

# 3D-EasyCalib

## Toolkit zur Kalibrierung von Kameras und Robotern

DIE BEDIENUNGSANLEITUNG

Darko VEHAR und andere

ZBS e. V.

Werner-von-Siemens-Straße 12  
98693 Ilmenau



## **Impressum**

©2024 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

<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1 Einleitung</b>	<b>5</b>
1.1 Geschichte der Kamerakalibrierung . . . . .	5
1.2 Was ist der Zweck der Kamerakalibrierung? . . . . .	7
1.3 Was zeichnet eine gute Kamerakalibrierung aus? . . . . .	7
<b>2 Theorie des Lochkameramodells</b>	<b>9</b>
2.1 Parameter des Lochkameramodells . . . . .	11
2.2 Modell der Linsenverzeichnung . . . . .	12
<b>3 Kalibriertarget</b>	<b>15</b>
3.1 Raster des Schachbrett-Kalibriernormals . . . . .	15
3.2 Subpixelgenaue Bestimmung der Targetpunkte . . . . .	16
3.3 Hinweise für Aufnahmen von Kalibriertargets . . . . .	17
<b>4 Benutzeroberfläche des 3D-EasyCalib</b>	<b>21</b>
4.1 Aufbau der Projekt-Datei . . . . .	23
4.2 Darstellung der intrinsischen Parameter in 3D-EasyCalib . . . . .	23
4.3 Darstellung der extrinsischen Parameter in 3D-EasyCalib . . . . .	25
4.4 Beispiel einer Kamera-zu-Welt-Transformation . . . . .	26
4.5 Unterstützte Bildformate . . . . .	28
<b>5 Intrinsische Kamerakalibrierung</b>	<b>29</b>
5.1 Reprojektionsfehler . . . . .	30

5.2	Eingabe	30
5.3	Visualisierung	32
5.4	Einstellungen	32
5.5	Ausgabe	34
5.6	Verwendung eigener Passpunkte bzw. eines 3D-Normals	35
<b>6</b>	<b>Extrinsische Stereo-Kalibrierung</b>	<b>37</b>
6.1	Extrinsische Parameter $R$ und $t$	38
6.2	Eingabe	39
6.3	Visualisierung	40
6.4	Einstellungen	41
6.5	Ausgabe	43
6.6	Stereo-Rektifizierung	45
<b>7</b>	<b>Kamera-Projektor-Kalibrierung</b>	<b>47</b>
7.1	Eingabe	48
7.2	Einstellungen	52
7.3	Ausgabe	53
<b>8</b>	<b>Robot/World- und Tool/Flange-Kalibrierung</b>	<b>57</b>
8.1	Eingabe	58
8.2	Einstellungen	61
8.3	Ausgabe und Visualisierung	62
<b>9</b>	<b>Roboterwerkzeug zu Kamera Kalibrierung (Hand/Eye)</b>	<b>67</b>
9.1	Eingabe	68
9.2	Einstellungen	70
9.3	Ausgabe und Visualisierung	72
	<b>Literaturverzeichnis</b>	<b>75</b>

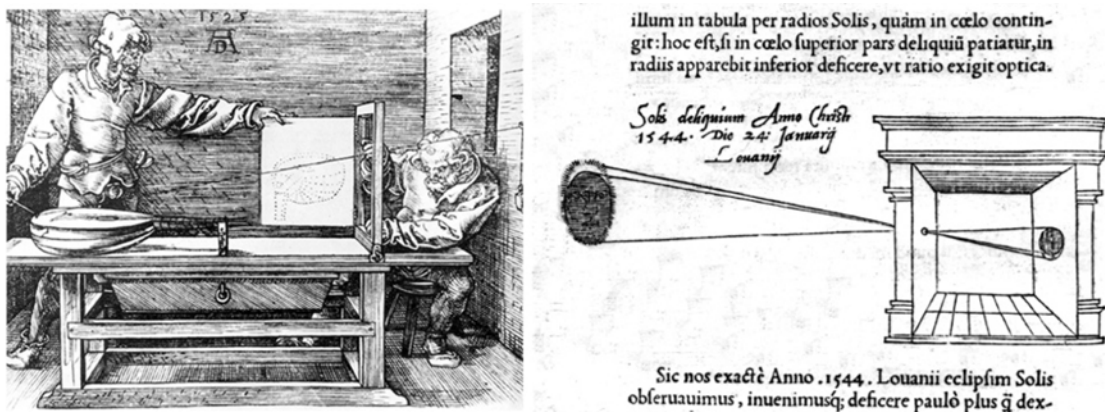
---

# 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 [Abbildung 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, Unterweisung 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 [8] 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 [22] 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 [23] 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 Mehrkammersystemen 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 Objektiveneinstellungen sollten unverändert bleiben (insbesondere der Autofokus sollte ausgeschaltet sein).

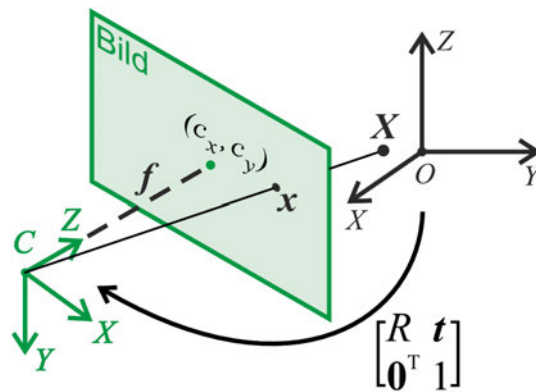
**Die korrekte Wahl der zu optimierenden Kameraparameter** hängt von der verwendeten Kamera und Optik ab. Obwohl viele freie Parameter zu kleineren Kalibrie-

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 Lochkammeramodells

Das Lochkammermodell 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



**Abb. 2.1:** Die Komponenten einer Transformation des 3D-Raums auf die 2D-Ebene, wie sie mit einer Lochkamera realisiert wird, bestehen aus der Bildebene, dem Punkt C, der nicht auf der Bildebene liegt, dem optischen Zentrum und der Bildweite  $f$ , dem Abstand zwischen dem Punkt 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  $(C, X)$  mit der Bildebene.

Strahl, der durch das Projektionszentrum der Kamera geht.

Das Modell lässt sich anhand der [Abbildung 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  $3 \times 4$ -Blockmatrix, die aus der  $3 \times 3$ -Einheitsmatrix  $I$  und dem rechtsstehenden Spaltenvektor  $\mathbf{0} = (0, 0, 0)^T$  besteht. Die  $3 \times 3$ -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, siehe Abbildung [Abbildung 2.1](#)) mit der Bildebene. 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 Normalfall 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.

Für die Verzeichnungskorrektur hat sich in der digitalen Bildverarbeitung das Brown-Conrady-Modell [5] etabliert. Das Modell berücksichtigt die sogenannte radiale und tangentiale Verzerrung. Dem ersten Typ liegen die geometrisch-optischen Eigenschaften eines zentrierten optischen Systems zugrunde. Die tangentiale Verzeichnung modelliert Abweichungen von der Rotationssymmetrie durch den Effekt eines dünnen Prismas. Bei der radialen Verzerrung wird der Bildpunkt radial, bei der tangentialen Verzerrung azimuthal zum Verzerrungszentrum verschoben. Die entzerrten Koordinaten lauten:

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

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

wobei der Radius  $r = \sqrt{x^2 + y^2}$  den Abstand vom Hauptpunkt zum unverzerrten Punkt  $(x, y)$  definiert. Die radialen Koeffizienten  $k_1, k_2, k_3$  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

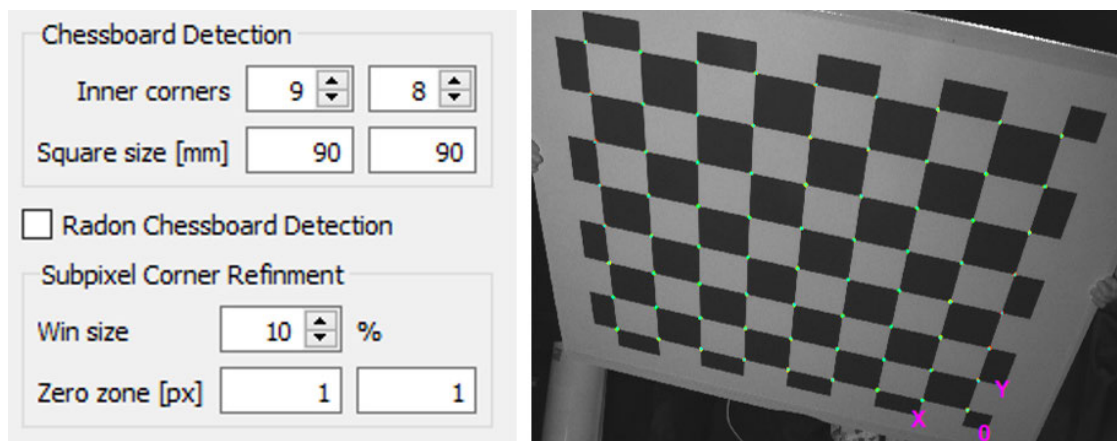
Die Voraussetzung für gute Kalibrierungsergebnisse ist ein präzise gefertigtes Kalibriernormal, auch Kalibriertarget genannt. Ein klassisches Target stellt mehrere Raum- (Pass-)Punkte mit bekannter geometrischer Lage im Raum (3D-Target) oder auf einer Ebene (2D- bzw. planares Target) bereit. Die Gestaltung 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, wie in der Abb. 3.1 rechts dargestellt, 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

**Inner corners** gibt die Anzahl der inneren Ecken eines Schachbrettmusters an. In Abb. 3.1 rechts sind diese Ecken durch grüne Punkte markiert. Die erste Zahl (hier 9) definiert die X-Achse des Targets und die zweite (hier 8) die Y-Achse. Mit 0 wird der Koordinatenursprung des Targets bezeichnet.

Um die Orientierung des Kalibriernormals eindeutig zu bestimmen, muss die Anzahl der Ecken in X- und Y-Richtung von verschiedener Parität sein. D.h. wenn die An-



**Abb. 3.1:** Einstellungen für die Erkennung eines 9x8-Kalibriertargets und die subpixelgenaue Vermessung der Passpunkte. Auf dem Target sind die Passpunkte grün und die Koordinatenachsen magenta markiert.

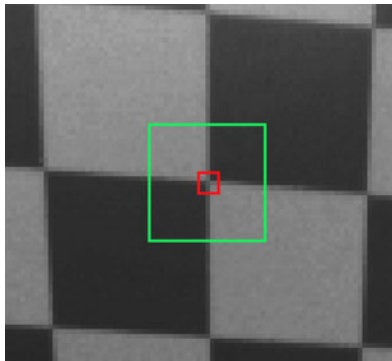
zahl der Ecken in X-Richtung ungerade ist, muss sie in Y-Richtung gerade sein und umgekehrt. Somit besitzt das Normal genau eine Symmetrieachse und ist im Raum eindeutig orientierbar.

**Square Size [mm]** gibt die Größe eines Quadrats auf dem Target an. Sie ist notwendig für die Erzeugung von Soll-Koordinaten des Kalibriernormals. 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.

## 3.2 Subpixelgenaue Bestimmung der Targetpunkte

Für die subpixelgenaue Berechnung der Passpunkte (Subpixel Corner Refinement in Abb. 3.1) stehen zwei Methoden zur Wahl: Die sogenannte Radon Chessboard Detection nach Duda [9] und ein klassischer Ansatz nach Förstner [10]. Für den Radon-Detektor ist keine Parametereinstellung notwendig und liefert bei verrauschten Bildern bessere Ergebnisse.

Wenn Radon Chessboard Detection nicht ausgewählt ist, werden die Passpunkte mit dem klassischen Ansatz berechnet. Das klassische Verfahren berechnet die subpixelgenaue Position der Ecken innerhalb eines Fensters und hat folgende Einstellmöglichkeiten:



**Win size.** Fenstergröße (grün) in Prozent um den Passpunkt. Die Zahlenangabe ist als kleinste horizontale oder vertikale Distanz zum benachbarten Passpunkt definiert.

**Zero zone.** Die Nullzone (rot) ist die Breite des von der Berechnung ausgeschlossenen Bereichs um den Passpunkt in Pixeln. Bei der Eingabe -1, -1 werden alle Punkte innerhalb des Suchfensters berücksichtigt.

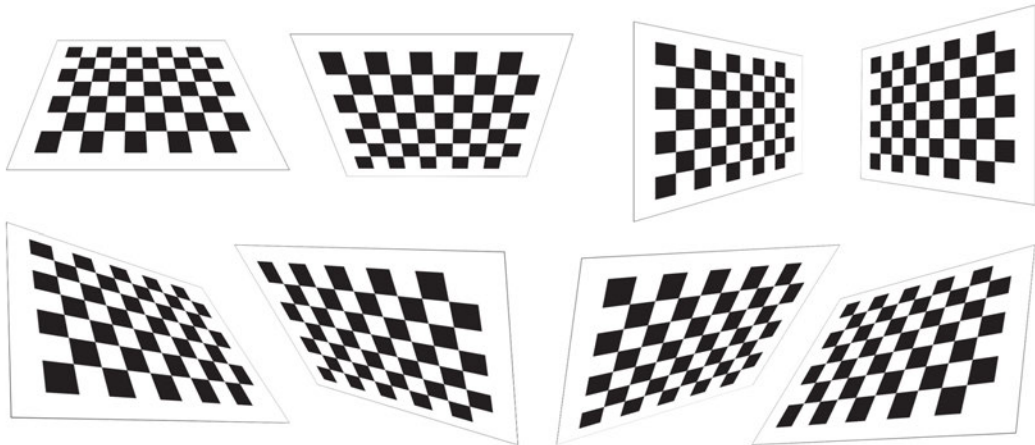
### 3.3 Hinweise für Aufnahmen von Kalibriertargets

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:

- Das Target 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.

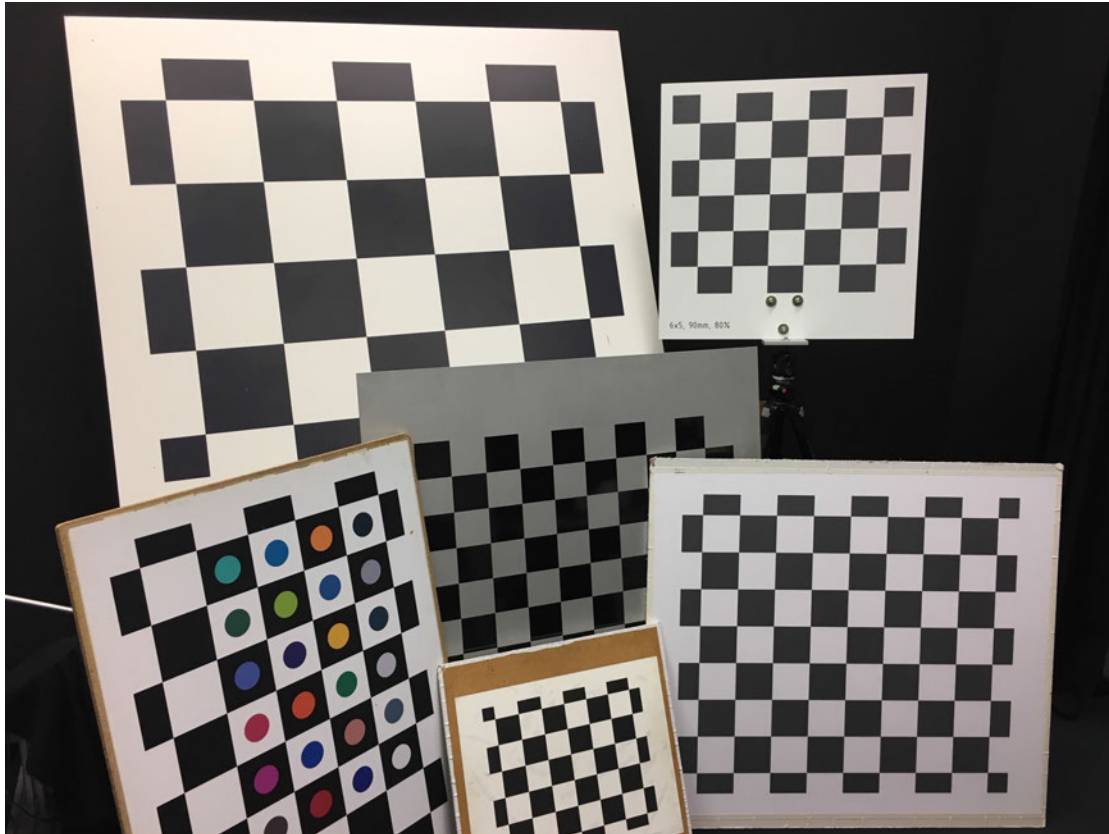
#### Bemerkung zur Wahl der geometrischen Form für das Kalibriertarget

Die meisten Targets verwenden entweder Kreise oder Schachbrettmuster. Bei einem verzerrungsfreien Kameraobjektiv ist es irrelevant, welche Art von Merkmalen auf



**Abb. 3.2:** 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.

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 Objekteigenschaften stark verzeichnet wird, werden Kreise asymmetrisch verzerrt und die Koordinaten des ursprünglichen Zentrums sind nicht trivial zu bestimmen. Das wurde auch in [17] theoretisch und experimentell nachgewiesen. Daher wird für die Kalibrierung ein Schachbrettmuster anstelle eines Targets mit Kreisen empfohlen.



**Abb. 3.3:** Kalibriertargets für unterschiedliche Einsatzzwecke



---

## Benutzeroberfläche des 3D-EasyCalib

Das Hauptfenster des Programms ist in drei Bereiche aufgeteilt: auf der linken Seite sind die hinzugefügten Komponenten (Kameras, Koordinatensysteme) als Liste dargestellt (2). Mittig sind die zu den links gelisteten Komponenten entsprechenden Koordinatensysteme visualisiert (3). Rechts sind die Aktionen – die Kalibrierverfahren – gelistet (4). Über das Menü „File“ können Projekte gespeichert und geladen werden (1).

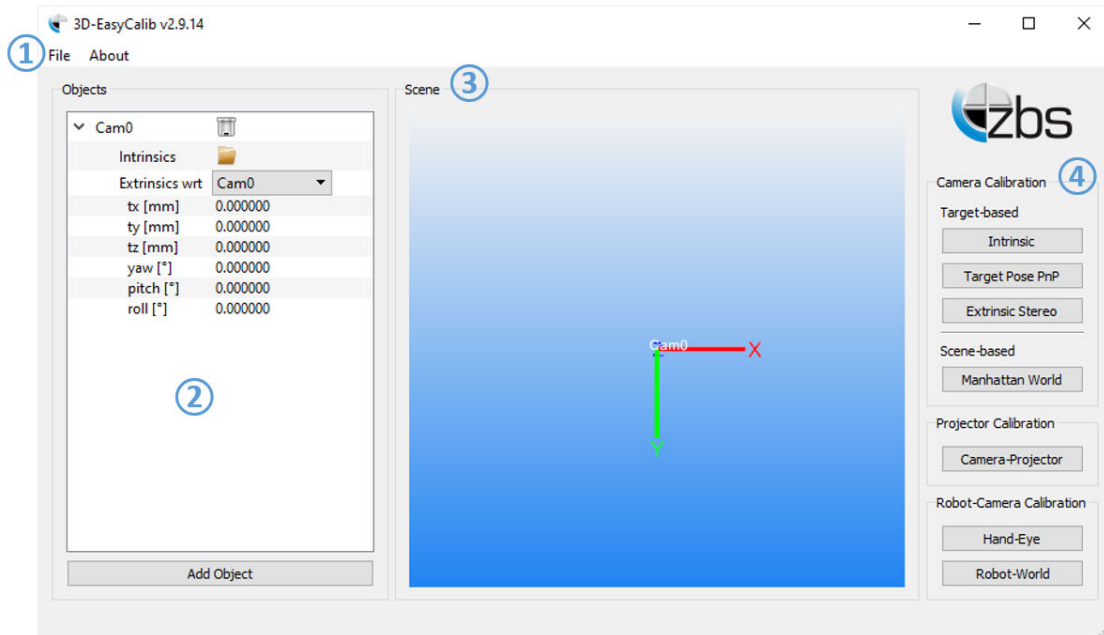
- ① Das File-Menü beinhaltet folgende projektbezogene Routinen:

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

**Load Project** lädt ein gespeichertes Projekt aus einer YML-Datei. Die Spezifikation des Formats ist im Abschnitt „[Aufbau der Projekt-Datei](#)“ beschrieben.

**Save Project** exportiert das Projekt als YML- und Json-Datei. Hier werden alle Kameras im „Objects“-Feld bezüglich des ausgewählten Koordinatensystems exportiert. Diese extrinsischen Daten transformieren die Koordinaten der jeweiligen Kamera in das ausgewählte Weltkoordinatensystem. Als Welt kann eine beliebige Kamera aus der „Objects“-Liste ausgewählt werden.

- ② Die Kameras können mit dem Button [Add Object] hinzugefügt werden, und die bereits hinzugefügten Kameras lassen sich mit dem Papierkorb-Button löschen. Die intrinsischen Daten für eine Kamera können mit dem Ordner-Button



**Abb. 4.1:** Hauptkomponenten der Benutzeroberfläche

geladen werden. Die Darstellung der intrinsischen bzw. extrinsischen Parameter ist in den Abschnitten 4.1ff. detailliert beschrieben.

- ③ **Szene.** In der Mitte des Hauptfensters werden die Komponenten symbolisch mit den Koordinatenachsen und einer Beschriftung dreidimensional dargestellt. Bedienung: linke Maustaste dreht die Ansicht, mittlere Maustaste verschiebt die Szene parallel zur Kamera, das Mausrad vergrößert / verkleinert (Zoom-In / Zoom-Out) die Ansicht. Die Taste **R** setzt die Zoom-Skalierung zurück.
- ④ Die Methoden für intrinsische und extrinsische Kalibrierung sind in drei Kategorien aufgeteilt: Kamera-, Projektor- und Roboter-Kamera-Kalibrierung. Die Kamerakalibrierung enthält targetbasierte Methoden zur intrinsischen Kamerakalibrierung [Intrinsic] (Abschnitt 5), zur Ermittlung der extrinsischen Parameter einer Kamera bzw. eines Targets [Target Pose PnP] und zur extrinsischen Kalibrierung eines Kamerapaares [Extrinsic Stereo] (Abschnitt 6). Die szenenbasierte extrinsische Kalibrierung ist in der Methode [Manhattan World] enthalten. Die Kamera-Projektor-Kalibrierung [Camera-Projector] (Abschnitt 7) ermittelt sowohl die intrinsischen Parameter eines Projektors wie auch dessen Lage und Orientierung bezüglich einer Kamera. Die Roboter-Kamera-Kalibrierung enthält zwei Methoden: die sogenannte Hand-Auge-Methode [Hand-

Eye] (Abschnitt 9) und die Tool-Flange- bzw. Roboter-Welt-Kalibrierung [Robot-World] (Abschnitt 8).

Jedes Verfahren wird in einem neuen Fenster geöffnet. Die kalibrierten Parameter werden automatisch in das Hauptfenster übertragen.

## 4.1 Aufbau der Projekt-Datei

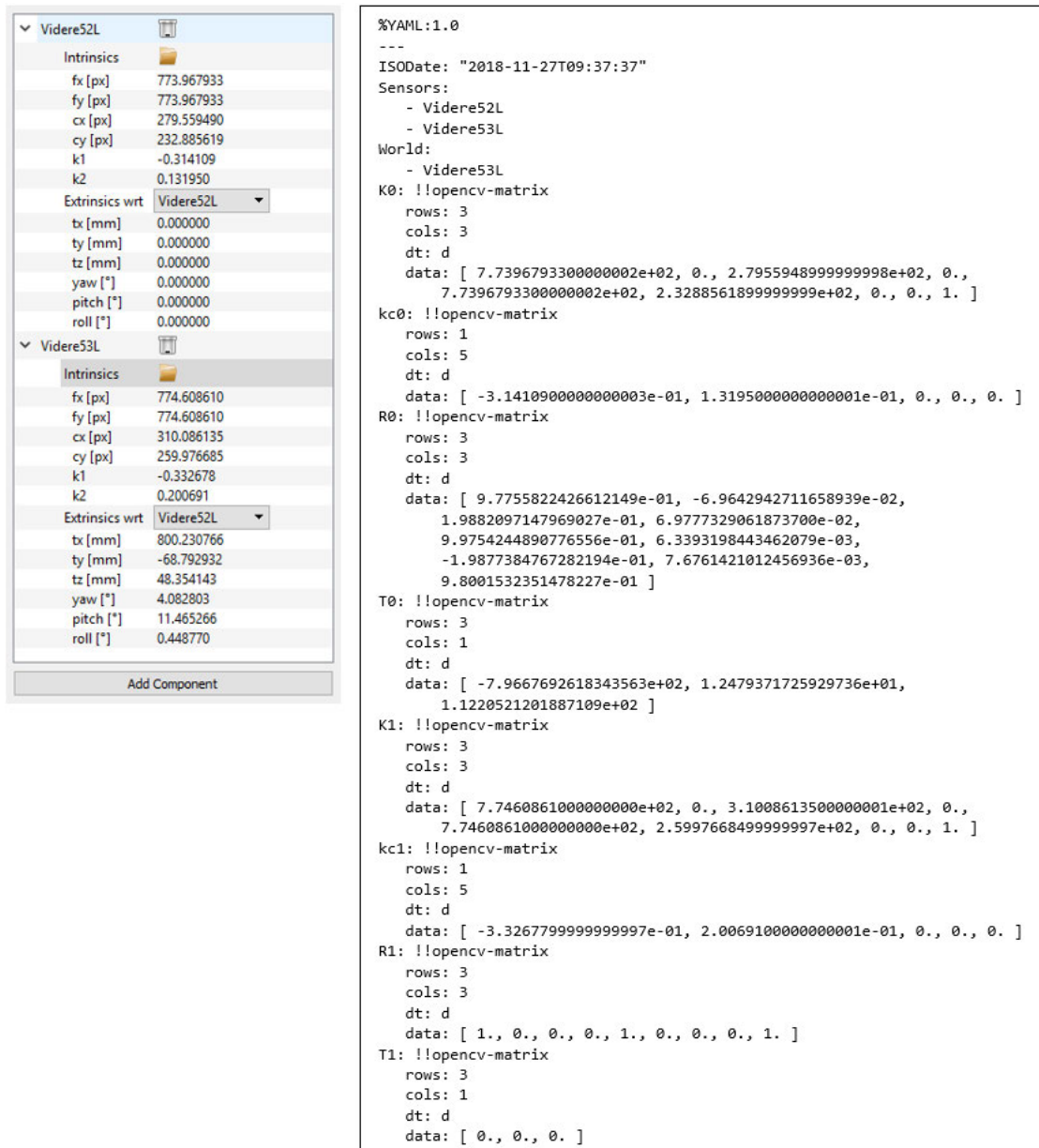
Die YAML-Datei (siehe Bild 4.2) enthält nach dem Header (%YAML:1.0) das Datum der gespeicherten Kalibrierung (ISODate). Danach kommen die Sensornamen (Sensors) und das Bezugsobjekt (World), zu dem alle aufgelisteten Komponenten extrinsisch kalibriert sind. Mit anderen Worten, alle Koordinatensysteme haben die gleiche Referenz – das Weltkoordinatensystem. Es folgen die intrinsischen ( $K_i, kc_i$ ) sowie die extrinsischen ( $R_i, T_i$ ) Parameter für alle Sensoren. Das  $i$  repräsentiert die Position des Sensors in der Sensorliste. Die Elemente der Kameramatrix  $K_i$  sind zeilenweise wie in Gleichung 2.2 gespeichert, und die Reihenfolge der Verzeichnungsparameter ist  $kc_i = [k_1, k_2, p_1, p_2, k_3]_i$  (siehe Abschnitt 2.2).  $R_i$  enthält die  $3 \times 3$ -Rotationsmatrix und  $T_i$  den Translationsvektor.

## 4.2 Darstellung der intrinsischen Parameter in 3D-EasyCalib

Die intrinsischen Parameter sind in der Objektliste (siehe Abb. 4.3, links) dargestellt. Aufgelistet sind die Parameter der Kameramatrix, die Bildweiten  $f_x, f_y$  und der Bildhauptpunkt ( $c_x, c_y$ ). Diese Parameter werden entweder durch eine in 3D-EasyCalib enthaltene Methode erzeugt oder aus einer YAML-Datei geladen. Das Dateiformat ist wie folgt definiert:

Die Kameramatrix  $K$  ist im Feld data des Objektes K0, K1 usw. in einer Zeile dargestellt. Die Reihenfolge der Verzeichnungsparameter im data-Feld kc0, kc1 usw. ist [ k1, k2, p1, p2, k3 ]. Als Dezimaltrennzeichen dient der Punkt; die Zahlen können auch in Exponentialform geschrieben werden. Immer wenn eine intrinsische Kalibrierung durchgeführt wird, entsteht eine Backup-Datei „Intr.yml“ mit dieser Syntax im gleichen Ordner wie die zugehörigen Bilddaten.

Die numerischen Werte in der GUI können durch einen Doppelklick beliebig verändert werden. Die Verzeichnungsparameter werden nur dann angezeigt, wenn sie nicht gleich Null sind.



The image shows the EasyCalib software interface on the left and the corresponding YAML project file content on the right.

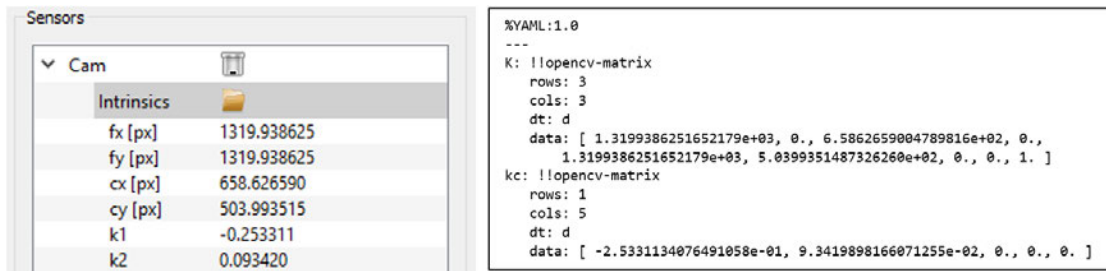
**Software Interface (Left):**

- Videre52L:**
  - Intrinsic parameters: fx [px] 773.967933, fy [px] 773.967933, cx [px] 279.559490, cy [px] 232.885619, k1 -0.314109, k2 0.131950.
  - Extrinsic parameters (wrt Videre52L): tx [mm] 0.000000, ty [mm] 0.000000, tz [mm] 0.000000, yaw [°] 0.000000, pitch [°] 0.000000, roll [°] 0.000000.
- Videre53L:**
  - Intrinsic parameters: fx [px] 774.608610, fy [px] 774.608610, cx [px] 310.086135, cy [px] 259.976685, k1 -0.332678, k2 0.200691.
  - Extrinsic parameters (wrt Videre52L): tx [mm] 800.230766, ty [mm] -68.792932, tz [mm] 48.354143, yaw [°] 4.082803, pitch [°] 11.465266, roll [°] 0.448770.

**YAML File Content (Right):**

```
%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 Projekt-Datei für zwei Kameras. Links ist die Darstellung im Programm und rechts der Inhalt der Projekt-Datei im YAML-Format.



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

### 4.3 Darstellung der extrinsischen Parameter in 3D-EasyCalib

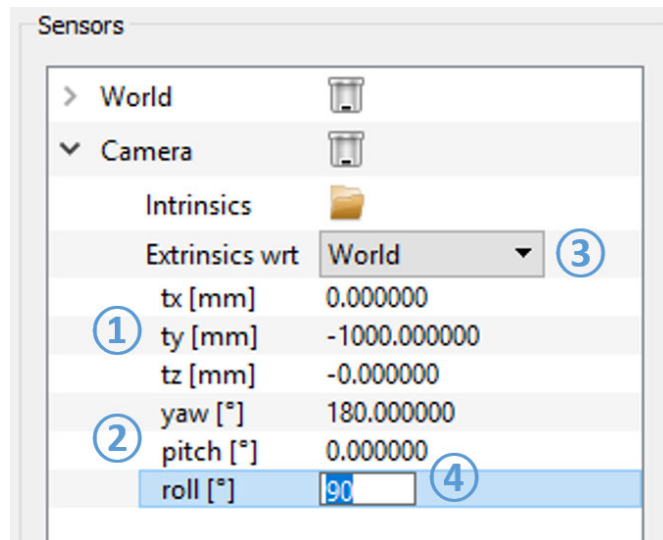
Das Kamerakoordinatensystem ist stets wie in Abb. 2.1 definiert: die Z-Achse ist mit der optischen Achse identisch und zeigt in das Bild, die X-Achse ist parallel zu den Bildzeilen und die Y-Achse parallel zu den Bildspalten ausgerichtet. Der Koordinatenursprung liegt im Abstand  $f$  von der Bildebene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

- ① Der Translationsvektor  $\mathbf{t}$  ist mit den Komponenten  $t_x$ ,  $t_y$  und  $t_z$  (in Millimeter) angegeben und die Rotation  $R$  als Gier-, Nick- und Rollwinkel (engl. yaw, pitch, roll).
- ② Die Eulerschen Winkel sind in der Konvention  $z \rightarrow y' \rightarrow x''$  zu verstehen: „Gieren“ ist die Rotation um die z-Achse des Referenz-Koordinatensystems (hier



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

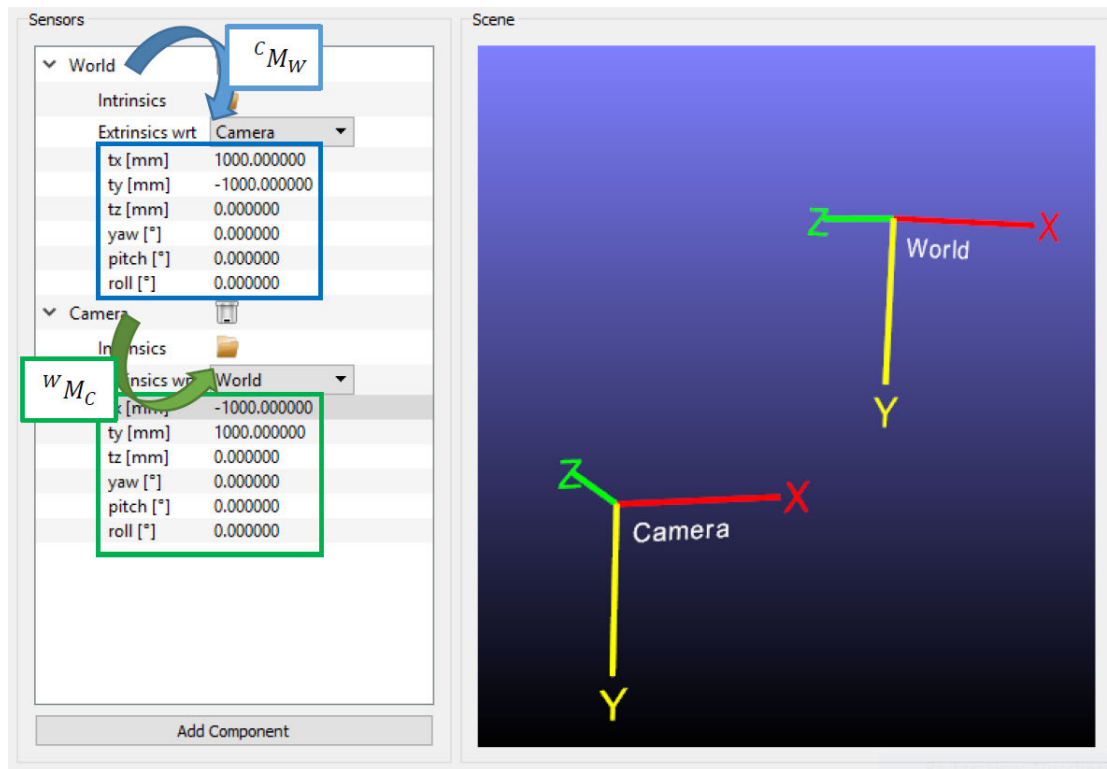
Weltkoordinatensystem) um den Winkel yaw in Winkelgrad gegen den Uhrzeigersinn. Analog ist „Nicken“ die Rotation um die y-Achse und „Rollen“ die Rotation um die x-Achse des Referenz-Koordinatensystems. Die Drehungen werden in dieser Reihenfolge ausgeführt, d.h. zuerst Rotation um die z-Achse, dann um die y-Achse und schließlich um die x-Achse.

- ③ Das Referenz-Koordinatensystem kann in der Combo-Box ausgewählt werden.
- ④ Die Werte der extrinsischen Parameter können mit Doppelklick auf das entsprechende Feld und Tastatur-Eingabe verändert werden.

## 4.4 Beispiel einer Kamera-zu-Welt-Transformation

Anhand des folgenden Beispiels soll sowohl die Transformation veranschaulicht als auch die Darstellung eines dreidimensionalen Punktes in verschiedenen Koordinatensystemen hergeleitet werden.

In 3D-EasyCalib ist der Bezug einer Kamera zu den anderen Kameras (bzw. zu einer Welt) aus Abb. 4.5 abzulesen: Die extrinsischen Parameter der **Camera** sind bezüglich (engl. Abkürzung wrt = with respect to) der **World** aufgelistet. Wenn die Punkte im Kamerakoordinatensystem erfasst wurden, können sie mit den extrinsischen Parametern in das gewählte Weltkoordinatensystem transformiert werden.



**Abb. 4.5:** Die Extrinsischen Parameter der **Camera** sind bezüglich (engl. Abkürzung wrt = with respect to) der **World** aufgelistet.

Um die Welt-zu-Kamera- bzw. Kamera-zu-Welt-Transformation für einen Punkt zu verdeutlichen soll die folgende Notation verwendet werden. Sei  $R$  als  ${}^C R_W$  und  $\mathbf{t}$  als  ${}^C \mathbf{t}_W$  geschrieben. Hier sind die Indizes als “von W zu C” zu lesen (siehe auch Abb. 4.5). Die Transformation

$${}^C M_W = \begin{bmatrix} {}^C R_W & {}^C \mathbf{t}_W \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (4.1)$$

transformiert den im Weltkoordinatensystem definierten Punkt  ${}^W \mathbf{P} = (X, Y, Z, 1)^T$  in das Kamerakoordinatensystem:

$${}^C \mathbf{P} = {}^C M_W \cdot {}^W \mathbf{P} \quad (4.2)$$

**Bemerkung** Die Notation ist wegen der folgenden, leicht zu merkenden, Regel gewählt: M transformiert den Punkt “von W zu C” und die Punkte links bzw. rechts von einer Transformation M müssen den gleichen tief- bzw. hochgestellten Index (hier C oder W) haben:

$${}^C \mathbf{P} = {}^C M_W \cdot {}^W \mathbf{P} \quad (4.3)$$

Diese Schreibweise ist auch bei einer Kettentransformation über mehrere Koordinatensysteme hilfreich: zwei aneinandergereihte Transformationen müssen den gleichen tief- bzw. hochgestellten Index haben (hier O):

$${}^C\mathbf{P} = {}^C M_O \cdot {}^O M_W \cdot {}^W \mathbf{P} \quad (4.4)$$

## 4.5 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 image format - \*.pbm, \*.pgm, \*.ppm
- Sun rasters - \*.sr, \*.ras
- TIFF files - \*.tiff, \*.tif

Falls im Dialog „Bild Laden“, das gewünschte Bildformat nicht gelistet ist, sollte „All files“ in der Combo-Box gewählt werden.

**Hinweis.** Es wird empfohlen, die aufgenommenen Bilder in einem verlustfrei komprimierten Bildformat wie Portable Network Graphics (PNG) zu speichern.

## Intrinsische Kamerakalibrierung

Ziel einer intrinsischen Kalibrierung ist die Bestimmung der inneren Parameter einer Kamera gemäß Gleichung<sup>1</sup>  $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$ . Von Interesse sind die Koeffizienten der Kameramatrix  $K$  sowie die Verzeichnungsparameter. Voraussetzung für die Kalibrierung ist die Aufnahme eines Kalibriertargets in unterschiedlichen Posen.

Die Kameramatrix  $K$  wird mit Aufnahmen eines geometrischen Kalibriertargets, dessen bekannte Welt-Punkte  $\mathbf{X}_i$  und dazu korrespondierende Bildpunkte  $\mathbf{x}_i$ , durch Lösung von  $\mathbf{x} = K[R|\mathbf{t}]\mathbf{X}$  bestimmt. In der „Community“ hat sich die Methode von Zhang [23] durchgesetzt, insbesondere wegen frei verfügbarer Implementierungen, wie der einfach bedienbaren Matlab-Umsetzung von Bouguet [2] und dessen Portierung in der OpenCV [3]. Nach diesem Ansatz werden im ersten Schritt die Targetposen mit der Homographie zwischen der Target- und Bildebene geschätzt. Danach wird aus  $n$  Aufnahmen des Kalibriertargets (vgl. Abb. 5.1) und  $m$  bekannten Koordinaten der Kalibrierpunkte  $\mathbf{X}_j$  auf dem Target sowie deren Abbildung  $\mathbf{x}_{ij}$  auf der Bildebene

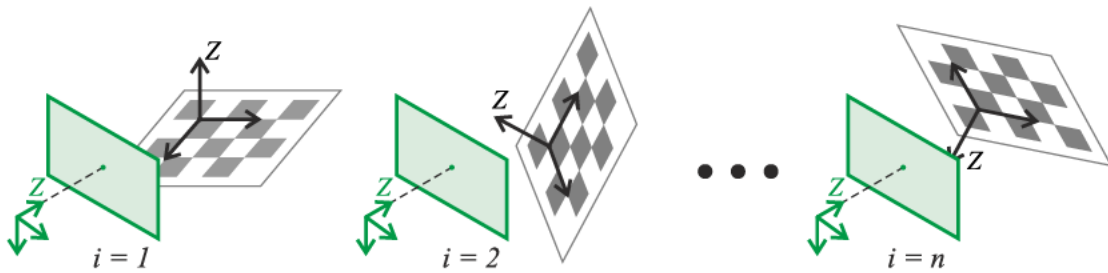
$$\begin{aligned}\mathbf{x}_{1j} &= K[R_1|\mathbf{t}_1]\mathbf{X}_j, \\ \mathbf{x}_{2j} &= K[R_2|\mathbf{t}_2]\mathbf{X}_j, \\ &\dots \\ \mathbf{x}_{nj} &= K[R_n|\mathbf{t}_n]\mathbf{X}_j,\end{aligned}\tag{5.1}$$

das Funktional

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

minimiert.

<sup>1</sup>Die Verzerrung wird in dieser Gleichung vernachlässigt. Sie steht logisch zwischen  $K$  und  $[R|\mathbf{t}]$ .



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

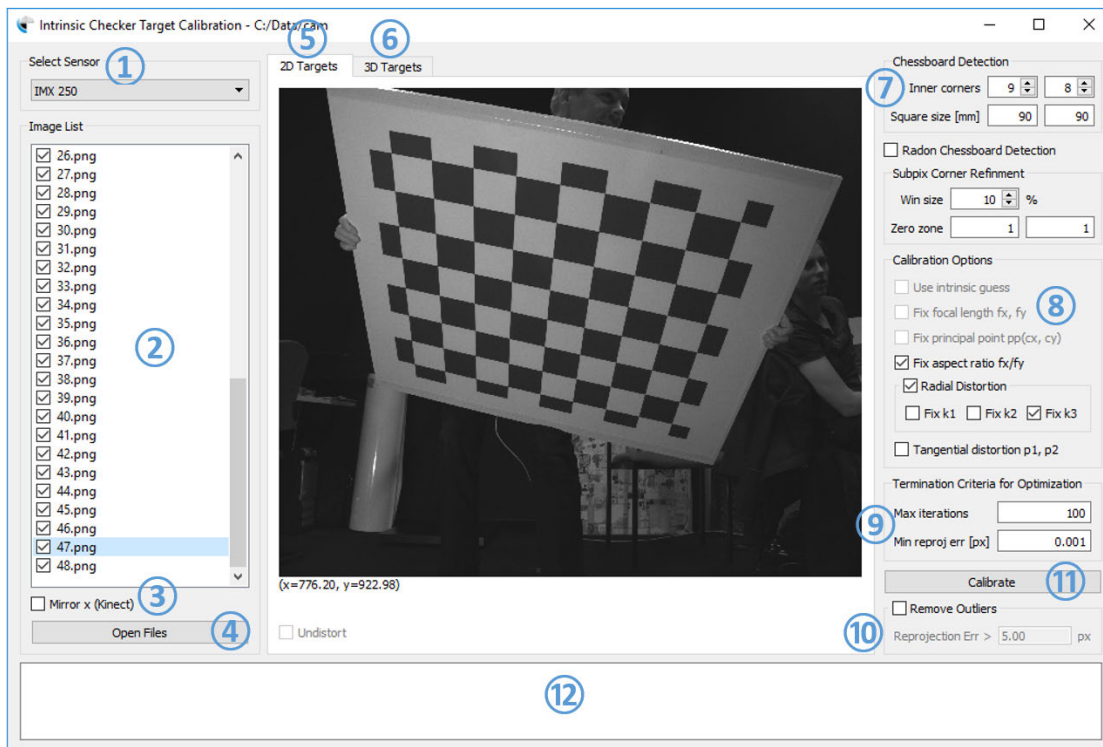
## 5.1 Reprojektionsfehler

Die in der Gleichung (5.2) enthaltenen euklidischen Abstände zwischen den nach dem Modell projizierten Kalibrierpunkten  $\mathbf{x}(K, R_i, \mathbf{t}_i, \mathbf{X}_j)$  und den im Bild vermessenen Punkten  $\mathbf{x}_{ij}$  werden *Reprojektionsfehler* genannt. Diese Fehlermetrik, angegeben für jeden Kalibrierpunkt sowie als Gesamtfehler (RMS) für jede Kalibriertarget-Pose, ist ein Indiz für die Güte einer Kalibrierung und wird im Toolkit dementsprechend dargestellt, vgl. Abb. 5.4. Da bei realen Aufnahmen die Koordinaten der Passpunkte z.B. wegen Verdeckung, Schattenwurf, Kamerarauschen oder Bewegung während der Aufnahme unpräzise oder falsch detektiert werden können, werden potentielle Ausreißer farblich gekennzeichnet. Diese können auf dieser Grundlage manuell aus dem Kalibrierungs-Datensatz entfernt werden.

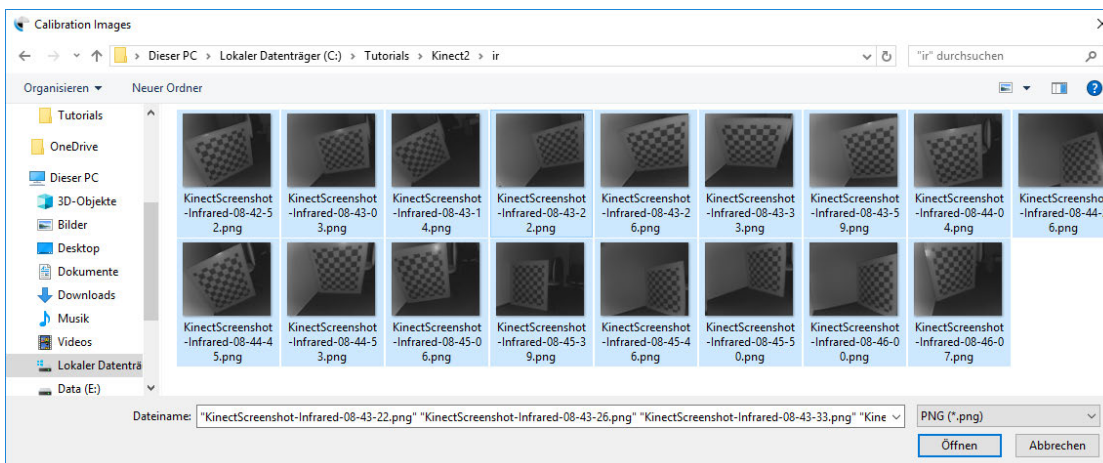
Die GUI für intrinsische Kamerakalibrierung (siehe Abb. 5.2) besteht aus folgenden Elementen:

## 5.2 Eingabe

- ① **Select Sensor:** Die zu kalibrierte Kamera soll in der Combo-Box ausgewählt werden. Hier stehen die im Hauptfenster (Kapitel 4) hinzugefügten Komponenten zur 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 **Mirror x (Kinect):** Diese Einstellung spiegelt die Bilder horizontal. Das ist für einige Kameras, wie z.B. Kinect2, die das Bild gespiegelt ausgeben, notwendig.



**Abb. 5.2:** Die Benutzeroberfläche beinhaltet die Eingabe (links), Einstellparameter (rechts), Visualisierung des Targets und der Kalibrierpunkte (mittig) und Protokollierung im Log-Fenster (unten).



**Abb. 5.3:** Beispielbilder für eine intrinsische Kalibrierung. Die Bilder werden durch Auswählen in das Programm geladen.

Wenn in solchen Fällen das Bild unverändert bleibt, ist es in einem linkshändigen Koordinatensystem definiert. Die extrinsische Kalibrierung der Kameras zueinander erzeugt bei unterschiedlich definierten Koordinatensystemen ungültige Parameter. 3D-EasyCalib verwendet stets rechtshändige Koordinatensysteme.

- ④ **Open Files (Bilder laden):** Öffnet den Dialog zum Laden von Bildern des Kalibriertargets. Die unterstützten Bildformate sind im Abschnitt 4.5 genannt.

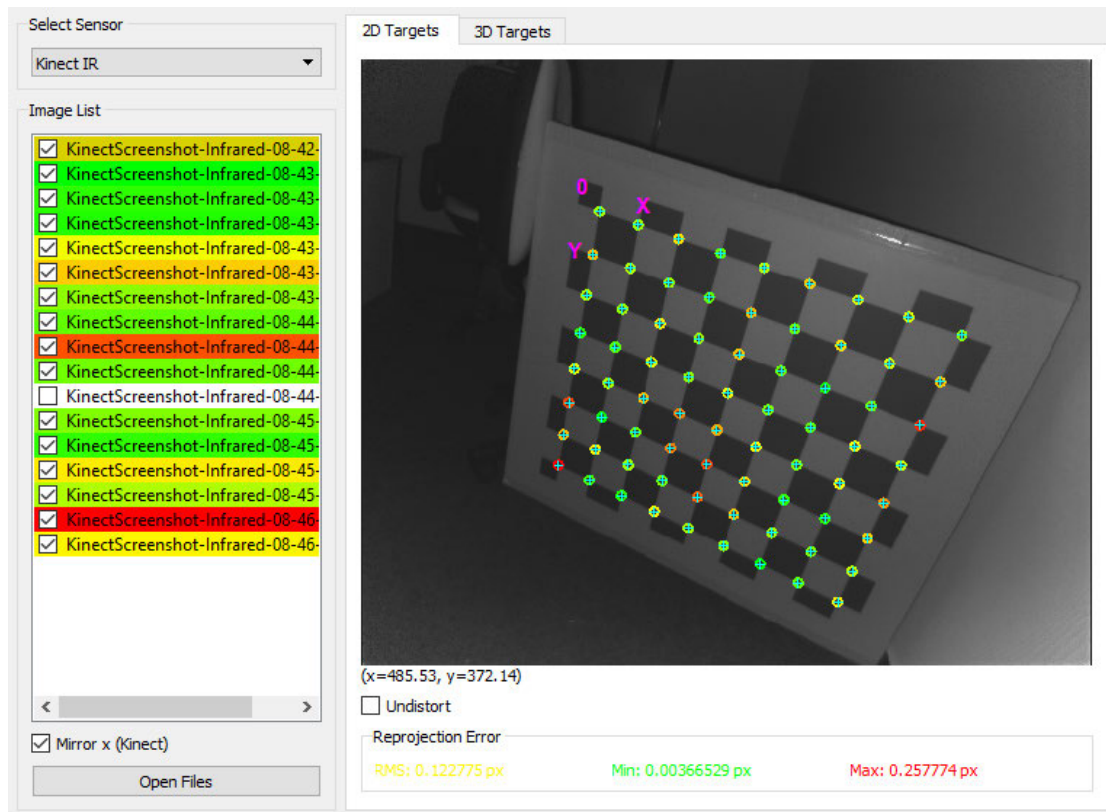
### 5.3 Visualisierung

Die dargestellten Bilder können mit dem Mousrad vergrößert / verkleinert und in der 3D-Ansicht mit der linken Maustaste gedreht werden.

- ⑤ **2D Targets:** In diesem Bereich wird das ausgewählte Bild angezeigt. Unter dem Bild werden die Bildkoordinaten des Mauszeigers dargestellt. Falls die Verzerrungsparameter vorhanden sind, kann das Bild mit [x] **undistort** entzerrt dargestellt werden. Nach der Kalibrierung werden die Reprojektionsfehler für das Target gelistet: RMS über alle Punkte auf dem Target (links) sowie minimaler (mittig) und maximaler (rechts) Fehler des Targets (siehe Abb. 5.4 unten).
- ⑥ **3D Targets:** Nach der Kalibrierung werden die Kalibriernormale und die Kamera dreidimensional dargestellt. Die Farbe des Targets entspricht dem Reprojektionsfehler bezüglich des min. und max. Fehlers über alle Kalibriertargets. Die X- und Y-Achse des Targets in der 3D-Ansicht sind durch dickere Linien hervorgehoben.

### 5.4 Einstellungen

- ⑦ **Chessboard Detection:** Die Parameter für die Erkennung des Kalibriernormals sind im Kapitel 3, Seite 15, beschrieben.
- ⑧ **Calibration Options:** Die Optionen mit „Fix“, wie z.B. **Fix focal length** bedeuten, dass die betreffenden Parameter nicht optimiert werden sollten – sie bleiben konstant (fixiert). Ist die Einstellung abgewählt, werden die betreffenden Parameter bei der Optimierung berücksichtigt.
  - **Use intrinsic guess:** Innere Kameraparameter als Initialwerte für die Optimierung verwenden. Diese Option ist nach einer bereits durchgeführten



**Abb. 5.4:** Links im Bild ist die Liste der Kalibrierbilder zu sehen. Diese sind farblich nach der Größe des Gesamt-Reprojektionsfehlers eines Targets von grün (min. Fehler) bis rot (max. Fehler) dargestellt. Rechts befindet sich das Bild einer ausgewählten Target-Pose mit eingezeichneten detektierten Kalibrierpunkten (Kreise) und nach dem Modell berechneten Punkten (Kreuze). Die Farbe der Kreise stellt den Reprojektionsfehler ([Abschnitt 5.1](#)) dar. Das Bild in der GUI kann mit dem Mausrad vergrößert/verkleinert werden.

Kalibrierung freigeschaltet. Sie ist ebenfalls freigegeben, wenn die Parameter für die ausgewählte Kamera im Hauptfenster geladen wurden.

- **Fix focal length  $f_x, f_y$ :** Die effektiven Bildweiten bleiben während der Optimierung unverändert (fixiert) und sind gleich den Initialwerten.
  - **Fix principal point  $pp(c_x, c_y)$ :** Wenn aktiviert, dann bleibt der Hauptpunkt während der Optimierung unverändert (gleich dem Initialwert).
  - **Fix aspect ratio  $f_x/f_y$ :** Wenn aktiviert, bleibt das Verhältnis der Bildweiten während der Optimierung fest. Siehe Hinweise im Abschnitt Bildweite  $f$  auf der Seite 6.
  - **Radial Distortion:** Wenn ausgewählt, wird die radiale Verzerrung geschätzt, sonst werden alle radialen Verzerrungsparameter gleich der Null gesetzt und fixiert. Die Theorie ist im [Abschnitt 2.2](#) beschrieben.
  - **Fix  $k_1$ , Fix  $k_2$ , Fix  $k_3$ :** Fixiert die einzelnen radialen Koeffizienten (angewählt) oder optimiert sie (abgewählt)
  - **Tangential distortion  $p_1, p_2$ :** Wenn aktiviert, werden die tangentialen Koeffizienten bei der Optimierung geschätzt, sonst nicht.
- ⑨ **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.
- ⑩ **Remove Outliers** Ausreißer-Beseitigung: Nachdem die Kalibrierung einmal durchgeführt wurde, können die einzelnen Kalibrierpunkte auf dem Target mit dem Kriterium „**Reprojection Err** >“ von der Optimierung ausgeschlossen werden. Die Ausreißer-Beseitigung ist in zwei Schritten realisiert: nach der Kalibrierung werden die Ausreißer, die diese Bedingung erfüllen, markiert und dann bei der erneuten Kalibrierung entfernt.
- ⑪ **Calibrate:** Das nichtlineare Minimierungsproblem (5.2) wird mit dem Levenberg-Marquard-Algorithmus gelöst. Dabei wird der Reprojektionsfehler (siehe [Abschnitt 5.1](#)) minimiert.

## 5.5 Ausgabe

Nach der Kalibrierung werden die berechneten intrinsischen Parameter in das Hauptfenster übertragen ([Abschnitt 4.2](#)), als „Intr.yml“ im Ordner der geladenen Kalibrierbilder gespeichert und im Log-Fenster ausgegeben.

```

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

```

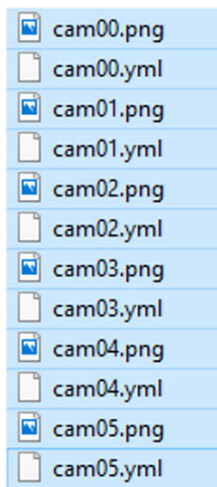
**Abb. 5.5:** Die Ereignisse während des Kalibriervorganges werden im Log-Fenster protokolliert. Eine rote Zahl unter **Decoding chessboards** bedeutet eine fehlgeschlagene Targeterkennung. Nachdem die Optimierung abgeschlossen ist, werden die intrinsischen Parameter ausgegeben. Die Abweichungen (+/-) entsprechen der dreifachen Standardabweichung  $3\sigma$ .

- 12 **Log-Fenster:** Dient zur Protokollierung der Ereignisse während des Kalibrierungsvorganges (siehe Abb. 5.5). Erfolgreich erkannte Targets werden mit einer grünen, die nicht erkannten mit einer roten und die abgewählten Targets mit einer grauen Zahl repräsentiert. Nach der Kalibrierung werden der RMS-Wert des Reprojektionsfehlers (rms reprojection error) sowie die optimierten intrinsischen Parameter ausgegeben. Die Abweichungen (+/-) entsprechen der dreifachen Standardabweichung  $3\sigma$ . Die intrinsischen Parameter werden in das Hauptfenster übertragen sowie in der Datei „Intr.yml“ gespeichert. Diese YAML-Datei kann im Hauptfenster verwendet werden, um die intrinsischen Parameter zu laden.

## 5.6 Verwendung eigener Passpunkte bzw. eines 3D-Normals

Wenn Aufnahmen eines Bildaufnahmeegerätes nicht mit den oben genannten Ecken-detektoren realisierbar sind, bzw. die Passpunkte mit einem eigenen Verfahren berechnet wurden, dann können diese in 3D-EasyCalib ebenfalls geladen und für die Kalibrierung verwendet werden.

Zusätzlich zu jedem Bild sollte noch eine Datei mit dem gleichen Präfix wie das Bild selbst und mit der Dateierdung „.yml“ vorhanden sein (siehe Abb. 5.6 links). Der Inhalt für eine beispielhafte YML-Datei ist in Abb. 5.6 rechts angegeben und besteht aus zwei Datensätzen, „corners“ und „obj“. In „corners“ sind die Bildpositionen der Kalibrierpunkte ( $\mathbf{x}$  in der Gleichung  $\mathbf{x} = K[R|t]X$ ) aufgelistet, wobei x- und y-Koordinaten abwechselnd zu schreiben sind. Der Datensatz „obj“ enthält die X, Y und Z-Koordinaten der entsprechenden Weltpunkte ( $X$  in der Gleichung  $\mathbf{x} = K[R|t]X$ ), angegeben in mm.



```
%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. 5.6:** Beispiel einer Bildfolge mit bereits erkannten Kalibrierpunkten. Im Datensatz „corners“ sind die Passpunkte in Bildkoordinaten  $(x, y)$  enthalten, im Datensatz „obj“ die Koordinaten auf dem Kalibriertarget  $(X, Y, Z)$ .

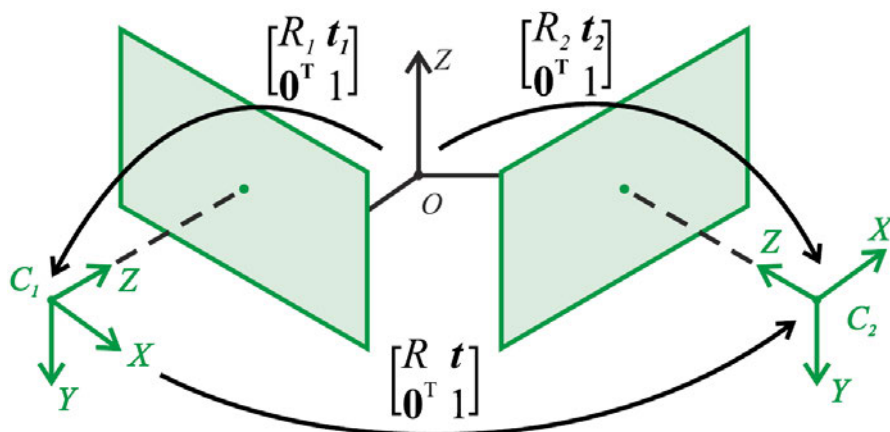
Die ersten drei korrespondierenden Punktepaare sind farblich markiert (Cyan, Magenta, Orange). Die „rows“ enthalten die Anzahl der Passpunkte (hier 30) und „dt“ den Datentyp, nämlich zweidimensionaler (2f) und dreidimensionaler (3f) Punkt mit Koordinaten als float-Zahlen.

Falls ein 3D-Target verwendet wird, also wenn die Z-Koordinate in den „obj“-Punkten nicht null ist, dann müssen im Hauptfenster Anfangswerte für die Bildweite  $f$  und den Bildhauptpunkt  $(c_x, c_y)$  eingetragen werden. Als Bildweite könnte beispielsweise die Brennweite des Objektivs dividiert durch die Größe eines Pixels auf dem Kameraschip verwendet werden. Als Bildhauptpunkt kann die Bildmitte eingetragen werden. Zusätzlich muss die Option **Use intrinsic guess** in den **Calibration Options** ausgewählt werden.

## Extrinsische Stereo-Kalibrierung

Ziel der extrinsischen Stereo-Kalibrierung ist die Bestimmung der relativen Lage  $\mathbf{t}$  und Orientierung  $R$  der Kamera  $C_1$  bezüglich der Kamera  $C_2$ , wie in der Abb. 6.1 dargestellt.

Analog zur Berechnung der Targetposen für die intrinsische Kamerakalibrierung (5, Seite 29) wird das Target hier simultan von beiden Kameras erfasst. Die Pose des Targets aus der Sicht der beiden Kameras wird berechnet und daraus die relative Pose der Kameras zueinander hergeleitet (siehe Abschnitt 6.1). Die resultierenden extrinsischen Parameter  $R$  und  $\mathbf{t}$  transformieren Punkte vom Koordinatensystem der ersten



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

Kamera ( $X_1$ ) in das Koordinatensystem der zweiten Kamera ( $X_2$ ).

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

Die nichtlineare Optimierung ist der intrinsischen Kamerakalibrierung ähnlich und die Parameter  $R$  und  $\mathbf{t}$  werden so optimiert, dass die Summe der Reprojektionsfehler in beiden Bildern minimal wird.

#### Voraussetzungen für eine Sterokalibrierung:

- Die 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.

**Bemerkung.** Idealerweise wurden die Kameras bereits intrinsisch kalibriert, so dass nur die Rotation und Translation bei diesem Vorgang bestimmt werden. Sind die intrinsischen Parameter unbekannt, dann werden sie vor der Stereokalibrierung geschätzt und mitoptimiert.

## 6.1 Extrinsische Parameter $R$ und $\mathbf{t}$

Wird ein Weltpunkt  $\mathbf{X}$  auf dem Target im ersten und dann in zweitem Kamerakoordinatensystem betrachtet, lauten die transformierten Punkte (siehe [Abbildung 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 die Gleichung (6.2) nach  $\mathbf{X}$  aufgelöst und in (6.3) eingesetzt wird:

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

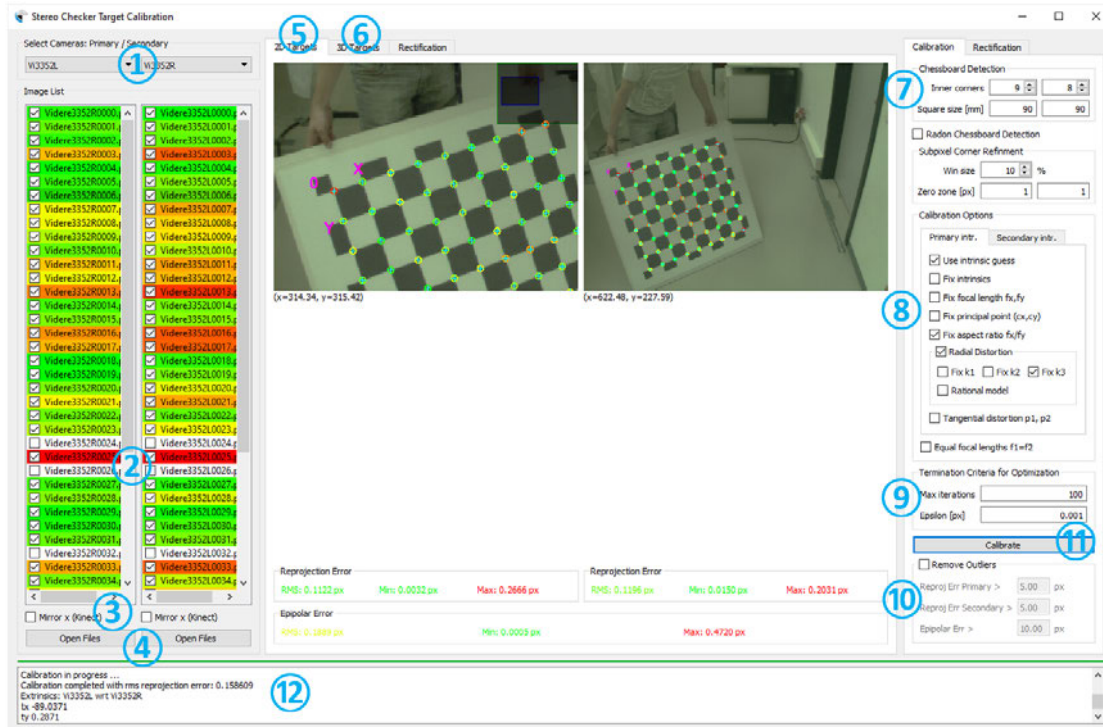


Abb. 6.2: Benutzeroberfläche für die Kalibrierung von zwei Kameras.

Die gesuchten Parameter, Rotation  $R$  und Translation  $\mathbf{t}_1 = \overrightarrow{C_1 C_2}$  lauten somit:

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

## 6.2 Eingabe

Die GUI für die extrinsische Kamerakalibrierung (siehe Abb. 6.2) ähnelt der GUI für die intrinsische Kalibrierung, wurde jedoch um Input und Visualisierung für die zweite Kamera erweitert. Die Einstellparameter für die Optimierung erweitern sich um die intrinsischen Parameter der zweiten Kamera. Die GUI besteht aus folgenden Elementen:

- 1 **Select Sensors:** Die zu kalibrierten Kameras können in der Combo-Box ausgewählt werden. Hier stehen die im Hauptfenster hinzugefügten Komponenten zur Auswahl.

- ② **Image List:** In der Bildliste sind die Dateinamen der geladenen Bilder enthalten. Nach Auswahl eines Bildes in der Liste 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.

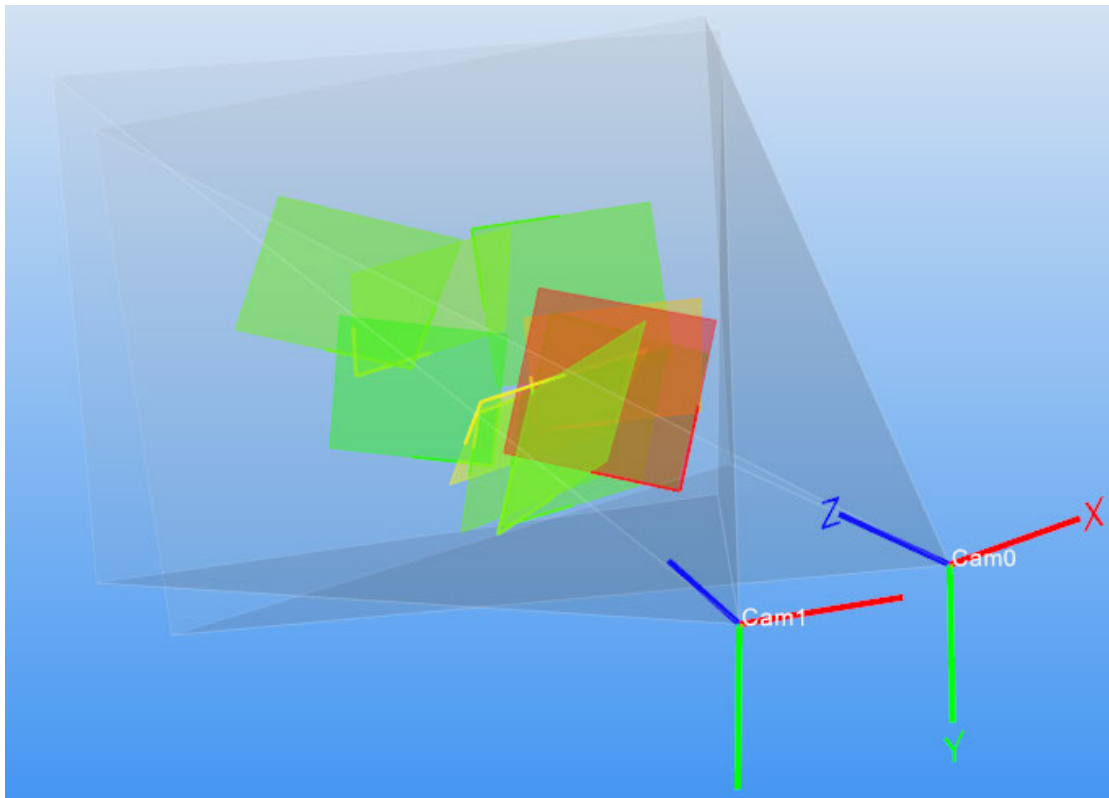
- ③ **Mirror x (Kinect):** Diese Einstellung spiegelt die Eingabebilder horizontal. Das ist für einige Kameras, wie z. B. Kinect notwendig. Wenn das Bild gespiegelt ist, ist es in einem linkshändigen Koordinatensystem definiert. Eine extrinsische Kalibrierung der Kameras zueinander erzeugt ungültige Parameter, wenn die Koordinatensysteme unterschiedlich definiert sind. Das 3D-EasyCalib verwendet stets rechtshändige Koordinatensysteme.
- ④ **[Open Files] Bilder laden:** 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 4.5](#) gelistet.

## 6.3 Visualisierung

- ⑤ **2D Targets:** In diesem Bereich wird das ausgewählte Bildpaar gezeigt. Unter dem jeweiligen Bild sind die Bildkoordinaten des Mauszeigers dargestellt. Nach der Kalibrierung werden unter dem jeweiligen Bild (siehe [Abb. 6.2](#)) die Reprojektionsfehler (Reprojection Error) für das Target gelistet: RMS über alle Ecken auf dem Target (links) sowie minimaler (mittig) und maximaler (rechts) Fehler 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 der Orientierung der Kameras zueinander dar.

- ⑥ **3D Targets:** Nach der Kalibrierung werden die Kalibriernormale und die Kameras dreidimensional dargestellt (siehe [Abb. 6.3](#)). Die Farbe des Targets entspricht der Größe des Reprojektionsfehlers, gemittelt über die beiden Kameras.



**Abb. 6.3:** 3D-Darstellung der zwei Kameras und der Kalibriertargets. Grün bedeutet einen kleinen und rot einen großen Fehler.

Die X- und Y-Achse des Targets in der 3D-Ansicht sind mit einer dickeren Linie verdeutlicht.

## 6.4 Einstellungen

- ⑦ **Chessboard Detection:** Die Parameter für die Erkennung des Kalibrierungsnormals sind im Kapitel 3, Seite 15, beschrieben.
- ⑧ **Calibration Options:** Die Einstellungen sind für die primäre (**Primary Intr.**) und sekundäre Kamera (**Secondary Intr.**) getrennt und können unabhängig voneinander parametrisiert werden.

Für die Parameteroptimierung sind folgende Einstellungen verfügbar. Die Optionen mit **Fix**, wie z.B. **Fix focal length** bedeuten, dass die betreffenden Parameter nicht optimiert werden – sie bleiben konstant (fixiert). Ist die Einstellung

The image shows two panels from a software interface for stereo calibration. The left panel, titled 'Calibration Options', has two tabs: 'Primary intr.' and 'Secondary intr.'. It contains several checkboxes: 'Use intrinsic guess' (checked), 'Fix intrinsics' (unchecked), 'Fix focal length  $f_x, f_y$ ' (unchecked), 'Fix principal point  $(c_x, c_y)$ ' (unchecked), 'Fix aspect ratio  $f_x/f_y$ ' (checked), 'Radial Distortion' (checked), 'Fix  $k_1$ ' (unchecked), 'Fix  $k_2$ ' (unchecked), 'Fix  $k_3$ ' (checked), 'Rational model' (unchecked), 'Tangential distortion  $p_1, p_2$ ' (unchecked), and 'Equal focal lengths  $f_1=f_2$ ' (unchecked). The right panel, titled 'Termination Criteria for Optimization', has input fields for 'Max iterations' (100) and 'Epsilon [px]' (0.001). Below these is a 'Calibrate' button and a 'Remove Outliers' checkbox. Under 'Remove Outliers', there are three rows: 'Reproj Err Primary >' with a value of 5.00 px, 'Reproj Err Secondary >' with a value of 5.00 px, and 'Epipolar Err >' with a value of 5.00 px.

**Abb. 6.4:** Einstellparameter für die Stereo-Kalibrierung

abgewählt, werden die zutreffenden Parameter bei der Optimierung berücksichtigt.

- **Use intrinsic guess:** Bekannte innere Kameraparameter als Initialwerte für die Optimierung verwenden. Diese Option ist nach einer bereits durchgeführten Kalibrierung, oder wenn die Parameter für die entsprechende Kamera im Hauptfenster geladen wurden, freigeschaltet. Wenn diese Option nicht aktiviert ist, wird für die betreffende Kamera zuerst die intrinsische Kalibrierung (Kapitel 5, Seite 29) durchgeführt.
- **Fix intrinsics:** Alle intrinsischen Parameter bleiben unverändert und sind gleich den Initialwerten. Diese Option ist nur dann verfügbar, wenn **Use intrinsic guess** aktiviert ist. Wenn die Kameras präzise intrinsisch kalibriert sind, sollte diese Option gewählt werden.
- **Fix focal length  $f_x, f_y$ :** Die effektiven Bildweiten bleiben während der Optimierung unverändert (fixiert) und sind gleich den Initialwerten.
- **Fix principal point  $(c_x, c_y)$ :** Wenn aktiviert, bleibt der Hauptpunkt während der Optimierung unverändert (gleich dem Initialwert).
- **Fix aspect ratio  $f_x/f_y$ :** Wenn aktiviert, bleibt das Verhältnis der Bildweiten während der Optimierung fest.

- **Same focal lengths  $f_1=f_2$ :** Nur dann zu verwenden, wenn die beiden Kameras und Objektive identisch sind.
  - **Radial Distortion:** Wenn ausgewählt, wird die radiale Verzerrung geschätzt, sonst werden die radialen Verzerrungsparameter alle gleich Null gesetzt und fixiert. Die Theorie und Vorschläge zur Wahl der Parameter sind im [Abschnitt 2.2](#) beschrieben.
  - **Fix  $k_1$ , Fix  $k_2$ , Fix  $k_3$ :** fixiert die einzelnen radialen Koeffizienten (ausgewählt) oder optimiert sie (abgewählt).
  - **Tangential distortion  $p_1, p_2$ :** Wenn aktiviert, werden die tangentialen Koeffizienten bei der Optimierung geschätzt, sonst nicht. In der Regel ist diese Option für moderne Digitalkameras nicht zu empfehlen.
- 9 **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.
- 10 **Remove Outliers** (Ausreißer-Beseitigung): Nachdem die Kalibrierung einmal durchgeführt wurde, können einzelne Kalibrierpunkte auf dem Target mit dem Kriterium **Reprojection Err Primary** > und **Reprojection Err Secondary** > von der Optimierung ausgeschlossen werden. „Primary“ und „Secondary“ bezeichnen die im Bereich **Select Sensors** gelistete erste und zweite Kamera. Der Wert unter **Epipolar Err** > bewirkt die Entfernung der Punktpaare, die das Kriterium überschreiten. Die Ausreißer-Beseitigung ist in zwei Schritten umgesetzt: Nach der Kalibrierung werden die Ausreißer, die diese Bedingung erfüllen, markiert und bei der erneuten Kalibrierung entfernt.
- 11 **Calibrate** führt die Optimierung aus. Die Parameter werden so optimiert, dass die Summe der Reprojektionsfehler in korrespondierenden Bildpaaren minimal wird.

## 6.5 Ausgabe

Nach der Kalibrierung werden die berechneten extrinsischen (6.5) und gegebenenfalls intrinsischen Parameter in das Hauptfenster übertragen ([Abschnitt 4.2](#)), als **stereo.yml** im Ordner der geladenen Kalibrierbilder gespeichert und im Log-Fenster ausgegeben.



**Abb. 6.5:** Ergebnis der Bildentzerrung und Rektifizierung mit eingezeichneten Epipolarlinien (unten) der Eingabebilder (oben).

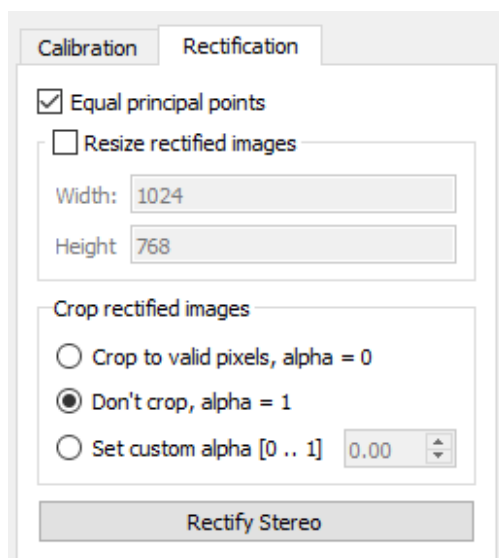
- 12 **Log-Fenster:** Dient zur Protokollierung der Ereignisse während des Kalibriervorganges. Erfolgreich erkannte Targets werden mit einer grünen, nicht erkannte mit einer roten und die abgewählten Targets durch eine graue Zahl repräsentiert. Nach der Kalibrierung werden der RMS des Reprojektionsfehlers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

**Bemerkung:** Die intrinsischen Parameter inklusive Verzerrungsparameter können unabhängig von den anderen Kamera optimiert werden. Das ist bei einem Kamera-paar mit unterschiedlichen Optik, Bildauflösung oder Modalität (z. B. Wärmebild- und visuelle Kamera) erforderlich. Diese Funktionalität ist bei anderen bekannten Programmen für die Stereokalibrierung meistens nicht gegeben.

## 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 (Spalten) eingeschränkt 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.



Originalbilder der Kameras in den rektifizierten Bildern erhalten bleiben (siehe Beispiel [Abbildung 6.5](#) 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).

### Ausgabe

Der Button [Rectify Stereo] führt die Rektifizierung durch und erzeugt zwei Ausgabedateien im gleichen Verzeichnis, in dem die Eingabebilder gespeichert sind: „recti-

**Equal principal points.** Setzt die beiden Bildhauptpunkte der rektifizierten Bilder auf gemeinsamen Punkt. Das vereinfacht die 3D-Berechnung aus dem Disparitätsbild, zum Beispiel mit der Funktion **reprojectImageTo3D** aus OpenCV.

**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

fication.yml“ und eine 3D-EasyCalib-Projektdatei. Die Projektdatei enthält die nach der Rektifizierung veränderten intrinsischen und extrinsischen Parameter. Die Datei „rectification.yml“ enthält die Matrizen  $R_0$ ,  $R_1$ ,  $P_0$ ,  $P_1$  und  $Q$ , die zur Berechnung der Bildkorrektur bzw. zur 3D-Triangulation verwendet werden.

- $R_0$ ,  $R_1$ : Rotationsmatrizen für die erste bzw. zweite Kamera. Diese Rotationen drehen die jeweilige Kamera so, dass die beiden Kameras parallel ausgerichtet sind.
- $P_0$ ,  $P_1$ : 3x4-Projektionsmatrizen für die rektifizierten Kameras.
- $Q$ : 4x4-Matrix, die eine Disparitätskarte in eine 3D-Punktwolke transformiert. Diese Matrix kann in die Funktion `reprojectImageTo3D()` von OpenCV eingegeben werden.

$$P_0 = \begin{bmatrix} f & 0 & c_{x1} & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.6)$$

$$P_1 = \begin{bmatrix} f & 0 & c_{x2} & t_x f \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.7)$$

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

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

---

## Kamera-Projektor-Kalibrierung

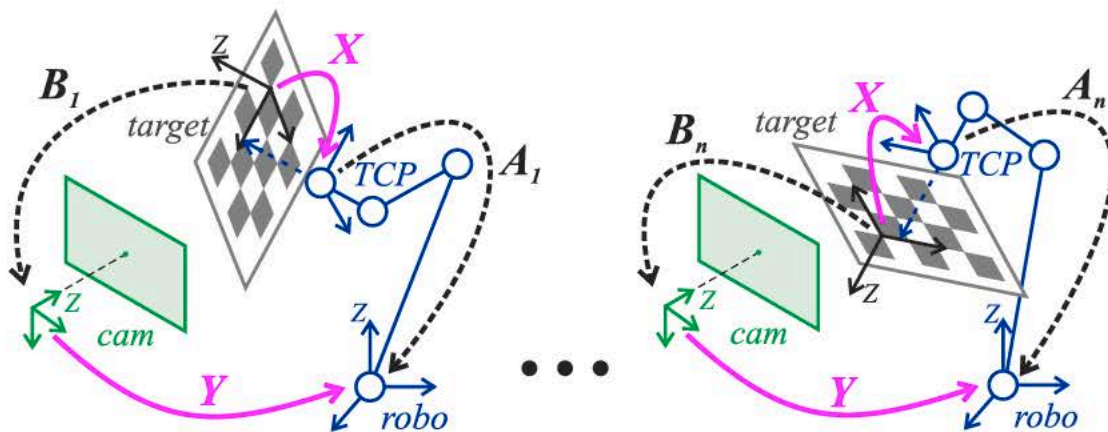
Bei einer Kamera wird die 3D-Szene auf die 2D-Bildebene projiziert während bei einem Projektor ein ebenes 2D-Muster in den 3D-Raum abgebildet wird. Daher werden Projektoren häufig als inverse Kameras bezeichnet und modelliert. Die der Abbildung zugrundeliegende mathematische Theorie ist für eine Kamera und eine inverse Kamera (Projektor) gleich. Folglich können dieselben Kalibrieralgorithmen verwendet werden. Die Herausforderung bei der Kalibrierung eines Projektors besteht lediglich darin, die Kalibrierpunkte in einem Projektor-Bild präzise zu bestimmen.

Dieses Problem lässt sich mit 3D-EasyCalib™ elegant mit einer Kamera lösen, die ...

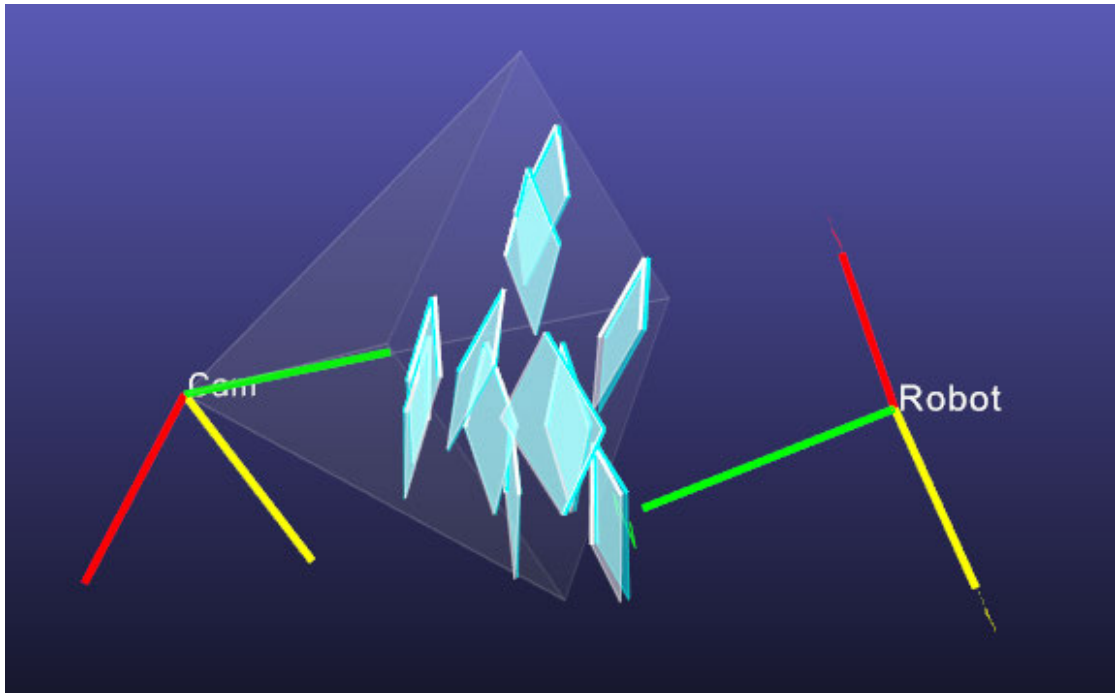


## Robot/World- und Tool/Flange-Kalibrierung

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, das Target mit dem Roboter in unterschiedlichen Positionen bewegt und mit der Kamera aufgenommen werden.



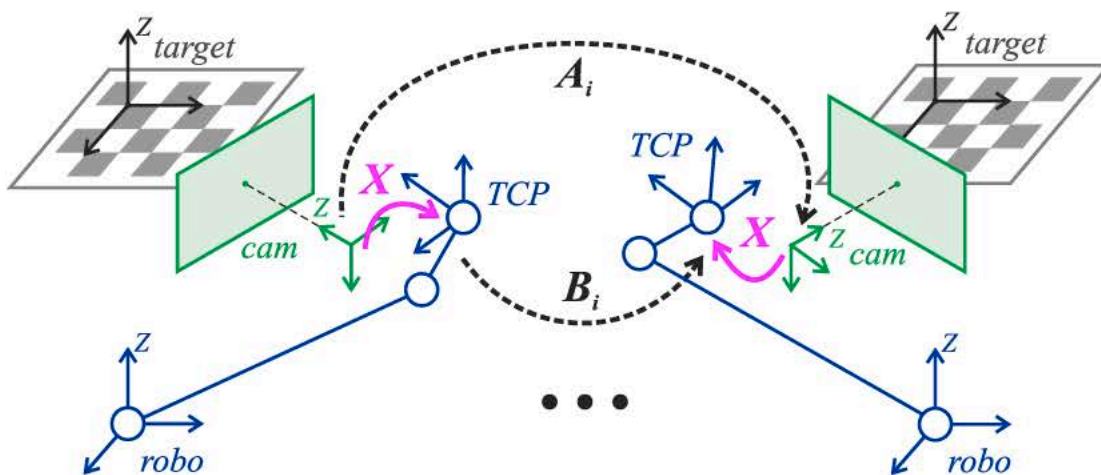
**Abb. 8.1:** Definition der Transformationen bei dem Roboter/Welt-, Tool/Flange-Problem.



**Fig. 8.6:** 3D-Ansicht des Roboters, der Kamerakoordinatensysteme und Kalibriertargets nach erfolgter Kalibrierung. Die cyanfarbenen Targets stellen die von der Kamera erfassten Zielpositionen dar, die in das Koordinatensystem des Roboters transformiert wurden ( $R_Y, t_Y$ ). Die weißen Ziele wurden zunächst mit den TCP-Daten und dann mit dem Offset zum TCP ( $R_X, t_X$ ) transformiert. Dies ist die grafische Darstellung des oben definierten 3D-Fehlers. Im Idealfall überschneiden sich die weißen und cyanfarbenen Rechtecke perfekt.

## Roboterwerkzeug zu Kamera Kalibrierung (Hand/Eye)

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. Die gesuchte Transformation wird



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





---

# Literaturverzeichnis

