Syntax ist die gleiche wie die der Felder K_i , kc_i der Projektdatei. Abb. 4.3 rechts zeigt ein entsprechendes Beispiel. Die numerischen Werte in der GUI können nach Doppelklick auf das entsprechende Feld verändert werden.

4.4 Darstellung der extrinsischen Parameter in 3D-EasyCalib

Das Kamerakoordinatensystem ist stets wie in Abb. 2.1 definiert: die Z-Achse ist mit der optischen Achse identisch und zeigt in das Bild, die X-Achse ist parallel zu den Bildzeilen und die Y-Achse parallel zu den Bildspalten ausgerichtet. Der Koordinatenursprung liegt im Abstand f von der Bildebene entfernt.

Orientierung und Lage einer Kamera in einem beliebigen Koordinatensystem werden mit der Rotationsmatrix R und dem Translationsvektor $\mathbf{t}$ beschrieben. Hier stellt R die Rotationsmatrix der Kamera mit dem Ursprung in C bezüglich einer Welt W dar, und $\mathbf{t}$ ist der Vektor von C zum Ursprung des Weltkoordinatensystems. Mit anderen Worten: Punkt $\mathbf{X}$ wird vom Welt- in das Kamera-Koordinatensystem transformiert:

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

Die in der GUI dargestellten extrinsischen Parameter (siehe Abb. 4.4) haben folgende Bedeutung:

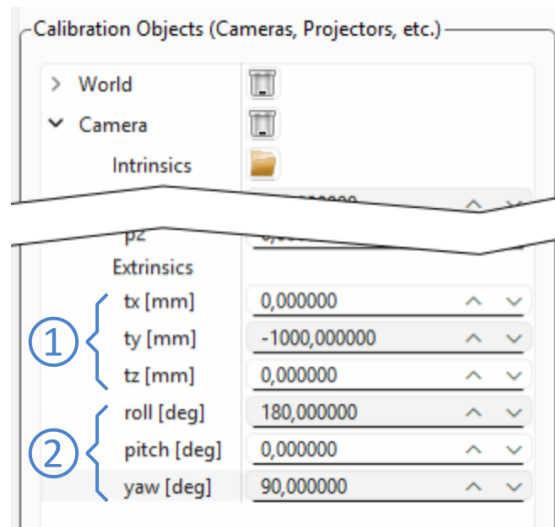


Abb. 4.4: Darstellung der extrinsischen Parameter (Translationsvektor $\mathbf{t}$ und Rotation R mit Eulerschen Winkeln.)

- ① Der Translationsvektor $\mathbf{t}$ ist mit den Komponenten t_x , t_y und t_z in Millimeter angegeben.
- ② Die Rotation R ist durch die *Eulerschen Winkel* definiert, d. h. durch Gier-, Nick- und Rollwinkel in Grad (engl. yaw, pitch, roll). „Gieren“ ist die Rotation um die z-Achse des Weltkoordinatensystems um den Winkel yaw, „Nicken“ die Rotation um die y-Achse um den Winkel pitch und „Rollen“ die Rotation um die x-Achse des Weltkoordinatensystems um den Winkel roll. Die Drehungen werden in dieser Reihenfolge durchgeführt. Alle Winkel werden rechtshändig in Grad gemessen.

Die Werte der extrinsischen Parameter können mit Doppelklick auf das entsprechende Feld und Tastatur-Eingabe verändert werden.

4.5 Bezugssystem und Formelnotation

Von den Objekten, die im Hauptfenster links aufgelistet sind, dient genau eines als Bezugsobjekt. Das mit diesem Objekt verbundene Koordinatensystem wird als Bezugssystem bzw. Weltkoordinatensystem definiert. Die extrinsischen Parameter der anderen Objekte sind als linear-affine Transformation vom Bezugsobjekt zu dem betreffenden Objekt zu interpretieren. Das Bezugsobjekt wird vom Programm nicht direkt gekennzeichnet. Es ist aber über folgende Merkmale indirekt festgelegt:

- In der Objektliste sind alle extrinsischen Parameter des Bezugsobjektes gleich null.
- Beim Start einer „symmetrischen“ Kalibrierungsaufgabe (derzeit nur die extrinsische Stereo-Kalibrierung) wird das Bezugssystem vom Nutzer festgelegt, siehe [Kapitel 6](#).
- Beim Start einer „asymmetrischen“ Kalibrierungsaufgabe, z. B. der Projektor-Kalibrierung, dient ein bestimmter Objekttyp als Bezugsobjekt (z. B. die Kamera). Dies ist in den Kapiteln 7 bis 9 für jeden Fall dokumentiert.
- Nach der Kalibrierung kann das Bezugsobjekt bei Speichern des Projektes geändert werden. Beim Aufruf des Menübefehls „File → Save Project“ erscheint eine entsprechende Abfrage. Nach erneutem Öffnen des Projektes wird die Änderung wirksam.

Um die beteiligten Koordinatensysteme in einer Transformationsgleichung eindeutig zu kennzeichnen, wird eine linear-affine Transformationsmatrix vorzugsweise durch folgendes Symbol dargestellt:

$${}^D T_S$$

Hierbei sind

- D das Symbol für das Zielobjekt der Transformation,
- S das Symbol für das Objekt, von dem die Transformation ausgeht,
- T das Grundsymbol der Transformation.

Allgemein wird die linear-affine Transformation durch eine homogene Matrix entsprechend (2.3) beschrieben, die eine Rotation ${}^D R_S$ und eine Translation ${}^D \mathbf{t}_S$ zusammenfasst:

$${}^D T_S = \begin{bmatrix} {}^D R_S & {}^D \mathbf{t}_S \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (4.1)$$

Der Koordinatenvektor eines Punktes wird durch ein Grundsymbol mit einem linken oberen Index dargestellt, wobei der Index dasjenige Objekt bezeichnet, auf dessen Koordinatensystem sich der Vektor bezieht. Beispielsweise ist

$${}^S \mathbf{P} = (X, Y, Z, 1)^T$$

ein homogener Koordinatenvektor bezüglich eines Objektes S .

Die Zusammensetzung von Transformationen ergibt sich durch Multiplikation der entsprechenden Matrizen. Dabei werden die Transformationen in der Reihenfolge vom rechten Faktor zum linken Faktor ausgeführt. Das Zielobjekt einer Teil-Transformation muss mit dem Startobjekt der nächsten Teil-Transformation übereinstimmen. Deshalb ist für einen gültigen Ausdruck in der o. g. Notation zu fordern, dass benachbarte Indizes gleich sind. Ein unterer Index muss mit dem rechts daneben stehenden oberen Index übereinstimmen. Diese Indexregel wird auch auf den Fall angewendet, dass das Produkt rechts mit einem Koordinatenvektor zu einer Punkt-Transformation erweitert wird. Ein gültiger Ausdruck sieht etwa folgendermaßen aus:

$${}^C\mathbf{P} = {}^C M_B \cdot {}^B M_A \cdot {}^A\mathbf{P}. \quad (4.2)$$

Die Formel beschreibt die Transformation eines Punktes vom System A über ein System B zu einem Zielsystem C .

4.6 Unterstützte Bildformate

3D-EasyCalib unterstützt die gängigsten Bildformate wie PNG, JPEG, TIFF und BMP. Alle Formate sind sowohl als 8-bit wie auch 16-bit lesbar. Die vollständige Liste der unterstützten Bildtypen ist:

- Windows bitmaps (*.bmp, *.dib)
- JPEG files (*.jpeg, *.jpg, *.jpe)
- JPEG 2000 files (*.jp2)
- Portable Network Graphics (*.png)
- WebP (*.webp)
- Portable Any Map (*.pbm, *.pgm, *.ppm)
- Sun raster (*.sr, *.ras)
- TIFF files (*.tiff, *.tif)

Hinweis

Es wird empfohlen, die aufgenommenen Bilder in einem verlustfrei komprimierten Bildformat wie Portable Network Graphics (PNG) zu speichern. JPEG ist wenig geeignet.

Intrinsische Kamerakalibrierung

5.1 Grundlagen

Ziel einer intrinsischen Kalibrierung ist die Bestimmung der inneren (intrinsischen) Parameter einer Kamera. Von Interesse sind die Elemente der Kameramatrix K sowie die Verzeichnungskoeffizienten laut (2.7). Letztere werden zu einem Verzeichnungsvektor $\mathbf{k}_c$ zusammengefasst. Durch Verbindung des Lochkameramodells (2.4) mit dem Verzeichnungsmodell erhält man eine allgemeinere Transformationsgleichung der Form

$$\mathbf{x} = f(K, \mathbf{k}_c, R, \mathbf{t}; \mathbf{X}) \quad (5.1)$$

mit den homogenen Objektkoordinaten $\mathbf{X}$, den homogenen Bildkoordinaten $\mathbf{x}$, den inneren Kameraparametern $K, \mathbf{k}_c$ und den äußeren Parametern (Rotation und Translation) $R, \mathbf{t}$.

Voraussetzung für die Kalibrierung ist die Aufnahme eines Kalibriertargets in unterschiedlichen Posen, siehe Abb. 5.1. Bei jeder Aufnahme betrachten wir ein auf dem Kalibriertarget fixiertes Koordinatensystem als Welt- bzw. Referenzsystem. Bei ebenen Targets wird das Kalibrieremuster üblicherweise in die X-Y-Ebene gelegt, so dass alle Kalibrierpunkte im Referenzsystem die Koordinate $Z = 0$ erhalten. Da das Target bewegt wird, sind Rotation und Translation der Kamera bei jeder Aufnahme unterschiedlich. Zwar bleibt die Kamera raumfest, entscheidend ist jedoch die Relativbewegung zwischen Kalibriertarget und Kamera. Die Spezialisierung der Formel (5.1) für Aufnahme Nr. i und Kalibrierpunkt Nr. j lautet:

$$\mathbf{x}_{ij} = f(K, \mathbf{k}_c, R_i, \mathbf{t}_i; \mathbf{X}_j). \quad (5.2)$$

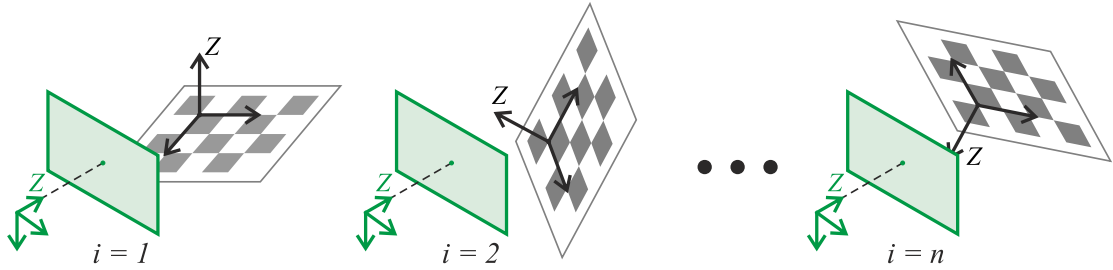


Abb. 5.1: Der Kalibrierdatensatz mit einem planaren Target setzt sich aus mehreren Aufnahmen des Targets in unterschiedlichen Posen zusammen.

Die Indizierung verdeutlicht, welche Größen von der Aufnahme bzw. dem Kalibrierpunkt abhängig sind.

Bekannt sind die Kalibrierpunkte $\mathbf{X}_j$ relativ zum Target. Aus den Kamerabildern werden die Bildkoordinaten $\hat{\mathbf{x}}_{ij}$ der Kalibrierpunkte ermittelt, die von den theoretischen Punkten in (5.2) im Allgemeinen geringfügig abweichen. Unbekannt sind die Parameter $K, \mathbf{k}_c, R_i, \mathbf{t}_i$, die aus der Gesamtheit der Gleichungen (5.2) zu bestimmen sind. Dieses nichtlineare Gleichungssystem ist durch die Vielzahl der Aufnahmen und Kalibrierpunkte überbestimmt. Es muss im Sinne einer Minimierung der Fehlerquadratsumme

$$\sum_{i,j} \|\hat{\mathbf{x}}_{ij} - \mathbf{x}_{ij}\|^2 = \sum_{i,j} \|\hat{\mathbf{x}}_{ij} - f(K, \mathbf{k}_c, R_i, \mathbf{t}_i; \mathbf{X}_j)\|^2 \quad (5.3)$$

iterativ gelöst werden. Vielfache Überbestimmtheit ist zwecks Reduzierung statistischer Fehler sogar erwünscht. Eine etablierte Lösungsmethode für dieses Problem stammt von Zhang [20] und wurde in Matlab [2] und OpenCV [3] implementiert.

Die Lösungen für die Kameraposen $R_i, \mathbf{t}_i$ gehören nicht zu den intrinsischen Parametern und können verworfen werden. In 3D-EasyCalib werden sie jedoch zur Visualisierung der Targetpositionen im „Target Preview“ (mittlere Ansicht des Hauptfensters) verwendet.

5.2 Reprojektionsfehler

Die in der Gleichung (5.3) enthaltenen euklidischen Abstände zwischen den theoretischen Bildpunkten $f(K, \mathbf{k}_c, R_i, \mathbf{t}_i; \mathbf{X}_j)$ und den im Bild vermessenen Punkten $\hat{\mathbf{x}}_{ij}$ werden *Reprojektionsfehler* genannt. Diese Fehlermetrik, angegeben für jeden Kalibrierpunkt sowie als Gesamtfehler (RMS) für jede Kalibriertarget-Pose, ist ein Indiz für die Güte der Kalibrierung und wird im Programm visuell und numerisch dargestellt, siehe Abb. 5.2. Da bei realen Aufnahmen die Koordinaten der Passpunkte z. B. wegen

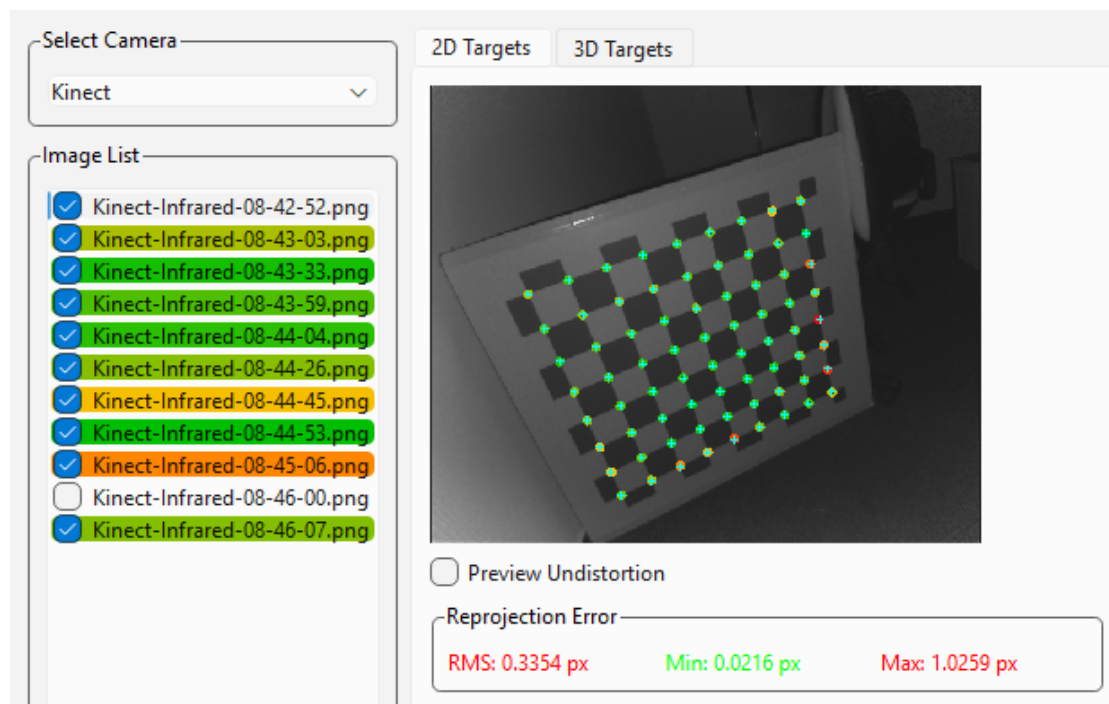


Abb. 5.2: Ausschnitt aus der Benutzeroberfläche der intrinsischen Kalibrierung nach einer Kalibrierung. In der Liste links sind die Kalibrierbilder entsprechend ihres Reprojektionsfehlers farblich markiert (Ausnahme: der Name des ausgewählten Bildes ist anders markiert). Die Target-Aufnahme rechts ist mit den detektierten Kalibrierpunkten (Kreise) und den berechneten Punkten (Kreuze) annotiert. Die Farbe der Kreise stellt den Reprojektionsfehler dar.

Verdeckung, Schattenwurf, Kamerarauschen oder Bewegung während der Aufnahme unpräzise oder falsch detektiert werden können, entstehen potentielle Ausreißer, die farblich gekennzeichnet sind. Aufnahmen mit zu vielen Ausreißern können aus dem Kalibrierdatensatz entfernt werden.

5.3 Eingabe

Von dieser Stelle an bis zu [Abschnitt 5.7](#) wird der Ablauf der intrinsischen Kalibrierung mittels 3D-EasyCalib beschrieben. Die Benutzeroberfläche der intrinsischen Kalibrierung ist in [Abb. 5.3](#) dargestellt und enthält folgende Elemente zur Eingabe:

- ① **Select Camera:** Die zu kalibrierende Kamera soll in der Combo-Box ausgewählt werden. Hier stehen die im Hauptfenster ([Kapitel 4](#)) aufgelisteten Objekte zur

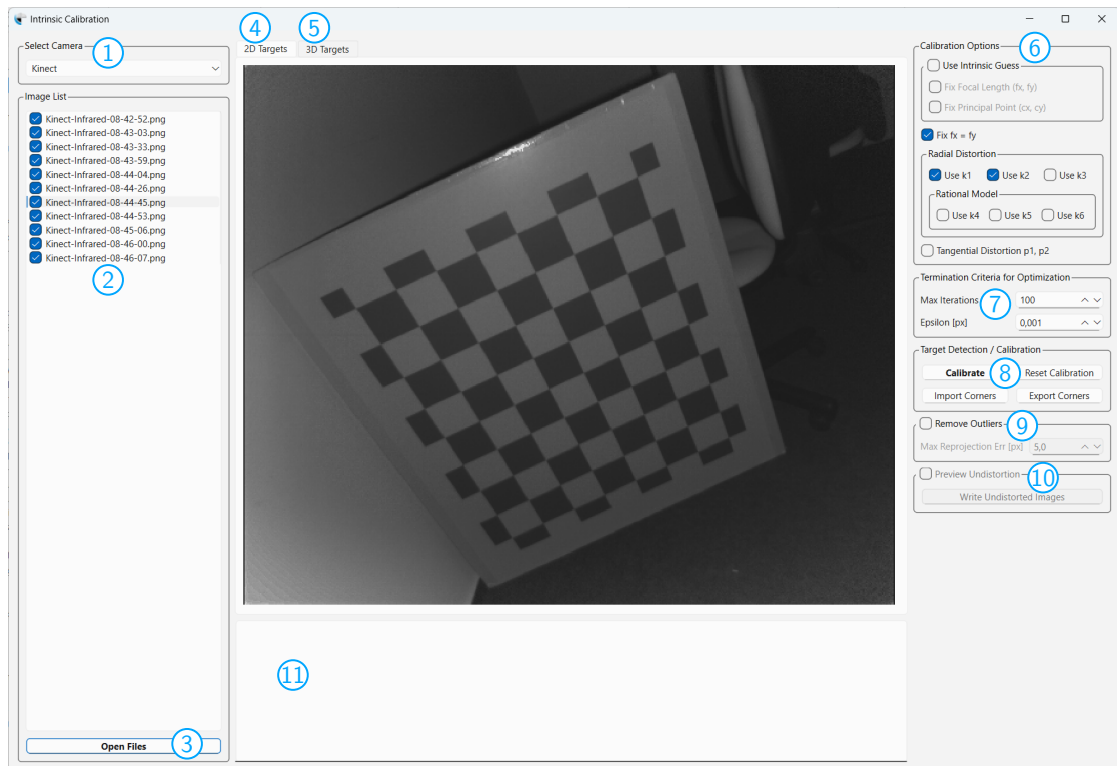


Abb. 5.3: Benutzeroberfläche für die intrinsische Kamerakalibrierung. Zustand vor der Kalibrierung.

Auswahl.

- ② **Image List:** In der Bildliste sind die geladenen Bilder mit den Dateinamen dargestellt. Ein ausgewähltes Bild wird in der Mitte des Fensters angezeigt. Mit der Check-Box vor den Dateinamen können einzelne Bilder von der Kalibrierung ausgeschlossen werden.
- ③ [OPEN FILES]: Öffnet den Dialog zum Laden von Bildern des Kalibriertargets in verschiedenen Posen. Die unterstützten Bildformate sind im Abschnitt 4.6 genannt. Alle benötigten Bilder müssen im Auswahldialog markiert und in einem Schritt geladen werden. Ein Nachladen von Bildern ist nicht möglich.

5.4 Visualisierung

Die Benutzeroberfläche der intrinsischen Kalibrierung (Abb. 5.3) enthält folgende Elemente zur Visualisierung:

- ④ **2D Targets:** In diesem Bereich wird das ausgewählte Bild angezeigt. Falls Verzeichnungskoeffizienten berechnet wurden, kann das Bild durch Markieren der Check-Box „Preview Undistortion“ entzerrt werden. Nach der Kalibrierung werden die Reprojektionsfehler für das Target unter der Bildansicht angezeigt: RMS über alle Punkte auf dem Target sowie minimaler und maximaler Fehler des Targets (siehe [Abb. 5.2](#) unten).
- ⑤ **3D Targets:** Nach der Kalibrierung werden die Kalibriernormale und das Kamera-Koordinatensystem dreidimensional dargestellt. Die Farbe des Targets entspricht seinem Reprojektionsfehler: grün kleiner, gelb mittlerer, rot großer Fehler.

Dargestellte Bilder können mit dem Mausrad vergrößert und verkleinert werden. In der 3D-Ansicht lässt sich der Bildinhalt mittels linker Maustaste drehen und in Kombination mit der Umschalttaste verschieben.

5.5 Einstellungen und Aktionen

Die Benutzeroberfläche der intrinsischen Kalibrierung ([Abb. 5.3](#)) enthält folgende Elemente zur Einstellung und Aktivierung:

- ⑥ **Calibration Options:** Folgende Optionen stehen zur Verfügung:
 - **Use intrinsic guess:** Innere Kameraparameter werden als Initialwerte für die Optimierung verwendet. Welche dies sind, wird durch die darunter stehenden, mit „Fix ...“ beschrifteten Optionen festgelegt. Wenn noch keine Parameter geladen oder berechnet wurden, ist eine anschließende Kalibrierung nicht erfolgreich.
 - **Fix focal length f_x, f_y :** Die effektiven Bildweiten bleiben während der Optimierung unverändert.
 - **Fix principal point $pp(c_x, c_y)$:** Der Hauptpunkt bleibt während der Optimierung unverändert.
 - **Fix aspect ratio f_x/f_y bzw. Fix $f_x = f_y$:** Ist **Use intrinsic guess** aktiviert, wird das Verhältnis der Bildweiten f_x/f_y während der Optimierung auf dem Initialwert fixiert. Ist **Use intrinsic guess** nicht aktiviert, werden die Bildweiten intern mit $f_x = f_y$ initialisiert und dieses Verhältnis beibehalten, sodass die Option effektiv $f_x = f_y$ erzwingt. Die Beschriftung der Option passt sich entsprechend an.

- **Radial Distortion:** Radiale Verzeichnungskoeffizienten werden entsprechend den nachfolgenden Optionen berechnet. Bei Abwahl aller Option werden alle Koeffizienten auf null gesetzt und fixiert. Die Theorie ist in [Abschnitt 2.2](#) beschrieben.
- **Use k1, Use k2, Use k3:** Aktiviert die Optimierung der einzelnen radialen Koeffizienten (gewählt) oder setzt sie auf null (abgewählt).
- **Rational model:** Die radiale Verzeichnung wird mittels der Koeffizienten k_4 , k_5 , k_6 durch eine gebrochen-rationale Funktion modelliert, siehe (2.7). Damit lässt sich eine starke Verzeichnung genauer approximieren.
- **Use tangential distortion p1, p2:** Bei Aktivierung wird ein tangentialer Anteil der Verzeichnung berücksichtigt. Bei Abwahl werden die Koeffizienten auf null gesetzt. Dieser Anteil kann bei großen Fertigungstoleranzen signifikant sein.

⑦ **Termination Criteria for Optimization:** Wenn sich während der Optimierung der Reprojektionsfehler für alle Targetaufnahmen in zwei aufeinanderfolgenden Iterationsschritten um weniger als **Epsilon** verändert oder die Anzahl der Iterationen (**Max Iterations**) überschritten wird, dann wird die Optimierung beendet.

⑧ **Target Detection / Calibration:**

- [CALIBRATE]: Die inneren Kameraparameter werden durch iterative Minimierung der Fehlerquadratsumme (5.3) berechnet. Wenn die Option **Remove Outliers** aktiviert ist, wird getestet, ob die Reprojektionsfehler aller Kalibrierpunkte unterhalb des Grenzwertes liegen. Falls nicht, wird der Nutzer über das Log-Fenster aufgefordert, die Aktion [CALIBRATE] zu wiederholen.
- [RESET CALIBRATION]: Alle Bilder der Bildliste werden für die nächste Kalibrierung freigegeben. Dies ist nützlich, wenn bei einer vorherigen Kalibrierung einige Bilder ausgeschlossen wurden. Außerdem werden eigene Passpunkte, die vorher mit [IMPORT CORNERS] geladen wurden, wieder entfernt.
- [IMPORT CORNERS]: Ermöglicht die Verwendung eigener Passpunkte, siehe [Abschnitt 5.7](#).
- [EXPORT CORNERS]: Speichert die Passpunkte der aktuellen Kalibrierung im Arbeitsverzeichnis. Zu jedem Bild wird eine YAML-Datei der Passpunkte mit gleichem Basisnamen erzeugt.

- ⑨ **Remove Outliers:** Wenn aktiviert, werden vor der nächsten Kalibrierung die Ausreißer in allen Target-Bildern beseitigt. Als „Ausreißer“ gelten Kalibrierpunkte, deren Reprojektionsfehler größer ist als die unter **Max Reprojection Err [px]** angegebene Grenze.
- ⑩ **Preview Undistortion:** Bei Aktivierung werden die Bilder mit Hilfe der aktuellen Verzeichnungskoeffizienten korrigiert und angezeigt. Mittels [WRITE UNDISTORTED IMAGES] werden die korrigierten Bilder im Unterverzeichnis `Undistorted` des Arbeitsverzeichnisses gespeichert.

5.6 Ausgabe

Nach der Kalibrierung werden die berechneten intrinsischen Parameter in das Hauptfenster übertragen ([Abschnitt 4.3](#)), als Datei

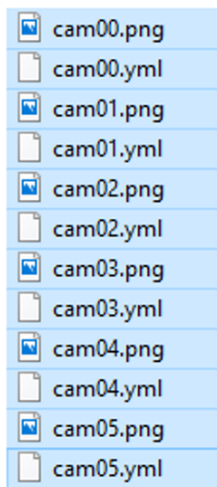
```
<Objektname>.intrinsics.yml
```

im Arbeitsverzeichnis gespeichert und im Log-Fenster ausgegeben. Die Datei kann verwendet werden, um in der Objektliste (Hauptfenster) die intrinsischen Parameter einer Kamera als Vorgabewerte einer weiteren Kalibrierung zu laden.

- ⑪ **Log-Fenster:** Dient zur Protokollierung der Ereignisse während des Kalibriervorganges. Hinter dem Text „Detecting targets“ werden erfolgreich erkannte Targets durch eine grüne, nicht erkannte Targets durch eine rote und abgewählte Targets durch eine graue Zahl repräsentiert. Nach der Kalibrierung werden der RMS-Wert des Reprojektionsfehlers sowie die optimierten intrinsischen Parameter ausgegeben. Die Abweichungen (+/-) entsprechen der dreifachen Standardabweichung.

5.7 Verwendung eigener Passpunkte

Es besteht die Möglichkeit, korrespondierende Objekt- und Bildpunkte für die Aufnahmen eines Kalibriertargets vorzugeben. Dies ist z. B. für ein Target erforderlich, das vom Schachbrettmuster abweicht. Hierzu wird im Ordner der Targetbilder zu jedem Bild eine Datei angelegt, die den gleichen Basisnamen wie das Bild, aber die Erweiterung „.yml“ hat. Der Bildordner enthält dann beispielsweise die in [Abb. 5.4](#) links gelisteten Dateien. Die eigenen Passpunkte werden aber nur dann verwendet,



```
%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. ]
```

Abb. 5.4: Links: Beispiel für den Ordnerinhalt bei Vorgabe von Kalibrierpunkten. Rechts: Beispiel einer YAML-Datei mit den Kalibrierpunkt-Koordinaten. Die Bild- und Objektkoordinaten einander zugeordneter Punkte sind in der gleichen Farbe dargestellt.

wenn sie vor der Kalibrierung mittels Schaltfläche [IMPORT CORNERS] geladen wurden. Für Bilder, zu denen keine YAML-Dateien existierten, wird bei der Kalibrierung in jedem Fall der Eckendetektor aktiviert.

Der Aufbau einer solchen YAML-Datei ist in [Abb. 5.4](#) rechts dargestellt. Variabel sind lediglich die Zahlenwerte hinter rows und data. Der übrige Text ist in allen Dateien dieser Art gleich. Es gilt folgende Syntax:

- Hinter rows im Abschnitt corners und im Abschnitt obj steht jeweils die Anzahl der korrespondierenden Punktpaare. Beide Zahlen müssen gleich sein.
- Hinter data im Abschnitt corners werden die x- und y-Koordinaten der Bild-

punkte aufgelistet, immer x und y abwechselnd für die aufeinanderfolgenden Punkte. Die Liste muss also $2 \cdot \text{rows}$ Zahlen enthalten.

- Hinter data im Abschnitt obj werden die X-, Y- und Z-Koordinaten der Kalibrierpunkte des Targets aufgelistet, immer X, Y, Z abwechselnd für die aufeinanderfolgenden Punkte. Die Liste muss also $3 \cdot \text{rows}$ Zahlen enthalten. Die Koordinaten beziehen sich auf das mit dem Target fest verbundene Koordinatensystem. Bei einem ebenen Target sind normalerweise alle Z-Werte gleich null. Falls von null verschiedene Z-Werte vorkommen, müssen in der Objektliste (Hauptfenster) Anfangswerte für die intrinsischen Parameter f_x , f_y , c_x , c_y eingetragen werden. Zusätzlich ist die Option **Use intrinsic guess** zu aktivieren.

Extrinsische Stereo-Kalibrierung

6.1 Grundlagen

Ziel der extrinsischen Stereo-Kalibrierung ist die Bestimmung der relativen Lage $\mathbf{t}$ und Orientierung R einer Kamera C_1 bezüglich einer Kamera C_2 , wie in [Abb. 6.1](#) dargestellt. Hierzu wird ein Target simultan von beiden Kameras erfasst. Die Pose des Targets aus Sicht der beiden Kameras wird berechnet und daraus die Lage und Orientierung der Kameras zueinander hergeleitet. Zur Fehlerminimierung sind wie bei der intrinsischen Kalibrierung mehrere Target-Aufnahmen in verschiedener Position und Ausrichtung des Targets erforderlich.

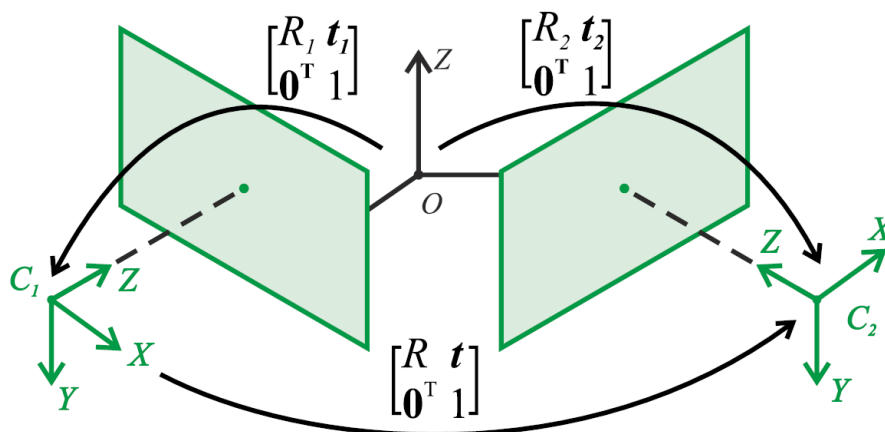


Abb. 6.1: Relative Lage und Orientierung der Kamera C_1 bezüglich der Kamera C_2

Die resultierenden *extrinsischen Parameter* R und $\mathbf{t}$ transformieren Punkte vom Koordinatensystem der ersten Kamera ($\mathbf{X}_1$) in das Koordinatensystem der zweiten Kamera ($\mathbf{X}_2$) wie folgt:

$$\mathbf{X}_2 = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_1. \quad (6.1)$$

Wird ein Weltpunkt $\mathbf{X}$ auf dem Target im ersten und dann im zweitem Kamerakoordinatensystem betrachtet, ergeben sich folgende Transformationsgleichungen für diesen Punkt (siehe Abb. 6.1):

$$\mathbf{X}_1 = \begin{bmatrix} R_1 & \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}, \quad (6.2)$$

$$\mathbf{X}_2 = \begin{bmatrix} R_2 & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}. \quad (6.3)$$

Die gesuchte Transformation (6.1) lässt sich herleiten, indem (6.2) nach $\mathbf{X}$ aufgelöst und in (6.3) eingesetzt wird. Das Ergebnis lautet:

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

Die gesuchten Parameter Rotation R und Translation $\mathbf{t} = \overrightarrow{C_1 C_2}$ lassen sich direkt ablesen:

$$R = R_2 R_1^T, \quad \mathbf{t} = \mathbf{t}_2 - R \mathbf{t}_1. \quad (6.5)$$

Die Größen $R_1, R_2, \mathbf{t}_1, \mathbf{t}_2$ werden in einem iterativen Prozess ähnlich der intrinsischen Kalibrierung für beide Kameras ermittelt. Denn die Lösung des Gleichungssystems der intrinsischen Kalibrierung (5.2) enthält für jede Target-Pose auch die Rotations- und Translationsparameter der betreffenden Einzelkamera.

Voraussetzungen für eine Stereokalibrierung:

- Lage und Orientierung der beiden Kameras bleiben während der Aufnahme der Kalibriertargets unverändert.
- Jedes Kalibriertarget wird von den beiden Kameras vollständig und zur gleichen Zeit erfasst.

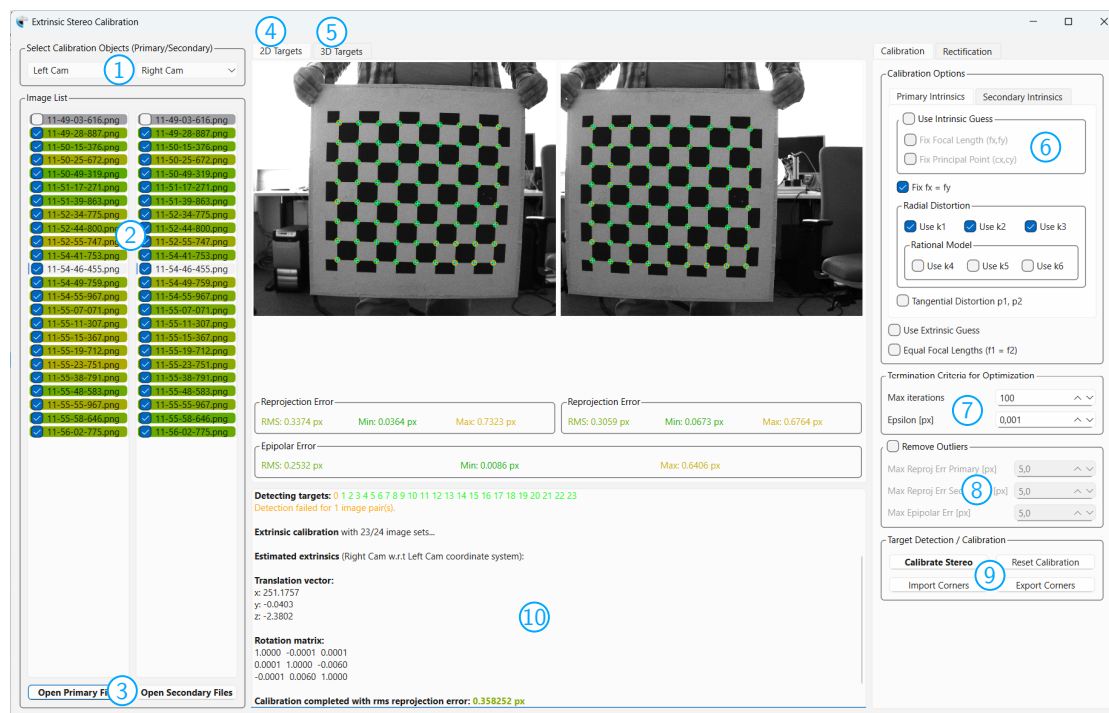


Abb. 6.2: Benutzeroberfläche für die Stereo-Kalibrierung. Zustand nach einer Kalibrierung.

Hinweis

Idealerweise wurden die Kameras bereits intrinsisch kalibriert, so dass nur Rotation und Translation zu bestimmen sind. Aufgrund der internen Verwendung der Gleichungen für die intrinsische Kalibrierung werden jedoch fehlende intrinsische Parameter automatisch berechnet.

6.2 Eingabe

Die GUI für die extrinsische Kamerakalibrierung (siehe Abb. 6.2) ähnelt der GUI für die intrinsische Kalibrierung, sie wurde jedoch um Ein- und Ausgaben für die zweite Kamera erweitert. Die GUI enthält folgende Elemente zur Eingabe des Bilddatensatzes:

- ① **Select Calibration Objects (Primary/Secondary):** Die zu kalibrierenden Kameras können in den Combo-Boxen ausgewählt werden. Es stehen die im Hauptfenster angelegten Objekte zur Auswahl.

- ② **Image List:** In der Bildliste sind die Dateinamen der geladenen Bilder enthalten. Nach Auswahl eines Bildes wird das entsprechende Bildpaar in der Mitte des Fensters angezeigt. Mit der Check-Box vor dem Dateinamen können einzelne Bilder von der Kalibrierung ausgeschlossen werden. Das Abwählen einer Check-Box in einer Spalte führt dazu, dass bei der Kalibrierung auch das zugehörige Bild in der zweiten Spalte ignoriert wird.

Nach der Kalibrierung wird jeder Eintrag in der Liste entsprechend dem Reprojektionsfehler farblich codiert. Rote Farbe entspricht dem maximalen und grüne dem minimalen Fehler.

- ③ **Open {Primary | Secondary} Files:** Für die jeweilige Kamera werden mit diesem Button die Bilder geladen, Man achte darauf, die Bilder in gleicher Reihenfolge für beide Kameras zu laden. Die unterstützten Bildformate sind in [Abschnitt 4.6](#) gelistet.

6.3 Visualisierung

Folgende Elemente der GUI dienen zur Visualisierung:

- ④ **2D Targets:** In diesem Bereich wird das ausgewählte Bildpaar gezeigt. Nach der Kalibrierung werden unter dem jeweiligen Bild (siehe Abb. 6.2) die Reprojektionsfehler (Reprojection Error) für das Target gelistet: RMS, Minimum und Maximum über alle Kalibrierpunkte des Targets.

Darunter wird zudem der Epipolarfehler über alle Kalibrierpunkte auf den beiden Bildern ausgegeben. Epipolarfehler ist der kleinste Abstand eines Kalibrierpunktes in einem Bild zur dazugehörigen Epipolarlinie im zweiten Bild. Diese Fehler stellen insbesondere die Effekte der optimierten Lage und Orientierung der Kameras zueinander dar.

- ⑤ **3D Targets:** Nach der Kalibrierung werden die Kalibriernormale zu den in „Image List“ ausgewählten Bildpaaren und die Kameras in ihrer räumlichen Lage zueinander dargestellt¹ (siehe Abb. 6.3). Die Farbe des Targets entspricht der Größe des Reprojektionsfehlers, gemittelt über die beiden Kameras.

¹soweit es sich um Schachbrettmuster handelt

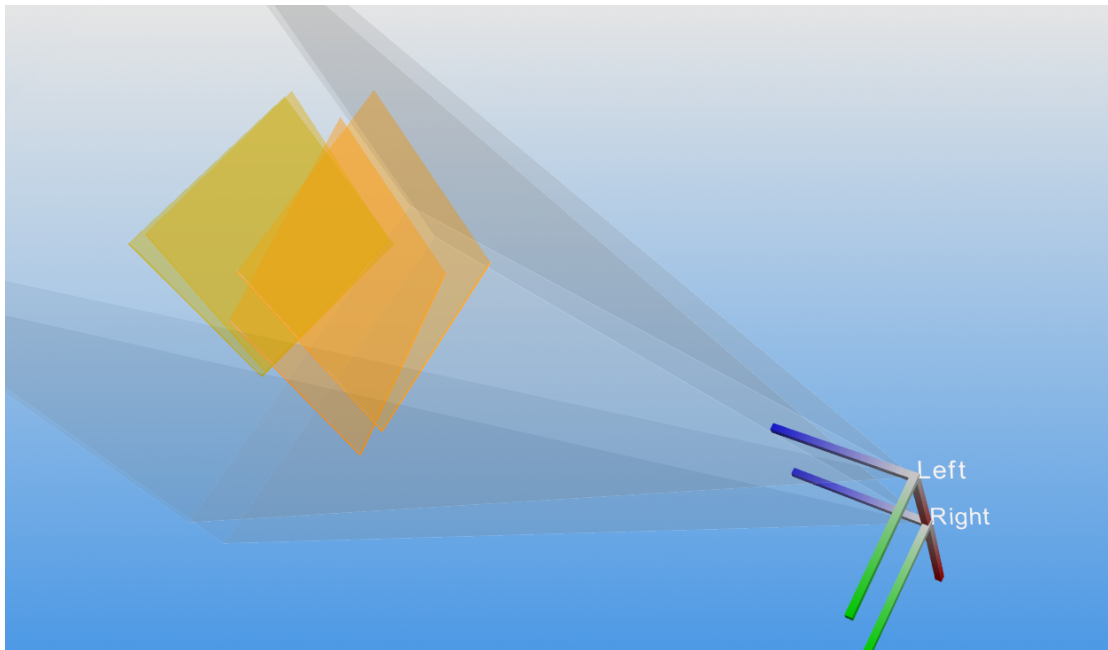


Abb. 6.3: 3D-Darstellung der Kameras und Kalibrier-Targets. Angedeutet sind auch die pyramidenförmigen Sichtfelder der Kameras.

6.4 Einstellungen und Aktionen

Folgende Elemente der GUI (siehe [Abb. 6.2](#)) dienen zur Festlegung von Kalibrier-Optionen und zum Starten des Kalibriervorgangs. Die Parameter des Kalibrier-Targets (Schachbrett oder ChArUco) müssen vorher im Hauptfenster festgelegt werden.

- ⑥ **Calibration Options:** Die meisten Einstellungen sind für beide Kameras getrennt vorzunehmen. Der Wechsel zur entsprechenden Kamera erfolgt über die Tab-Reiter **Primary Intrinsic**s und **Secondary Intrinsic**s.

Hinweis

Diese Einstellungen sind mit denen für die intrinsische Kalibrierung identisch und können in [Abschnitt 5.5](#) ab Seite 37 nachgelesen werden.

Zusätzlich sind folgende spezifische Einstellungen verfügbar:

- **Use Extrinsic Guess:** Wenn gewählt, werden die vorhandenen Parameter der extrinsischen Kalibrierung verwendet. Entscheidend sind die in der Objektliste (Hauptfenster) angezeigten Parameter. Bei Bedarf sind die

Objekt-Einträge aufzuklappen. Die Parameter können durch Öffnen einer Projektdatei über das Hauptmenü geladen werden.

- **Equal focal lengths $f_1=f_2$:** Erzwingt bei der Optimierung die Gleichsetzung der Brennweitenparameter f_x für beide Kameras und f_y für beide Kameras.

⑦ **Termination Criteria for Optimization:** Wenn sich während der Optimierung der Reprojektionsfehler für alle Targetaufnahmen in zwei aufeinanderfolgenden Iterationsschritten um weniger als Epsilon verändert oder die Anzahl der Iterationen (Max Iterations) überschritten wird, dann wird die Optimierung beendet.

⑧ **Remove Outliers:** Wenn aktiviert, werden vor der nächsten Kalibrierung die Ausreißer in allen Target-Bildern beseitigt. Als „Ausreißer“ gelten Kalibrierpunkte,

- deren Reprojektionsfehler größer ist als die unter **Max Reprojection Error** ... [px] angegebene Grenze für die jeweilige Kamera oder
- deren Epipolar-Abweichung größer ist als die angegebene Grenze.

⑨ **Target Detection / Calibration:**

- [CALIBRATE STEREO] führt die Optimierung aus. Die Parameter werden so optimiert, dass die Summe der Reprojektionsfehler über alle gültigen Kalibrierpunkte aller Targetbilder und Kameras minimal wird.
- [RESET CALIBRATION]: Alle Bildpaare der Bildliste werden für die nächste Kalibrierung freigegeben. Dies ist nützlich, wenn bei einer vorherigen Kalibrierung einige Bildpaare ausgeschlossen wurden. Außerdem werden eigene Passpunkte, die vorher mit [IMPORT CORNERS] geladen wurden, wieder entfernt.
- [IMPORT CORNERS]: Ermöglicht die Verwendung eigener Passpunkte, siehe [Abschnitt 5.7](#).
- [EXPORT CORNERS]: Speichert die Passpunkte der aktuellen Kalibrierung. Zu jedem Bild wird eine YAML-Datei der Passpunkte mit gleichem Basisnamen erzeugt und im Verzeichnis des Bildes gespeichert.

6.5 Ausgabe

Nach der Kalibrierung werden die berechneten extrinsischen (6.5) und gegebenenfalls intrinsischen Parameter in die Objektliste des Hauptfensters übertragen und



Abb. 6.4: Ergebnis der Bildentzerrung und Rektifizierung mit eingezeichneten Epipolarlinien (unten) der Eingabebilder (oben).

im Log-Fenster ausgegeben. Die Parameter können mittels Menübefehl **Save Project** des Hauptfensters als Datei gespeichert werden.

- 10 **Log-Fenster:** Dient zur Protokollierung der Ereignisse während des Kalibrierungsvorganges. Erfolgreich erkannte Targets werden mit einer grünen, nicht erkannte mit einer orangefarbenen und abgewählte Targets durch eine graue Zahl repräsentiert. Nach der Kalibrierung werden der RMS des Reprojektionsfehlers (rms reprojection error) sowie die optimierten extrinsischen Parameter ausgegeben.

6.6 Stereo-Rektifizierung

Bei einer Stereo-Korrespondenzanalyse ist es von Vorteil, wenn die Suche nach ähnlichen Strukturen in einem Stereo-Bildpaar auf die Bildzeilen oder -spalten einge-

schränkt ist. Diese Bedingung ist bei einer Standard-Stereo-Anordnung erfüllt: Die Bilder liegen auf einer Ebene, die beiden Kameras haben die gleichen intrinsischen Parameter (ohne Linsenverzeichnung) und sind achsparallel ausgerichtet. Die Epipolarlinien stimmen in diesem Fall mit den Bildzeilen überein. Bei bekannten inneren Kameraparametern und bekannter Rotations- sowie Translationsmatrix ist es möglich, die Bilder so geometrisch zu transformieren, als wären sie mit der Standard-Stereokonfiguration aufgenommen. Diese Transformation wird als Rektifizierung bezeichnet.

In 3D-EasyCalib wird analog der Methode von Bouguet [2] das jeweilige Bild zuerst auf die Fokalebene zurückprojiziert, dort gegebenenfalls entzerrt, in diesem euklidischen Koordinatensystem entsprechend rotiert und wieder auf die gemeinsame Ebene mit einer für beide Bilder gleichen Kameramatrix projiziert.

Die Eingabefelder für die Rektifizierung werden nachfolgend erläutert. Sie sind auf der rechten Seite des Kalibrierfensters über den Tabreiter **Rectification** zugänglich, siehe Abb. 6.5.

Abb. 6.5: Eingabefelder zur Rektifizierung

Equal principal points. Setzt die beiden Bildhauptpunkte der rektifizierten Bilder auf einen gemeinsamen Punkt. Das vereinfacht die 3D-Berechnung aus dem Disparitätsbild.

Resize rectified images. Skaliert die Bildweiten der rektifizierten Bilder so, dass diese Bilder die vorgegebene Bildgröße haben.

Crop rectified images. Der Parameter alpha definiert, wie viel von den rektifizierten (und entzerrten) Bildern noch sichtbar ist. Alpha gleich 1 bedeutet, dass alle Pixel der Originalbilder der Kameras in den rektifizierten Bildern erhalten bleiben (siehe Beispiel Abb. 6.4 unten). Alpha = 0 bedeutet, dass die entzerrten Bilder vergrößert und verschoben werden, so dass nur gültige Pixel sichtbar sind (keine schwarzen Bereiche nach der Entzerrung).

Calibrate Rectification Parameters. Führt die Rektifizierung durch. Die Ausgabe dieses

Prozesses ist unten erläutert.

Preview Rectified Images. Wenn aktiviert, werden im mittleren Teil des Kalibrierfensters unter dem Tabreiter **2D Targets** die rektifizierten Bildpaare mit Epipolarlinien angezeigt. Die Anzahl der Epipolarlinien kann im Eingabefeld festgelegt werden.

[WRITE RECTIFIED IMAGES]. Schreibt die rektifizierten Bilddateien in einen Unterordner innerhalb des Bildordners der linken bzw. rechten Kamera.

Ausgabe

Das Ergebnis der Rektifizierung wird als Datei

`<Arbeitsverzeichnis>/<Cam1>_<Cam2>.rectification.yml`

gespeichert. <Cam1> und <Cam2> sind die Namen der beteiligten Kameras in der Objektliste. Diese Datei enthält die Matrizen R0, R1, P0, P1 und Q, die zur Berechnung der Bildkorrektur bzw. zur 3D-Triangulation verwendet werden.

- R0, R1: Rotationsmatrizen für die erste bzw. zweite Kamera. Diese Rotationen drehen die jeweilige Kamera so, dass die beiden Kameras parallel ausgerichtet sind.
- P0, P1: 3x4-Projektionsmatrizen für die rektifizierten Kameras.
- Q: 4x4-Matrix, die eine Disparitätskarte in eine 3D-Punktwolke transformiert. Diese Matrix eignet sich als Argument der Funktion `reprojectImageTo3D()` der *OpenCV*.

$$P0 = \begin{bmatrix} f & 0 & c_{x1} & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.6)$$

$$P1 = \begin{bmatrix} f & 0 & c_{x2} & t_x f \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.7)$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_{x1} \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{t_x} & \frac{c_{x1}-c_{x2}}{t_x} \end{bmatrix} \quad (6.8)$$

Wenn **Equal principal points** gesetzt ist, dann sind in rektifizierten Bildern nicht nur die vertikalen, sondern auch die horizontalen Komponenten der Bildhauptpunkte gleich, $c_{x1} = c_{x2}$.

6.7 Verwendung eigener Passpunkte

Auch hier besteht die Möglichkeit, den eingebauten Eckendetektor zu umgehen und stattdessen eigene Passpunkte zu nutzen. Diese Punkte werden zu jeder Bilddatei in eine namensgleiche YAML-Datei geschrieben und im gleichen Ordner wie die Bilddatei gespeichert. Jede YAML-Datei muss die 3D-Koordinaten des Kalibriertargets und die entsprechenden 2D-Koordinaten des Kamerabildes enthalten. Die Syntax ist genau die gleiche wie bei der intrinsischen Kalibrierung, siehe [Abschnitt 5.7](#). Bei der Stereo-Kalibrierung müssen jedoch zu den Bilddateien beider Kameras YAML-Dateien vorliegen.

Projektor-Kalibrierung

7.1 Grundlagen

Bei einer Kamera wird die 3D-Szene auf die 2D-Bildebene projiziert, während bei einem Projektor ein ebenes 2D-Muster in den 3D-Raum abgebildet wird. Daher werden Projektoren häufig als inverse Kameras bezeichnet und modelliert. Die Theorie, die der optischen Abbildung zugrunde liegt, ist für eine Kamera und eine inverse Kamera (Projektor) gleich. Folglich können dieselben Kalibrier-Algorithmen verwendet werden.

Robot/World- und Tool/Flange-Kalibrierung

8.1 Grundlagen

Bei einer Kamera-zu-Roboter-Kalibrierung wird eine stationäre Kamera zu einem Roboter-System mithilfe eines Schachbrett-Kalibriertargets registriert. Hierzu soll das Kalibriertarget an den End-Effektor des Roboters fest montiert werden. Für den Kalibriervorgang muss das Target mit dem Roboter in unterschiedliche Positionen bewegt und mit der Kamera aufgenommen werden.

Roboterwerkzeug zu Kamera Kalibrierung (Hand/Eye)

9.1 Grundlagen

Das Problem der Hand-Auge-Kalibrierung (engl. Hand/Eye) besteht in der Berechnung der starren Transformation (Rotation und Translation) zwischen einer an einem Roboteraktuator montierter Kamera und dem Aktuator selbst, d. h. der euklidischen Transformation zwischen dem Kamera- und dem Werkzeugkoordinatensystem. Diese Transformation wird mithilfe eines im Raum fixierten Kalibriertargets bestimmt.

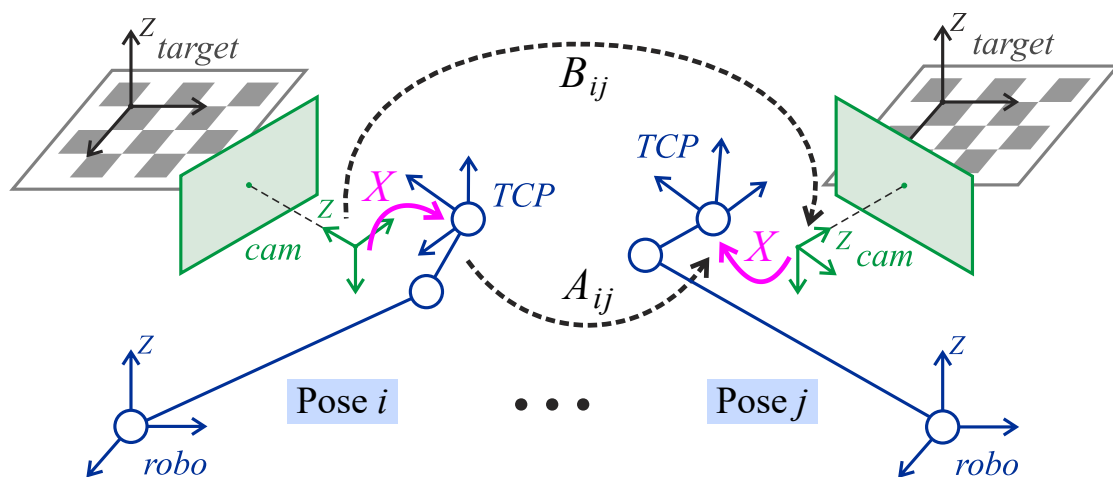


Abb. 9.1: Definition der Transformationen beim Hand-Auge-Problem



Literaturverzeichnis

Eine vollständige Dokumentation ist
Bestandteil jeder lizenzierten Programmversion.

Wir erstellen Ihnen gern auf Anfrage per eMail
3d-easycalib@zbs-ilmenau.de
ein attraktives Angebot!

