

Kalibrierung und Implementierung eines Laserprojektors als Autopointer

Stefan Sollinger



MASTERARBEIT

KALIBRIERUNG UND IMPLEMENTIERUNG EINES LASERPROJEKTORS ALS AUTOPOINTER

Freigabe:

Die Bearbeiter:

Unterschriften

Stefan Sollinger

Betreuer:

Simon Kielhöfer

Der Abteilungsleiter

Der Institutsdirektor

Dieser Bericht enthält 67 Blatt, davon 37 Bilder und 0 Diagramme

Masterarbeit

Kalibrierung und Implementierung eines Laserprojektors als Autopointer

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Robotik und Mechatronik

Betreuung: Dipl.-Ing. S. Kielhöfer

Hochschule Regensburg

Fakultät Maschinenbau

Betreuung: Dr.-Ing. P. Gschwendner



eingereicht am 15. Juni 2012 von:

Stefan Sollinger

(Matrikelnummer: 2649241)

Abstract

Die vorliegende Arbeit beschreibt die Kalibrierung und Implementierung eines Laserprojektors als Autopointer. Ziel ist es, mit Hilfe eines Pico Laserprojektors herkömmliche Augmented Reality Methoden durch Projektion in die reale Welt zu erweitern.

Es soll ein "Autopointer" implementiert werden, der Informationen auf bekannte Oberflächen projiziert. Dabei muss ständig die 6D-Lage des Autopointers gemessen werden. Die Projektion wird entgegen der erfassten Bewegung nachgeführt, so dass das Bild ortsfest auf dem Objekt verbleibt.

Das System soll u.a. im medizintechnischen Bereich Verwendung finden, z.B. um präoperativ geplante Einstiche anzuzeigen.

Um die Verzeichnungserscheinungen des Laserprojektors auszugleichen, wurden verschiedenen Projektionen mit Hilfe einer kalibrierten Messkamera aufgenommen und unter Verwendung des homographischen Zusammenhangs ausgewertet. Diese Vorgehensweise hat sich als sehr schnell und effektiv erwiesen und ergab eine subpixelgenaue Beschreibung des Abbildungsverhaltens.

Die Untersuchung der Projektionseigenschaften des Autopointers liefert vielversprechende Ergebnisse.

Inhaltsverzeichnis

Abkürzungsverzeichnis	v
1 Einleitung	1
1.1 Motivation und Zielsetzung	1
1.2 Anwendungen	2
2 Theoretische Grundlagen der Kamerakalibrierung	3
2.1 Extrinsische Kalibrationsparameter	4
2.2 Intrinsische Kalibrationsparameterter	6
2.2.1 Das Lochkameramodell	6
2.2.2 Erweiterung des Lochkameramodells	7
2.2.3 Radiale Linsenverzeichnung	8
2.3 Homographie	10
3 Laserprojektorkalibrierung	12
3.1 Versuchsaufbau und Komponenten	13
3.1.1 Trackingsystem	16
3.1.2 Laserprojektor	17
3.1.3 Kamera	20
3.2 Intrinsische Messkamerakalibrierung	21
3.3 Extrinsische Messkamerakalibrierung unter Verwendung der Homographie	22
3.4 Projektion des pattern und Detektion der Eckpunkte	23
3.5 Abbildungseigenschaften des Laserprojektors	24
3.6 Beweis des konstanten Abbildungsverhaltens unter Verwen- dung der Pearson-Korrelation	28

3.7	Validierung der Messergebnisse	29
3.8	Extrinsische Kalibrierung des Laserprojektors	31
4	Software	32
4.1	OpenGL ES Grundlagen	32
4.2	Simulation des Autoprojektors	35
4.3	Software des Autoprojektors	36
4.3.1	Interfaces Trackingsystem und Beagleboard	36
4.3.2	Koordinatentransformation	37
4.3.3	Bilineare Interpolation	39
4.3.4	Window System, Interface und Devilspie	41
5	Genauigkeitsuntersuchung	43
5.1	Statische Genauigkeit	44
5.2	Dynamische Genauigkeit	48
6	Zusammenfassung	50
A	Anhang	52
A.1	Radiale Linsenverzeichnung	52
A.2	Ermittlung der Homographiematrix	53
A.3	Ergebnisse der Kalibrierung der Messkamera	55
A.4	Ergebnisse der Kalibrierung des Laserprojektors	55
A.5	ARTCom	56
A.6	Interface Trackingsystem und Beagleboard	57
A.7	Simulation des Autointers	58

Abbildungsverzeichnis

2.1	Koordinatentransformation	5
2.2	Lochkameramodell	6
2.3	Bildhauptpunkt $BHP(o_u, o_v)$ und Scherungsparameter γ . . .	8
2.4	Radiale Linsenverzeichnung	8
2.5	Transformation mittels Homographiematrix	10
3.1	Versuchsstand zur Kalibrierung des Laserprojektors	14
3.2	Skizze zur Bestimmung der Leinwandmaße	15
3.3	Schematische Darstellung des Trackingsystems DTrack2 . . .	16
3.4	Microvision PicoP Display Engine Development Kit	17
3.5	Integrated Photonics Module	18
3.6	MEMS Spiegel	19
3.7	Visualisierung des Bildaufbaus - Sinusförmige Horizontalschwin- gung	19
3.8	Messkamera: Marlin F-046C	20
3.9	Detektierte Eckpunkte: Visualisierung via CalDe und Matlab	21
3.10	Radiale Verzeichnungserscheinungen der Messkamera	22
3.11	Ideales Kalibrationsmuster	23
3.12	Aufnahme des Kalibrationsmusters	24
3.13	Verzeichnungserscheinungen des Laserprojektors ohne Verwen- dung des radialen Verzeichnungsmodells	25
3.14	Absoluter Pixelfehler ohne Ausgleich der Verzeichnungserschei- nungen des Laserprojektors	25
3.15	Verzeichnungserscheinungen des Laserprojektors unter Ver- wendung des radialen Verzeichnungsmodells	26
3.16	Farbverlaufsbild der horizontal interpolierten Residuen	26

3.17	Farbverlaufsbild der vertikal interpolierten Residuen	27
3.18	Farbverlaufsbild der tatsächlichen Residuen	29
3.19	Farbverlaufsbild der optimierten Residuen	30
3.20	Absoluter Pixelfehler mit Ausgleich der Verzeichnungserscheinungen des Laserprojektors	30
4.1	Quadrat via openGL ES 2.0	33
4.2	Simulation des Autopointers	35
4.3	Benutzeroberfläche der DTrack-Software	37
4.4	Kalibrationswinkel	38
4.5	Im Autopointer implementierte Koordinatentransformationen	38
4.6	Ermittlung der nächstgelegenen Stützpunkte und Residuen .	40
4.7	Gewichtung der interpolierten Residuen	41
5.1	Detektion des projizierten Eckpunkts via CalDe	43
5.2	Ermittelte Projektionsgenauigkeit des Autopointers	44
5.3	Seitenansicht von 2 Markerkugeln des Trackingtargets	45
5.4	Visualisierung des Einflusses des Winkels zwischen Trackingkameras und TKS	46
5.5	Projektionen des Autopointers bei einem Arbeitsabstand von 40 cm	47

Abkürzungsverzeichnis

BE	Bildebene
BHP	Bildhauptpunkt
DLR	Deutsches Zentrum für Luft- und Raumfahrt
IPM	Integrated Photonics Module
KKS	Kamerakoordinatensystem
LUT	Lookup Tabelle
PDK	Pico Development Kit
PDE	PicoP Display Engine
PKS	Projektorkoordinatensystem
TKS	Trackingkoordinatensystem
WKS	Weltkoordinatensystem

Kapitel 1

Einleitung

1.1 Motivation und Zielsetzung

Minimal-invasive Operationstechniken gewinnen seit den 80er Jahren immer mehr an Bedeutung. Ziel hierbei ist es die Verletzungen des Patienten so gering wie möglich zu halten, um die postoperative Genesung zu verbessern. Die damit einhergehende Weiterentwicklung technischer Hilfsmittel ist zentraler Bestandteil der modernen Medizintechnik. Neben dem Einsatz von Robotersystemen oder fortschrittlicher Diagnosetechniken wird versucht, die Realitätswahrnehmung des Chirurgen computergestützt zu erweitern. Diese sog. *Augmented Reality* erlebt nicht nur im medizintechnischen Bereich einen starken Aufschwung. Sie gilt als Kerntechnologie des 21. Jahrhunderts und wird in Smartphones, bei Fahrerassistenzsystemen oder zur Visualisierung von GPS Daten verwendet. [15] [3]

Trotzdem war die Augmented Reality bisher durch die eingeschränkten Möglichkeiten zur Visualisierung der Informationen limitiert. So erfolgt die computergestützte Realitätserweiterung entweder auf sehr begrenztem Raum, z.B. mit Hilfe von Displays oder mit umständlichen und kostspieligen Instrumenten, wie 3-D Brillen oder Head-up-Displays. Nicht zuletzt deswegen wecken Pico-Projektoren, mit denen sich Bilder, Filme oder Präsentationen auf jede beliebige Oberfläche projizieren lassen, weltweit großes Interesse. Insbesondere der Microvision SHOWWX+ Laserprojektor setzt seit seiner

Markteinführung im Jahre 2008 Maßstäbe, da er kontrastreiche Projektionen ermöglicht, ohne dass eine vorherige Fokussierung nötig ist. [15] [3]

Ziel der vorliegenden Arbeit ist es daher, mit Hilfe eines Pico-Laserprojektors durch Projektion in die reale Welt herkömmliche Augmented Reality Methoden zu erweitern. Es soll ein "Autopointer" implementiert werden, der Informationen aus der virtuellen Welt auf bekannte Oberflächen in der realen Welt projiziert. Dabei muss ständig die 6D-Lage des Autopointers relativ zum realen Objekt gemessen werden. Die Projektion wird entgegen der erfassten Bewegung des Autopointer in Echtzeit nachgeführt, so dass das Bild ortsfest auf dem Objekt verbleibt. Im einfachsten Fall wird ein Ort auf dem Objekt durch einen Punkt markiert, möglich ist jedoch auch die Projektion komplexer Grafik. Um eine ausreichende Genauigkeit zu gewährleisten, muss ein geeignetes Kalibriersverfahren mit Untersuchung der Abbildungseigenschaften entwickelt und validiert werden.

1.2 Anwendungen

Mögliche Anwendungsszenarien des Autopointers liegen in der Medizintechnik. So lassen sich beispielsweise präoperativ gewonnene Daten, die auf moderne Diagnosetechniken wie Kernspin- oder Computertomographie beruhen, anzeigen lassen. Neben der Markierung von Einstichstellen für endoskopische Untersuchungen wäre es des Weiteren möglich, dem Chirurgen Informationen über den Gesundheitszustand des Patienten darzustellen. Bei Grenzwertüberschreitungen ist ein Warnsignal denkbar, das deutlich und im Sichtfeld des praktizierenden Arztes projiziert wird.

Neben dem medizintechnischen Bereich eignet sich der Autoprojektor für eine Vielzahl von Anwendungen in Industrie und Wirtschaft. So ist eine Implementierung als Montageassistenzsystem denkbar, bei dem zu bearbeitende Positionen eines Werkstücks markiert oder bei einer definierten Reihenfolge durchnummeriert werden.

Es ist im Folgenden zu klären, ob und mit welcher Genauigkeit die genannten Möglichkeiten umgesetzt werden können.

Kapitel 2

Theoretische Grundlagen der Kamerakalibrierung

Die nachfolgenden theoretischen Grundlagen der Kamerakalibrierung sind für die vorliegende Arbeit in zweierlei Hinsicht von Bedeutung. Zum einen muss die Messkamera, die die Abbildung des Laserprojektors erfasst, kalibriert werden, zum anderen sind die gewonnenen Erkenntnisse in der Software des Laserprojektors zu berücksichtigen, um eine verzerrungsfreie und pixelgenaue Abbildung zu gewährleisten. (siehe Kap. 3)

Das von einer Kamera erfasste Bild stellt die 2D-Projektion eines realen, im 3-dimensionalen Raum gegebenen, Punktes dar. Um von der Aufnahme Rückschlüsse auf die tatsächliche Position des Objekts ziehen zu können, müssen verschiedene Transformationsparameter bekannt sein, welche bei der Kamerakalibrierung ermittelt werden. Die Kalibrierung ist somit zentraler Bestandteil jeder computergestützten Messung mit Kameras. Es wird zwischen extrinsischer und intrinsischer Kalibrierung unterschieden, welche im Folgenden näher erklärt werden. [9] [10]

2.1 Extrinsische Kalibrationsparameter

Die extrinsischen Kalibrationsparameter¹ beschreiben die Lage und Orientierung der Kamera relativ zum Weltkoordinatensystem (WKS). Sie bestehen aus den 3 Elementardrehungen R_x , R_y und R_z sowie einer translatorischen Ursprungsverschiebung t und sind bei sich bewegender Kamera und konstantem WKS variabel. [10]

Die 3×3 Rotationsmatrizen bei einer Elementardrehung um die jeweilige Achse und um den Winkel α sind wie folgt definiert:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad (2.1)$$

$$\mathbf{R}_y = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix} \quad (2.2)$$

$$\mathbf{R}_z = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

Nach Craig [2] gilt für eine 3×3 -Rotationsmatrix mit Drehungen um die Winkel γ , β und α in der sog. „Roll-Pitch-Yaw“- Rotationsreihenfolge:

$$\mathbf{R}_{XYZ}(\gamma, \beta, \alpha) = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix} \quad (2.4)$$

mit $c_i = \cos(i)$ und $s_i = \sin(i)$. Die translatorische Verschiebung des Ursprungs wird mit dem Vektor t beschrieben:

$$\mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (2.5)$$

¹Parameter der äußeren Orientierung

Unter Verwendung einer homogenen 4×4 Matrix ${}^A_B\mathbf{T}$ kann die Transformation von einem System B in ein System A zusammengefasst werden. Hierbei beschreibt die 3×3 -Rotationsmatrix ${}^A_B\mathbf{R}$ die Lage der beiden Systeme zueinander und ${}^A_B\mathbf{t}$ die translatorische räumliche Verschiebung.

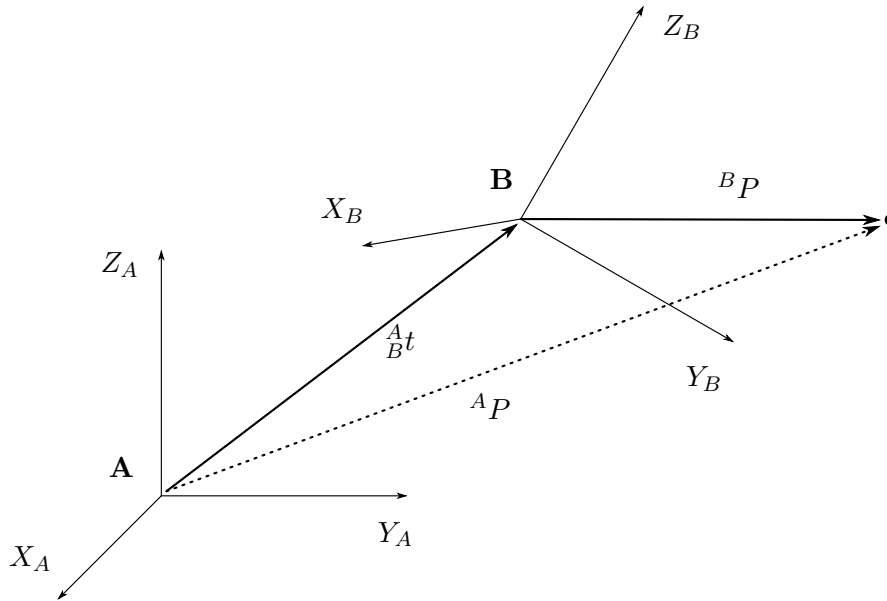


Abbildung 2.1: Koordinatentransformation eines Punktes ${}^B\mathbf{P}$ in das System A unter Verwendung von ${}^A_B\mathbf{T}$

$${}^A_B\mathbf{T} = \begin{pmatrix} {}^A_B\mathbf{R} & {}^A_B\mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Mit

$${}^A\mathbf{P} = {}^A_B\mathbf{T} {}^B\mathbf{P} \quad (2.7)$$

kann der homogene Vektor ${}^B\mathbf{P}$ mit Hilfe von ${}^A_B\mathbf{T}$ vom System B in das System A transformiert werden (vgl. Abb. 2.1). Die Transformationsmatrix dient dazu, Rotation und Translation in einfacher Matrixform darzustellen und lässt sich nach Craig [2] in invertierter Form anschreiben:

$${}^B_A\mathbf{T} = \begin{pmatrix} {}^A_B\mathbf{R}^T & -{}^A_B\mathbf{R}^T {}^A_B\mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

mit

$${}^A_B\mathbf{R}^T = {}^B_A\mathbf{R}$$

2.2 Intrinsische Kalibrationsparameter

Bei den intrinsischen Kalibrationsparametern² handelt es sich um konstante, kamerainterne Größen, die benötigt werden, um einen im Kamerakoordinatensystem (KKS) gegebenen Punkt auf die 2D-Bildebene (BE) zu transformieren. Es gibt verschiedene Ansätze, um die Parameter zu bestimmen, die sich je nach Aufbau der Kamera und geforderter Qualität der Kalibrierung unterscheiden. Die einfachste und meistverwendete Approximation der inneren Kamerageometrie ist ein erweitertes Lochkameramodell. Es ist nach Strobl [8] auch für die meisten moderne Kameras ausreichend genau und der Kernpunkt jedes Projektionsmodells für computergestütztes Sehen.

2.2.1 Das Lochkameramodell

Das Lochkameramodell als mathematisches Ersatzsystem zur Bestimmung der intrinsischen Kameraparameter ist in Abbildung 2.2 dargestellt. Linsensystem

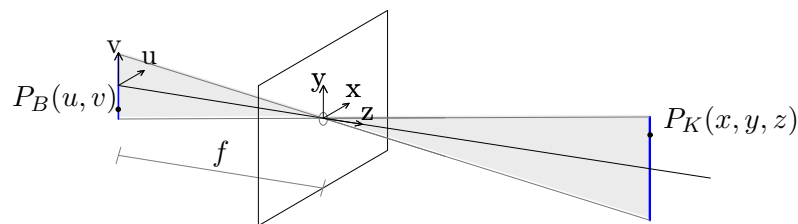


Abbildung 2.2: Schematische Darstellung des Lochkameramodells zur Projektion eines Punktes $P_{KKS}(x, y, z)$ auf die 2-D Bildebene ($P_{BE}(u, v)$)

und Objektiv der Kamera werden mit einem einzigen geometrischen Punkt, dem sog. *optischen Zentrum* approximiert. Treffen Lichtstrahlen eines leuchtenden oder beleuchteten Objekts auf das Zentrum, wird das 3-dimensionale Objekt auf eine 2-dimensionalen gegenüberliegenden Bildebene projiziert. Sie befindet sich im Abstand f , welcher als *Brennweite* bezeichnet wird, hinter dem optischen Zentrum. Der Normalenvektor durch das optische Zentrum auf die Bildebene wird als *optische Achse* bezeichnet. In der Praxis würde ein unendlich kleines Loch eine unendlich scharfe und unendlich lichtschwa-

²Parameter der inneren Orientierung

che Abbildung erzeugen. [4] [9] [5]

Demnach lässt sich die ideale Projektion eines Punktes vom 3-D KKS zur 2-D BE nach Normierung und unter Verwendung des Strahlensatzes mit den Gleichungen

$$u_{BE} = -f \frac{x_{KKS}}{z_{KKS}} \quad \text{und} \quad v_{BE} = -f \frac{y_{KKS}}{z_{KKS}} \quad (2.9)$$

und der Transformationsmatrix

$$\begin{matrix} BE \\ KKS \end{matrix} \mathbf{T}_{intr} = \begin{pmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

definieren.

2.2.2 Erweiterung des Lochkammermodells

Da das Modell, mit f als einzigem internen Parameter, das Abbildungsverhalten moderner Kameras nur ungenau approximiert, sind verschiedene weitere Faktoren zu berücksichtigen:

- Eine nicht orthogonale Anordnung von u und v , dargestellt in Abbildung 2.3, ist mit dem dimensionslosen Scherungsparameter γ auszugleichen. [5]
- Da die optische Achse nicht zwangsläufig den Mittelpunkt des Sensors durchstößt, muss ein Bildhauptpunkt ($BHP(o_u, o_v)$) definiert und berücksichtigt werden. [9]
- Bei entweder nicht-quadratischen Pixeln oder einer nicht-kreisrunden Linse, ist zwischen Brennweite in horizontaler (f_x) und vertikaler (f_y) Richtung zu unterscheiden. [9]

Nach Berücksichtigung der internen Parameter sowie eines Skalierungsfaktors s ergibt sich unter Verwendung der intrinsischen Kamerakalibrationsmatrix $\begin{matrix} BE \\ KKS \end{matrix} \mathbf{T}_{intr}$ die Transformation eines Punktes $P_{KKS}(x, y, z)$ vom KKS auf die BE zu:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_{BE} = \begin{pmatrix} f_x & \gamma & o_u \\ 0 & f_y & o_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}_{KKS} \quad (2.11)$$

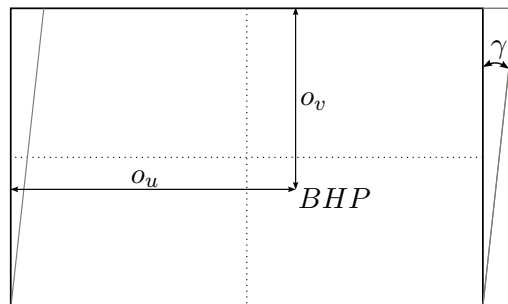


Abbildung 2.3: BHP (o_u, o_v) und Scherungsparameter γ

Es ist zu untersuchen, ob das der Kamerakalibrierung zugrundeliegende Lochkammermodell für die Kalibrierung des Laserprojektors verwendet werden kann oder ob ein komplexeres System gefunden werden muss, das die Transformationseigenschaften des Projektors besser beschreibt. [4] [8]

2.2.3 Radiale Linsenverzeichnung

Neben den linearen Einflüssen auf die Projektion gibt es verschiedene nicht-lineare Verzerrungen, wobei als dominierender Faktor die radiale Linsenverzeichnung zu nennen ist. Sie wirkt sich am stärksten auf die Abbildungseigenschaften der Kamera aus (siehe Abb. 2.4). [7]

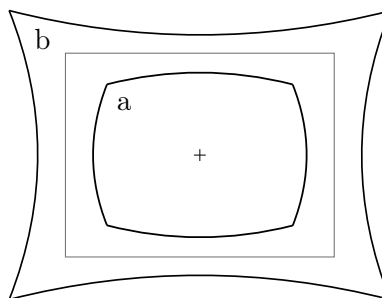


Abbildung 2.4: a) negative radiale Verzeichnung (tonnenförmig)
b) positive radiale Verzeichnung (kissenförmig)

Um die Aufnahmen der Kamera für die Vermessung der Projektionen mit Hilfe der Homographie verwenden zu können, werden die unverzerrten Bildkoordinaten (u, v) benötigt.

Nach Hirschmüller [4] lässt sich die radiale Verzeichnung mit Hilfe der beiden

Verzeichnungparameter 1. und 2. Ordnung k_1 und k_2 berücksichtigt.

$$\begin{pmatrix} \bar{P}_x \\ \bar{P}_y \\ \bar{P}_z \end{pmatrix} = \begin{pmatrix} \check{P}_x + \check{P}_x(k_1(\check{P}_x^2 + \check{P}_y^2) + k_2(\check{P}_x^2 + \check{P}_y^2)^2) \\ \check{P}_y + \check{P}_y(k_1(\check{P}_x^2 + \check{P}_y^2) + k_2(\check{P}_x^2 + \check{P}_y^2)^2) \\ 1 \end{pmatrix} \quad (2.12)$$

Mit $\bar{P}_{x,y,z}$ als entzerrte Koordinate und $\check{P}_{x,y,1}$ als verzerrte und normierte Koordinate im KKS.

Da die mit der Kamera aufgenommenen Punkte jedoch in verzerrten Bildkoordinaten (\check{u}, \check{v}) vorliegen und nicht im 3-dimensionalen Raum des KKS, muss die tatsächliche 2-D-Position (u, v) rekursiv ermittelt werden. Die übliche Vorgehensweise ist ein iteratives Verfahren [8]:

Nach Subtraktion des BHP(o_u, o_v)

$$\check{u}_{temp} = \check{u} - o_u \quad \text{bzw.} \quad \check{v}_{temp} = \check{v} - o_v \quad (2.13)$$

und Normierung mittels Division durch die Brennweiten

$$\check{u}_{norm} = \frac{\check{u}_{temp}}{f_x} \quad \text{bzw.} \quad \check{v}_{norm} = \frac{\check{v}_{temp}}{f_y} \quad (2.14)$$

dienen $\check{u}_{norm,n}$ und $\check{v}_{norm,n}$, also die normierte Koordinaten beim Iterationsschritt n , zusammen mit den Verzerrten \check{u} und \check{v} als Startwert für die Iteration. Mit den radialen Verzeichnungskoeffizienten k_1 und k_2 sowie

$$r_n = \check{u}_{norm,n}^2 + \check{v}_{norm,n}^2 \quad (2.15)$$

gilt:

$$\check{u}_{n+1} = \check{u} + k_1 \cdot \check{u}_{temp,n} \cdot r_n + k_2 \cdot \check{u}_{temp,n} \cdot r_n^2 \quad (2.16)$$

$$\check{v}_{n+1} = \check{v} + k_1 \cdot \check{v}_{temp,n} \cdot r_n + k_2 \cdot \check{v}_{temp,n} \cdot r_n^2 \quad (2.17)$$

Die verzerrten Punkte werden also temporär als unverzerrt angenommen und mit den radialen Verzeichnungsparametern k_1 und k_2 verzerrt. Der entstehende Fehler wird nun vom temporären Punkt abgezogen

$$\check{u}_{temp,n+1} = \check{u}_{temp,n} - error_u / \check{v}_{temp,n+1} = \check{v}_{temp,n} - error_v \quad (2.18)$$

und solange minimiert, bis das Abbruchkriterium

$$|(\check{u}_{n+1} - \check{u}_n) + (\check{v}_{n+1} - \check{v}_n)| \leq \varepsilon \quad \text{mit } \varepsilon = \frac{1}{100} \quad (2.19)$$

erfüllt ist. Als letztes werden die unverzerrten Bildkoordinaten (u, v) gespeichert:

$$u = \check{u}_{n+1} \quad \text{und} \quad v = \check{v}_{n+1} \quad (2.20)$$

Der ausführliche Quellcode ist Anhang A.1 zu entnehmen. [8] [11]

2.3 Homographie

Die Homographie ist eine projektive invertierbare Transformation, mit der Ebenen aufeinander abgebildet werden können. Der direkte Zusammenhang kollinearier Punkte kann mittels einer 3×3 Homographiematrix H genutzt werden, um aus einer 2D Pixelkoordinate die tatsächliche Position auf einer Ebene mit $n \geq 4$ bekannten Punkten zu berechnen. [6]

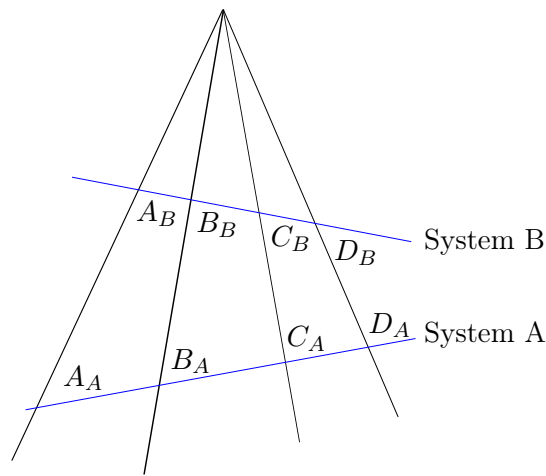


Abbildung 2.5: Transformation von 4 Punkten vom System A in das System B unter Verwendung der Homographiematrix ${}^B_A H$

Abbildung 2.5 zeigt die Transformation der Punkte A_A , B_A , C_A und D_A vom System A in das System B unter Verwendung der Homographiematrix ${}^B_A H$.

Es gilt:

$$p_B = {}^B_A H \cdot p_A \quad (2.21)$$

bzw.

$$\begin{pmatrix} x_B \\ y_B \\ k' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix} \quad (2.22)$$

Um eine eindeutige Lösung zu erhalten ist zu beachten, dass die Punkte nicht kollinear zueinander sind. Mit $k' = 1$ ergeben sich die nach 0 aufgelösten Gleichungen

$$h_{11}x + h_{12}y + h_{13} - h_{31}x' + h_{32}y' + h_{33}x' = 0 \quad (2.23)$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}y'x + h_{32}y'y + h_{33}y' = 0 \quad (2.24)$$

Da eine projektive Homographiematrix 8 Freiheitsgrade besitzt, sind zur Lösung des Gleichungssystems 4 Referenzpunkte nötig, um die Lösungsmatrix L nach $Lh = 0$ aufzustellen.

$$\begin{pmatrix} x_{A,1} & y_{A,1} & 1 & 0 & 0 & 0 & -x_{B,1} \cdot x_{A,1} & -x_{A,1} \cdot y_{A,1} & -x_{B,1} \\ 0 & 0 & 0 & x_{A,1} & y_{A,1} & 1 & -y_{B,1} \cdot x_{A,1} & -y_{B,1} \cdot y_{A,1} & -y_{B,1} \\ x_{A,2} & y_{A,2} & 1 & 0 & 0 & 0 & -x_{B,2} \cdot x_{A,2} & -x_{A,2} \cdot y_{A,2} & -x_{B,2} \\ 0 & 0 & 0 & x_{A,2} & y_{A,2} & 1 & -y_{B,2} \cdot x_{A,2} & -y_{B,2} \cdot y_{A,2} & -y_{B,2} \\ x_{A,3} & y_{A,3} & 1 & 0 & 0 & 0 & -x_{B,3} \cdot x_{A,3} & -x_{A,3} \cdot y_{A,3} & -x_{B,3} \\ 0 & 0 & 0 & x_{A,3} & y_{A,3} & 1 & -y_{B,3} \cdot x_{A,3} & -y_{B,3} \cdot y_{A,3} & -y_{B,3} \\ x_{A,4} & y_{A,4} & 1 & 0 & 0 & 0 & -x_{B,4} \cdot x_{A,4} & -x_{A,4} \cdot y_{A,4} & -x_{B,4} \\ 0 & 0 & 0 & x_{A,4} & y_{A,4} & 1 & -y_{B,4} \cdot x_{A,4} & -y_{B,4} \cdot y_{A,4} & -y_{B,4} \end{pmatrix} h = 0 \quad (2.25)$$

Der Spaltenvektor h entspricht dem Nullraum der Lösungsmatrix L und beschreibt, wie aus Gl. (2.22) ersichtlich, die Homographiematrix ${}^B_A H$.

Der ausführlicher Quellcode zur Erstellung der Lösungsmatrix und zur Berechnung des Nullraums mittels Singulärwertzerlegung ist Anhang A.2 zu entnehmen. [6]

Kapitel 3

Laserprojektorkalibrierung

Für die Laserprojektorkalibrierung wird das in Abbildung 3.11 dargestellte Schachbrettmuster (pattern) auf eine Leinwand projiziert und von einer kalibrierten Kamera erfasst. Ziel ist es, die reale Projektion möglichst genau zu vermessen, um analog zu einer Kamerakalibrierung die intrinsische Kalibrierparameter zu bestimmen. Dabei soll ermittelt werden, inwieweit das radiale Verzeichnungsmodell auf den Laserprojektor anwendbar ist. Aus den verbleibenden Residuen¹ sollen ggf. Rückschlüsse auf ein alternatives Verzeichnungsmodell gezogen werden.

Die Darstellung des pattern erfolgt mit OpenGL ES, einer plattformunabhängigen Programmierschnittstelle, die bei allen nachfolgenden Abbildungen Verwendung findet und in Kapitel 4.1 näher beschrieben ist. Um die Projektion mit Hilfe einer ortsfesten Kamera zu vermessen, wird der homographische Zusammenhang, also die Kollineation zwischen Bildsensor und ebener Leinwand, ausgenutzt. Neben den intrinsischen Parametern des Projektors wird die Lage des Laserprojektorkoordinatensystems (PKS) relativ zum WKS, also die extrinsischen Parameter, aus den Projektionen und den Lagekoordinaten ermittelt. Die Berechnung erfolgt mit Hilfe der am Deutschen Zentrum für Luft- und Raumfahrt (DLR) entwickelten Kamerakalibrationssoftware CalDe² und CalLab³, sowie eines Trackingsystems, das die Position des La-

¹Abweichungen vom Ideal

²Calibration Detection Toolbox

³Calibration Laboratory Toolbox

serprojektors erfasst. Für die Laserprojektorkalibrierung sind also folgende Arbeitsschritte nötig, die im Anschluss näher beschrieben werden:

- Intrinsische Kalibrierung der Messkamera:
Aufnahme eines bekannten pattern aus mind. 4 unterschiedlichen Perspektiven und Ermittlung der intrinsischen Transformationsmatrix $\mathbf{T}_{intr,CAM}$ via CalDe.
- Extrinsische Kalibrierung der Messkamera:
Orthogonale Positionierung und Vermessung von 4 Punkten auf der Leinwand, die mit der Messkamera aufgenommen wird. Nach Detektion der Pixelkoordinaten lässt sich die Homographiematrix zur Transformation der Projektion (Bildkoordinaten \rightarrow Leinwandkoordinaten) berechnen.
- Erstellung und Projektion des pattern aus 4 unterschiedlichen Perspektiven mit anschließender Detektion der Eckpunkte.
- Beschreibung der Abbildungseigenschaften des Laserprojektors inkl. Validierung der Messergebnisse.
- Intrinsische Kalibrierung des Laserprojektors:
Ermittlung der Transformationsmatrix $\mathbf{T}_{intr,LP}$ und Ausgleich des gemessenen Verzeichnungsverhaltens mit Hilfe einer LUT
- Extrinsische Kalibrierung des Laserprojektors:
Erfassung der Trackingdaten und Ermittlung der Transformationsmatrix $\mathbf{T}_{extr,LP}$ (Trackingkoordinaten \rightarrow Laserprojektorkoordinaten)

3.1 Versuchsaufbau und Komponenten

Der Versuchsaufbau, sowohl für die Kalibrierung des Laserprojektors, als auch für die Umsetzung des Autopointers, ist mit Ausnahme der Remote-PCs in Abbildung 3.1 dargestellt und besteht aus folgenden Komponenten:

1. Aluminiumrahmen
2. Matt-weiße Dibondplatte als Leinwand mit 4 Punkten zur Bestimmung der Homographiematrix

3. Stativ und Aluminiumplatte zur Aufnahme des Laserprojektors und des Trackingtargets
4. Laserprojektor
5. Trackingtarget bestehend aus 4 Markern
6. 2 Trackingcameras inkl. Netzteil
7. Controller des Trackingsystems
8. Messkamera
9. Remote-PC

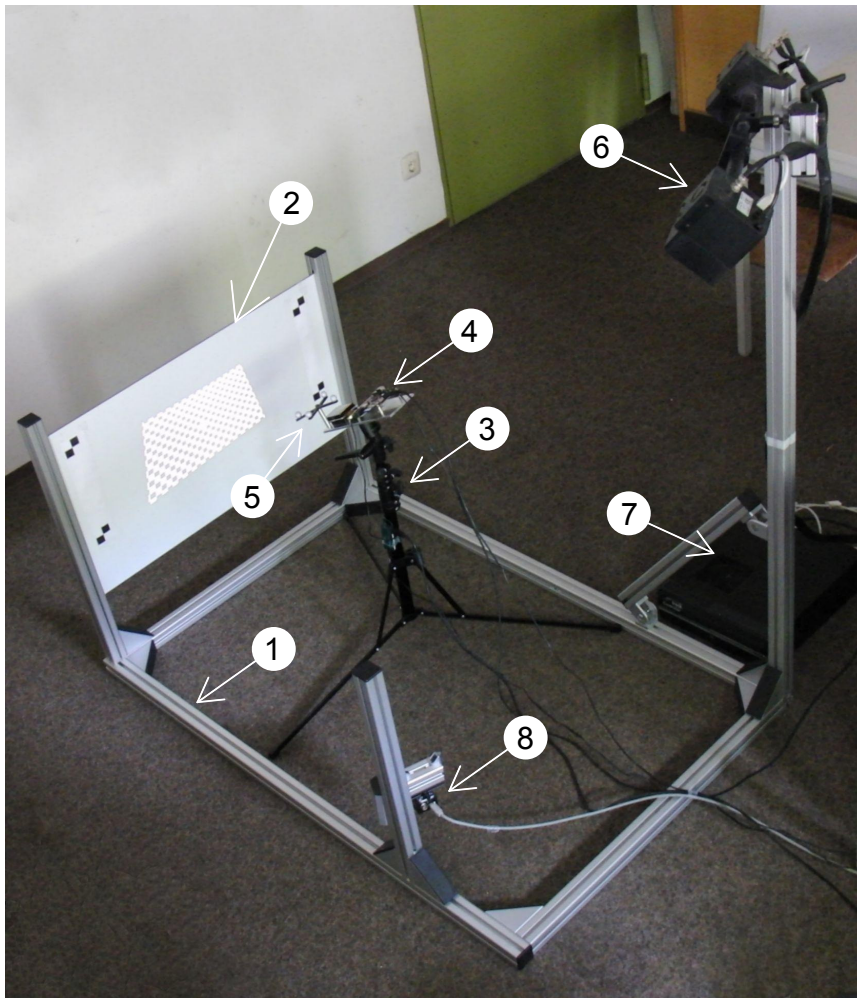


Abbildung 3.1: Versuchsstand zur Kalibrierung des Laserprojektors

Der Aufbau des Versuchsstandes orientiert sich an verschiedenen geometrischen Gegebenheiten, die eingehalten werden mussten:

Bei einem Arbeitsabstand l von ca. 30 cm sowie einem Einfallswinkel von $\pm 30^\circ$ und einer horizontalen Laserprojektorapertur A_{hor} von $43,2^\circ$ ergibt sich nach

$$\frac{b_{lw}}{2} = \tan\left(30^\circ + \frac{A_{hor}}{2}\right) \cdot l \quad (3.1)$$

eine Leinwandbreite b_{lw} von mindestens 75 cm (vgl. Abbildung 3.2).

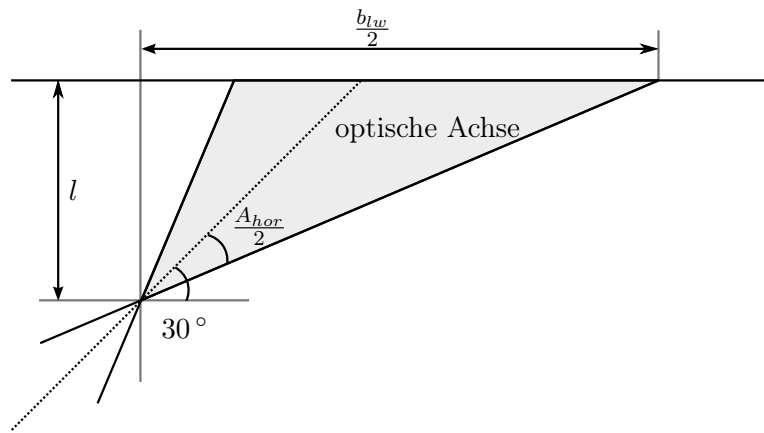


Abbildung 3.2: Schematische Ansicht des Laserprojektors von oben zur Bestimmung der Leinwandmaße

Mit einer Toleranz behaftet und unter Einhaltung der Seitenverhältnisse des Laserprojektors wurde eine Leinwandgröße von 85×45 cm gewählt. Um sie komplett zu Erfassen ist bei einer horizontalen Messkameraapertur von ca. 32° eine Rahmengesamtlänge von 1,20 m nötig. Das Stativ zur Halterung der Trackingkameras wurde so gewählt, dass die Marker im optimalen Messbereich des Trackingsystems (ca. 1,50 m) liegen.

Der stabile Aluminiumrahmen ist erforderlich, um die Ausrichtung der Messkamera relativ zu Leinwand und den darauf angebrachten Punkten zur Bestimmung der Homographie konstant zu halten. Da eine minimale Veränderung der Kameraposition starke Auswirkungen auf die Messergebnisse hat, muss sie trotzdem vor jeder Messung mit Hilfe des in Anhang A.2 hinterlegten Quellcodes überprüft werden.

3.1.1 Trackingsystem

Zur Bestimmung der Lagekoordinaten des Laserprojektors wird das Trackingsystem DTrack2 der Firma ART⁴ verwendet. Die beiden CCD⁵ Kameras senden synchronisierte Infrarot-Impulse aus, die von kugelförmigen, passiven Markern reflektiert werden. Sie sind mit einer retroreflektierenden Folie überzogen, um die einfallende Strahlung zur emittierenden Quelle zurückzuwerfen.

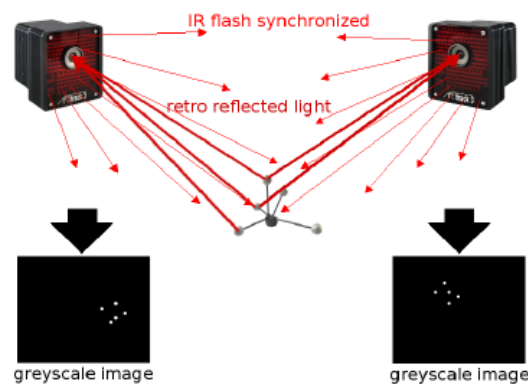


Abbildung 3.3: Tackingsystem DTrack2 bestehend aus 2 Kameras und einem Trackingtarget [1]

Die Kameras detektieren die Marker in einer Entfernung von bis zu 4 m bei einem optimalen Arbeitsabstand von ca. 1,5 m und erstellen ein Graustufenbild des Arbeitsbereichs. Da der Schwerpunkt der erfassten Flächen berechnet wird, haben größerer Markerkugeln eine verbesserte Trackinggenauigkeit zu Folge. In Abb. 3.3 ist das Gesamtsystem mit dem sog. Trackingtarget gut zu erkennen, welches mit Hilfe mehrerer Marker ein eindeutiges Koordinatensystem definiert. Nach einmaliger Kalibrierung lassen sich aus dem bekannten Offset der Kameras und den unterschiedlichen Graustufenbildern Lage und Orientierung des Targets bestimmen. Die Berechnungen erfolgen im ART Tracking Controller, der die Daten via ASCII und UDP mit Hilfe eines lokalen Netzwerks an einen Remote PC schickt. [1]

⁴Advanced Realtime Tracking GmbH

⁵charge couple device

3.1.2 Laserprojektor

Als Laserprojektor dient der SHOWWX+ Laser Pico Projector von Microvision in der Ausführung „Microvision PicoP Display Engine Development Kit“ (PDK) (siehe Abb. 3.4).

Es besteht aus einem Beagleboard inkl. PicoP Display Engine (PDE)-Beagleboard Interface-Modul sowie einem Integrated Photonics Module (IPM) als display engine.



Abbildung 3.4: Pico development kit bestehend aus beagleboard (rote Platine), PDE-Beagleboard Interface-Modul (grüne Platine) und display engine [16]

Das Beagleboard als computer-on-module system ist ein Einplatinen- Computer, der zusammen mit dem PDE-Beagleboard Interface zur Ansteuerung der IPM dient.

Das Board zeichnet sich durch minimalen Stromverbrauch und geringen Bau- raum aus und ist mit dem modularen Betriebssystem ubuntu-Linux openGL ES kompatibel, also geeignet für die gegebenen Anforderungen. Die techni- schen Daten des Beagleboards in der Version BB-XM sind Tabelle 3.1 zu entnehmen. In der Microvision IPM werden die 3 Laserstrahlen jeweils von einer Linse gebündelt und unter Verwendung eines dichroitischen Spiegels, welcher nur einen Teil des Lichtspektrums reflektiert, zu einem Strahl mit unterschiedlicher Rot-,Grün- und Blauintensität kombiniert (vgl. Abb. 3.5).

Tabelle 3.1

Technische Daten des Beagleboards von Texas Instruments

ARM Cortex-A8 Prozessor
256 MB LPDDR RAM
256 MB NAND Flash
DVI-D und S-video output
SD/MMC socket
USB 2.0
OpenGL ES 2.0 kompatibel
HD video kompatibel DSP

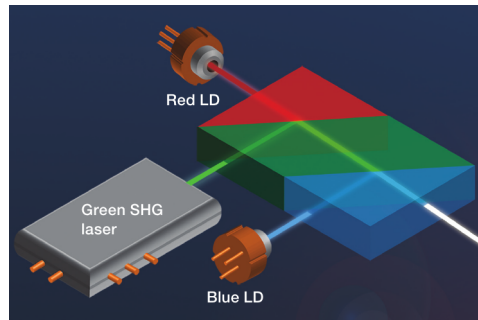


Abbildung 3.5: Integrated Photonics Module [16]

Der biaxiale MEMS-Spiegel (vgl. 3.6) mit einem Durchmesser von ca. 1 mm lenkt den Strahl in 2 orthogonale Richtungen ab. Er schwingt bei sinusförmiger Bewegung in Horizontalrichtung mit seiner Eigenfrequenz von 18 kHz, was den Energieverbrauch und damit die Wärmeentwicklung minimiert. Das Motiv wird zeilenweise bidirektional projiziert, wobei die horizontale Abbildungsgeschwindigkeit von der Position des Laserstrahls abhängt (vgl. Abb. 3.7). Um eine konstante vertikale Geschwindigkeit zu gewährleisten, wird eine Sägezahnwelle verwendet, die nach einmaligem Durchlauf von oben nach unten wieder an den Anfang der konstanten Bewegung springt. Es wird also nur in einer Bewegungsrichtung projiziert, wobei Geschwindigkeit und Position vom MEMS controller in Abhängigkeit vom piezoresistiven Sensor geregelt werden. Die Ansteuerung des Spiegels erfolgt über einen Galvanometerantrieb, d.h. dass die in Abbildung 3.6 dargestellten Leiterbahnen als

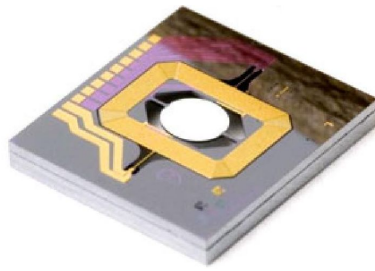


Abbildung 3.6: MEMS Spiegel [16]

Spule agieren und sich durch Veränderung des Stromflusses in einem Magnetfeld ausrichten.

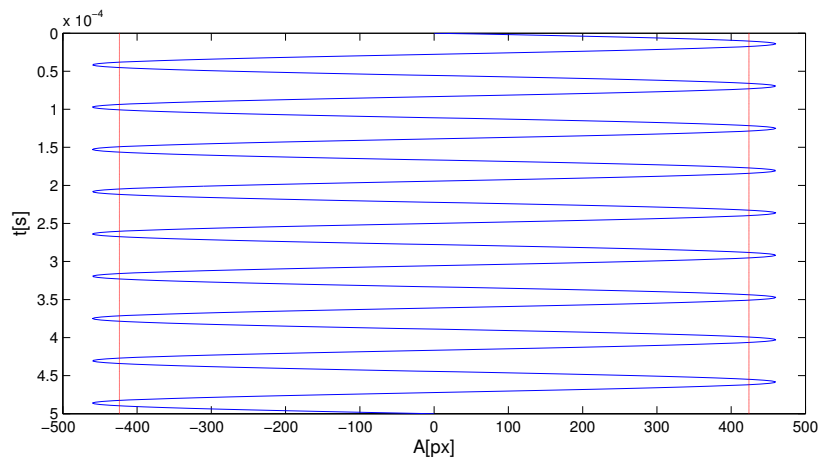


Abbildung 3.7: Visualisierung des Bildaufbaus - Sinusförmige Horizontalschwingung

In Abbildung 3.7 ist ein 0,5 ms dauernder Ausschnitt der sinusförmige Horizontalschwingung dargestellt. Um ein möglichst lineares Verhalten zu erreichen wird nicht die komplette Schwingung ausgenutzt, sondern nur der Bereich innerhalb der roten Grenzen. Es ist zu beachten, dass der Bildaufbau in beide Richtungen und bei einer framerate von 60 Hz erfolgt. Alle weiteren technischen Daten sind Tabelle 3.2 zu entnehmen.

Tabelle 3.2

Technische Daten des SHOWWX+ Laser Pico Projector von Microvision	
Apertur	$43,2^\circ \times 24,2^\circ$
Auflösung	848×480 px
Frequenz horizontal	18000 Hz
Framerate	60 Hz
Helligkeit	15 Lumen

3.1.3 Kamera

Die Aufnahme der Leinwand erfolgt mit einer Kamera des Typs Marlin F-046C inkl. Cinegon 1.4/8-0902 Objektiv von Schneider/Kreuznach (siehe Abb. 3.8). Die Kalibrierung der Kamera wurde in Kapitel 3.2 beschrieben. Die technischen Daten sind Tabelle 3.3 zu entnehmen.



Abbildung 3.8: Kamera zur Aufnahme der Leinwand - Marlin F046C

Tabelle 3.3

Technische Daten der Marlin F-046C inkl. Cinegon1.4/8-0902	
Apertur	$42,0^\circ \times 32^\circ$
Bildsensor	$\frac{1}{2}$ Zoll CCD
Auflösung	780×580 px
Maße	$58 \text{ mm} \times 44 \text{ mm} \times 29 \text{ mm}$

3.2 Intrinsische Messkamerakalibrierung

Im ersten Schritt ist die Kamera, die zur Vermessung der Projektion dient, zu kalibrieren. Hierzu wird ein pattern aus mindestens 4 unterschiedlichen Perspektiven aufgenommen, um im Anschluss die Eckpunkte auf dem Schachbrettmuster via CalDe zu erfassen (vgl. Abb. 3.9).

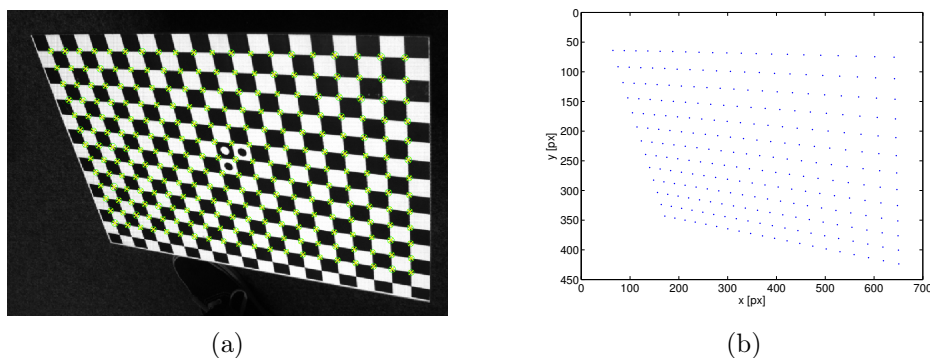


Abbildung 3.9: a) Detektion der Pixelkoordinaten eines bekannten pattern
b) Visualisierung der erfassten Punkte

CalDe sucht hierbei nach hohen Farbgradienten um die Kanten zu detektieren und ermittelt aus Symmetrien den Eckpunkt subpixelgenau. Um die intrinsische Transformationsmatrix $\mathbf{T}_{intr,CAM}$ zu berechnen wird die erfasste Punktmenge zusammen mit den aufgenommenen Bildern in CalLab geladen und ausgewertet.

Bei CalLab handelt es sich um eine am DLR entwickelte Software zur Kalibrierung von Messkameras. Es wird der linearen Zusammenhang zwischen Kalibrieremuster und Darstellung auf dem Bildsensor verwendet. CalLab optimiert, ausgehend von den bekannten Eckpunktabständen des Kalibrationspattern, die berechneten intrinsischen Kameraparameter und die radialen Verzeichnungskoeffizienten iterativ (vgl. Kapitel 2.2.3), sodass die Residuen als Restfehler übrig bleiben. [11]

Die Ergebnisse der Kalibrierung, bestehend aus der intrinsischen Transformationsmatrix $\mathbf{T}_{intr,CAM}$ und den radialen Verzerrungskoeffizienten r_1 und r_2 , befinden sich in Anhang A.3.

Die Pfeile in Abbildung 3.10 zeigen vom entzerrten zum verzerrten Bildpunkt

in 5-facher Vergrößerung. Das dargestellte Verzeichnungsverhalten muss ausgeglichen werden, um die Aufnahmen der Kamera zur Vermessung einer Projektion verwenden zu können. Wie in den Gleichungen (2.16) und (2.17) beschrieben, besteht ein polynomialer Zusammenhang zwischen Abstand zum BHP und Betrag der Verzerrung in radiale Richtung. Die Brennweiten f_x und f_y sind identisch, was auf quadratische Pixel im Bildsensor hinweist. Des Weiteren ist der verzerrungsfrei BHP (381.55 px, 285.91 px) gut zu erkennen.

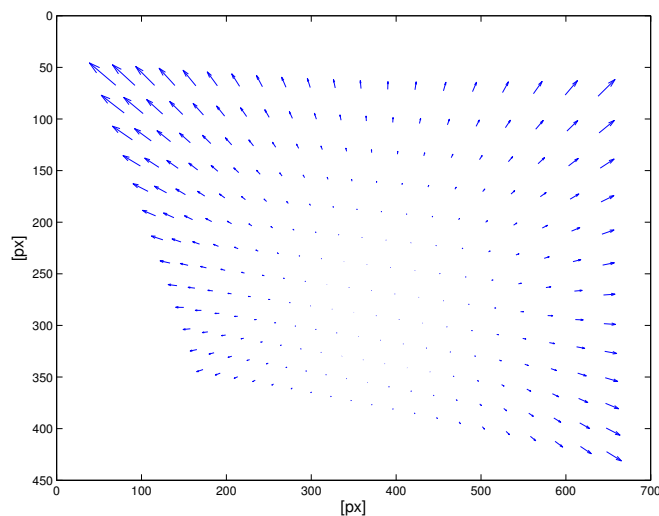


Abbildung 3.10: Visualisierung der Verzeichnungserscheinungen der Messkamera in 5-facher Vergrößerung.

3.3 Extrinsische Messkamerakalibrierung unter Verwendung der Homographie

Wie in Kapitel 2.3 beschrieben, lassen sich unter Verwendung der Homographie bei mindestens 4 bekannten Punktpaaren 2 Ebenen aufeinander abbilden. Es besteht also ein linearer Zusammenhang, der es ermöglicht aus den Aufnahmen der ortsfesten, kalibrierten Kamera die tatsächliche Position von Koordinaten auf der Leinwand zu bestimmen. Die hierbei erreichte Genauigkeit hängt stark von der korrekten Ausrichtung der Markierungen

ab, welche das Koordinatensystem aufspannen. Sie wurden manuell auf der Leinwand angebracht und vermessen. Die exakten Daten der Markierungspositionen und der entsprechenden Pixelkoordinaten sind Anhang A.2 zu entnehmen. Mit den Leinwandkoordinaten $p_{1\dots 4, lw}$ und den korrespondierenden, entzerrten Pixelkoordinaten $p_{1\dots 4, img}$ ergibt sich unter Verwendung der Formeln (2.22) und (2.25) die Homographiematrix H , mit

$$p_{i, lw} = H \cdot p_{i, img} \quad (3.2)$$

Der ausführliche Quellcode zur Ermittlung der Homographiematrix H ist Anhang A.2 zu entnehmen. Sie ist invertierbar, sodass mit

$$p_{i, img} = H^{-1} \cdot p_{i, lw} \quad (3.3)$$

bekannte Leinwandkoordinaten in Pixelkoordinaten transformiert werden können. Mit H lässt sich also die in Pixelkoordinaten gegebene Punktmenge der detektierten Stützpunkte in Leinwandkoordinaten umrechnen, sodass das reale Abbildungsverhalten des Laserprojektors vermessen und untersucht werden kann.

3.4 Projektion des pattern und Detektion der Eckpunkte

Im Gegensatz zur Kamerakalibrierung wird bei der Kalibrierung des Laserprojektors kein reales pattern aufgenommen, sondern ein Muster (vgl. Abb. 3.11) mit definierten Pixelwerten projiziert.

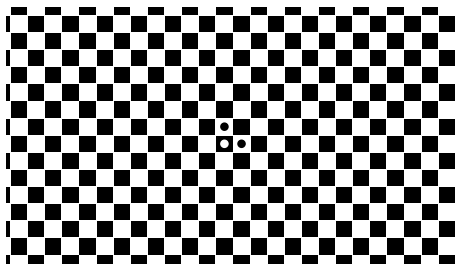


Abbildung 3.11: Ideales Kalibrationsmuster zur Untersuchung der Abbildungseigenschaften des Laserprojektors

Das „wahre“ Bild befindet sich also nicht in der Realität (Einheit mm), wie bei der Kamerakalibrierung, sondern auf der Sensorebene (Einheit px). Dieses ideale und nur virtuell existierende Pixelmuster wird aus 4 unterschiedlichen Perspektiven verzerrt projiziert (vgl. Abb. 3.12) und muss durch ein kalibriertes Messsystem erfasst werden. Abbildung 3.12 zeigt eine Messkameraaufnahme des projizierten pattern. Um die Residuen unter Verwendung

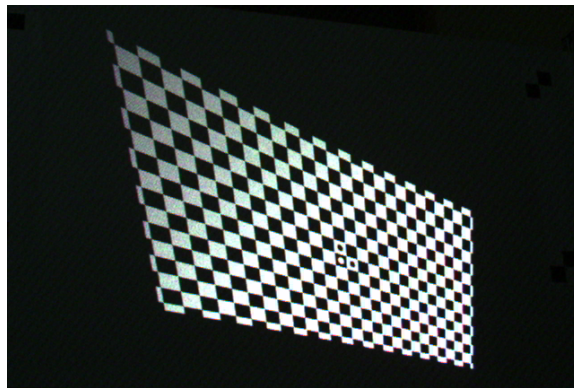


Abbildung 3.12: Aufnahme der Messkamera: Mit Laserprojektor projiziertes Schachbrettmusters

von CalLab zu ermitteln, müssen die Daten nur vertauscht werden. D.h. dass mit CalDe detektierte Punktmenge aus dem projizierten Bild in der Realität als Ideal definiert wird und die tatsächlich idealen Pixelkoordinaten in CalLab als „verzerrte Aufnahme“. Ergebnis sind die umgekehrten Residuen, also die negative Abweichung der Projektion vom pixelgenauen pattern, welches nur theoretisch existiert.

3.5 Abbildungseigenschaften des Laserprojektors

Es hat sich gezeigt, dass das für die Kamerakalibrierung übliche erweiterte Lochkammermodell unter Berücksichtigung der radialen Linsenverzeichnung nur in grober Näherung auf den Laserprojektor angewendet werden kann. Zwar ist die intrinsische Transformationsmatrix

$$\mathbf{T}_{intr,LP} = \begin{pmatrix} 1094.57 & -22.45 & 418.104 \\ 0 & 1086.88 & 466.499 \\ 0 & 0 & 1 \end{pmatrix}$$

via CalDe/CalLab berechenbar, allerdings handelt es sich bei den Verzeichnungserscheinungen nicht um einfache, radiale Zusammenhänge, wie sie bei der Messkamerakalibrierung auftraten.

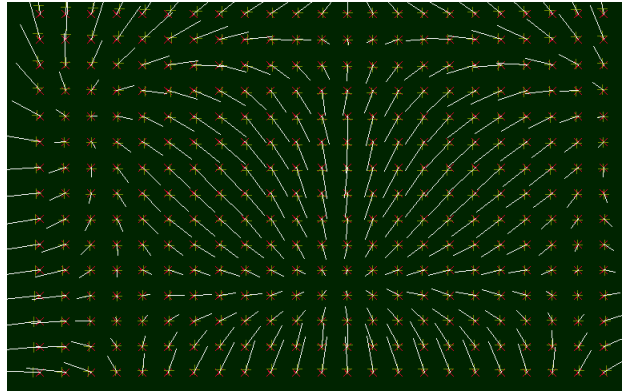


Abbildung 3.13: Betrag in 10-facher Vergrößerung und Orientierung der Verzeichnungserscheinungen des Laserprojektors

Abbildung 3.13 zeigt die tatsächliche Abweichung der Projektion vom idealen Muster in Betrag und Richtung. Da die Residuen der 4 unterschiedlichen Perspektiven nahezu identisch sind, ist die Betrachtung einer einzelnen Perspektive ausreichend. Die Ähnlichkeit der Residuen wird in Kapitel 3.6 bewiesen. Nach Optimierung der intrinsischen Parameter mit Hilfe von CalLab ergibt sich, ohne Ausgleich der radialen Verzeichnungserscheinungen, ein mittlerer Pixelfehler von 3,7 px. Eine exakte Häufigkeitsverteilung des Betrags der Residuen ist Abbildung 3.14 zu entnehmen.

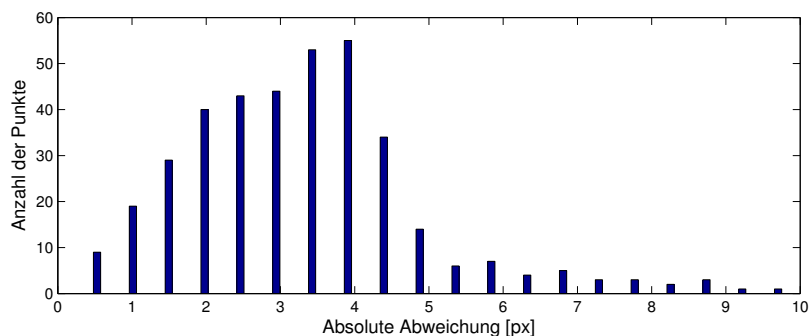


Abbildung 3.14: Absoluter Pixelfehler ohne Ausgleich der Verzeichnungserscheinungen des Laserprojektors

Ein radiales Verzeichnungsmodell kann, wie oben erwähnt, angewendet wer-

den, hat allerdings nur geringfügige Verkleinerung der Abweichungen zur Folge (vgl. Abb. 3.15). Da der innere Aufbau des Laserprojektors stark von

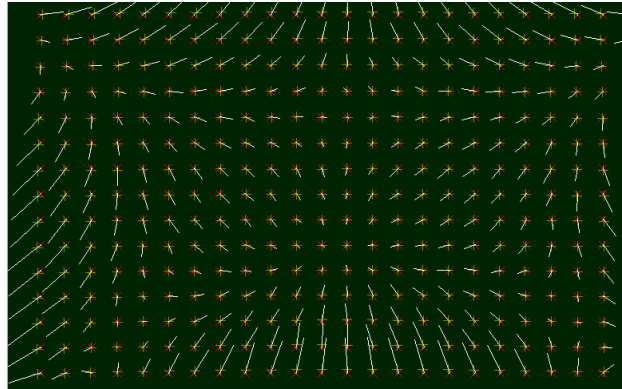


Abbildung 3.15: Betrag in 10-facher Vergrößerung und Orientierung der Verzeichnungserscheinungen des Laserprojektors unter Verwendung des radialen Verzeichnungsmodells

dem einer Kamera abweicht (vgl. Kap. 3.1), war es zu erwarten, dass die Berücksichtigung der radialen Verzerrungen nicht hinreichend genau sind, um das Projektionsverhalten zu optimieren. In der Nähe des BHP ist das Abbildungsverhalten zwar homogener, allerdings kommt es v.a. am linken unteren Bildrand, zu starken Fehlern im Bereich von ca. $6,0 px$.

Um die Abbildungseigenschaften weiter untersuchen zu können, wurden die Kalibrierresiduen in horizontale und vertikale Richtung aufgeteilt.

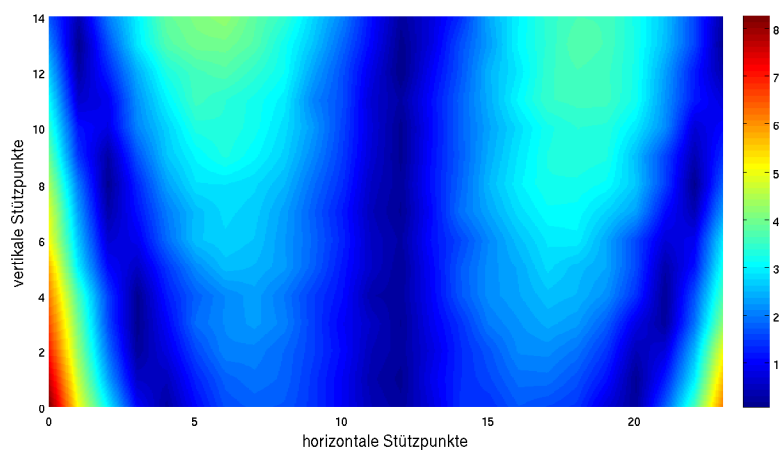


Abbildung 3.16: Interpolierter Betrag der horizontalen Residuen

Die beiden Farbverlaufsbilder 3.16 und 3.17 zeigen den interpolierten Betrag der Verschiebungsvektoren der einzelnen Stützpunkte. Es ist zu beachten, dass der Koordinatenursprung der Projektionen in der linken oberen Ecke liegt, wobei positive x-Werte nach rechts und y-Werte nach unten angetragen sind.

Abbildung 3.16 macht deutlich, dass die Projektion in horizontale Richtung nahezu ideal symmetrisch ist. Sowohl die Senkrechte durch den BHP, als auch die beiden Schrägen am Bildrand werden kaum verzeichnet. Es ist gut zu erkennen, dass die stärkste Abweichung links und rechts außen am unteren Bildrand auftritt.

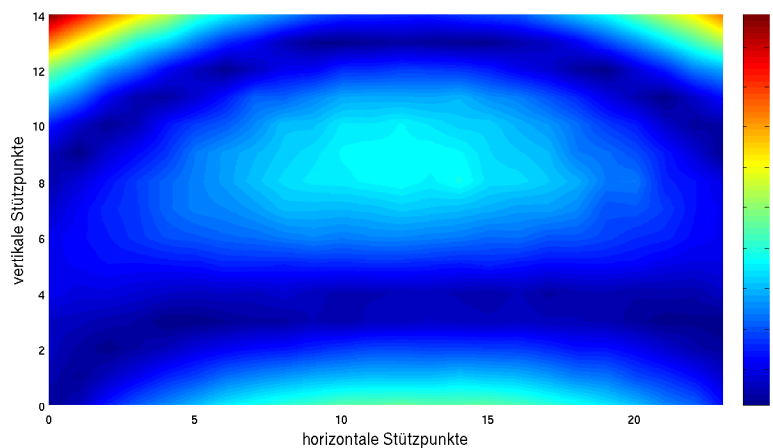


Abbildung 3.17: Interpolierter Betrag der der vertikalen Residuen

Die vertikalen Residuen (vgl. Abb. 3.17) sind in der Bildmitte annähernd monoton. Die stärkste Abweichung tritt links und rechts außen am oberen Bildrand auf. Des Weiteren sind 2 verzeichnungsfreie Bereiche zu erkennen. Der vertikale Versatz zum Bildmittelpunkt wurde vermutlich implementiert, um den Laserprojektor auf Tischen oder anderen horizontalen Gegenständen positionieren zu können.

Trotz des z.T. homogenen Verhaltens konnte kein mathematisches Ersatzmodell gefunden werden das die Verzeichnungen ausreichend genau beschreibt. Die einzelnen Bildabschnitte weisen zwar in sich monotonen Verhalten auf, allerdings ist keine allgemeingültige Systematik erkennbar. Bei einem Aus-

gleich der Abweichungen müssten unterschiedliche Bereiche mit verschiedene Modellen berücksichtigt werden. Da dies in der vorliegenden Arbeit nicht vorgesehen ist, wird auf eine LUT zurückgegriffen. Eine ausführliche Beschreibung der Vorgehensweise zur Nivellierung der Verzeichnungserscheinungen via LUT befindet sich in Kapitel 4.3.3.

3.6 Beweis des konstanten Abbildungsverhaltens unter Verwendung der Pearson-Korrelation

Um die Abbildungseigenschaften des Laserprojektors zu bestimmen, ist es erforderlich, die Residuen der 4 Projektionen auf relative Abweichungen zueinander zu untersuchen. Es muss gezeigt werden, dass das Verhalten des Projektors während der 4 Aufnahmen konstant ist. Hierfür wird die Pearson-Korrelation herangezogen, womit sich der Grad des linearen Zusammenhangs von Punktmengen berechnen lässt. Der empirische Korrelationskoeffizient

$$\mathbf{Kor}_e(x_1, x_2) = \frac{\frac{1}{n} \sum_{i=1}^n (x_{1,i} - \bar{x}_1)(x_{2,i} - \bar{x}_2)}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_{1,i} - \bar{x}_1)^2} \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{2,i} - \bar{x}_2)^2}} \quad (3.4)$$

kann Werte zwischen -1 und +1 annehmen, wobei 1 einen vollständig positiven linearen Zusammenhang bedeutet. Die vorangegangene Beschreibung der Abbildungseigenschaften der 4 Projektionen bezieht sich auf den Mittelwert aus den 4 Residuen am jeweiligen Stützpunkt. Dies war möglich, da die Punktmengen Korrelationskoeffizienten von knapp 1 aufweisen.

3.7 Validierung der Messergebnisse

Die Validierung der Verzeichnungserscheinungen erfolgt mit Hilfe eines 2. Iterationsschrittes. Hierbei wird das pattern in negative Residualrichtung verzerrt und projiziert. Der Quellcode des Musters, welches ausschließlich aus nicht-quadratischen Rechtecken besteht, ist Anhang ?? zu entnehmen. Abbildung 3.18 zeigt den interpolierten Betrag der tatsächlichen Verzeichnungserscheinungen.

Es wurde ein mittlerer Pixelfehler von $3,7 px$ gemessen sowie die folgenden Korrelationskoeffizienten relativ zur Residuenmenge 1:

$$K_{2 \rightarrow 1} = 0,9962 \quad , \quad K_{3 \rightarrow 1} = 0,9962 \quad \text{und} \quad K_{4 \rightarrow 1} = 0,9960$$

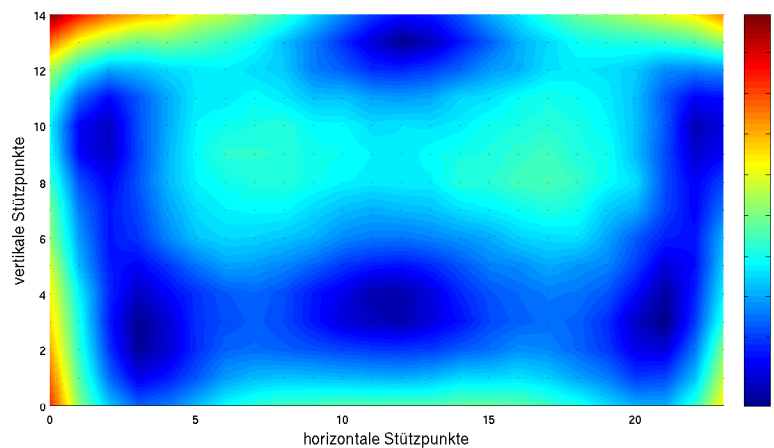


Abbildung 3.18: Bilineare Interpolation der absoluten Residuen bei Projektion des idealen pattern

Im Vergleich dazu ist der interpolierte Betrag der Verzeichnungserscheinungen nach Ausgleich der vorher ermittelten Residuen in Abbildung 3.19 dargestellt. Ein mittlerer Pixelfehler von $0,27 px$ macht deutlich, dass die Abbildungsgenauigkeit stark erhöht wurde. Die nun subpixelgenauen Projektionen weisen die folgenden Korrelationskoeffizienten auf:

$$K_{2 \rightarrow 1} = 0,5924 \quad , \quad K_{3 \rightarrow 1} = 0,5462 \quad \text{und} \quad K_{4 \rightarrow 1} = 0,7438$$

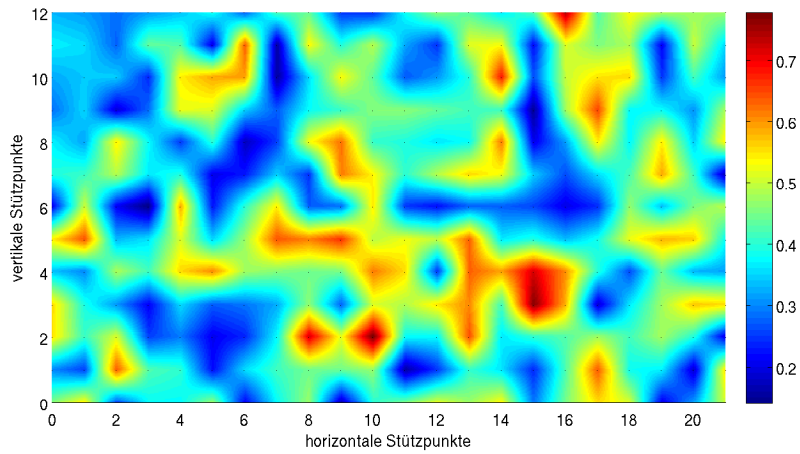


Abbildung 3.19: Bilineare Interpolation der absoluten Residuen bei Projektion des verzerrten pattern.

Sie sind also relativ zueinander stark unterschiedlich, was neben den annähernd normalverteilten Fehlerbetrag (vgl. Abb. 3.20) auf einen nicht systematischen Restfehler hinweist.

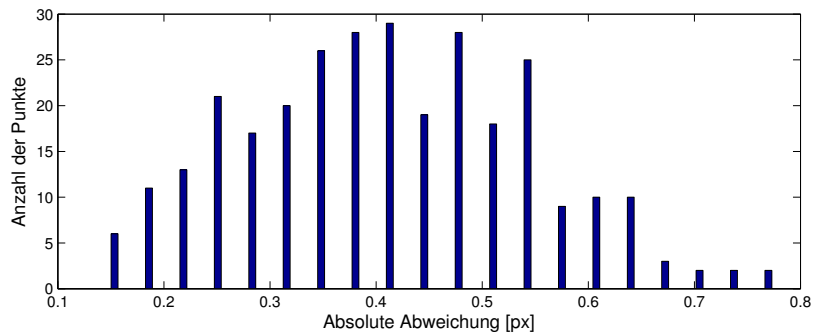


Abbildung 3.20: Absoluter Pixelfehler mit Ausgleich der Verzeichnungserscheinungen des Laserprojektors

Damit ist bewiesen, dass die Verzerrungen des Laserprojektors korrekt ermittelt und ausgeglichen wurden.

3.8 Extrinsische Kalibrierung des Laserprojektors

Bei der extrinsischen Kalibrierung wird die Lage des optischen Zentrums des Projektors relativ zu einem Referenzkoordinatensystem, im vorliegenden Fall das Trackingkoordinatensystem (TKS), ermittelt.

Die 4×4 Transformationsmatrix $\mathbf{T}_{extr,LP}$ lässt sich genau wie die intrinsischen Parameter mit Hilfe von CalDe/CalLab bestimmen. Die Kalibrationssoftware berechnet die Lage des PKS relativ zur Leinwand, sodass unter Verwendung der erfassten Position des TKS die Transformationsmatrix TKS→PKS berechnet werden kann.

Sie beinhaltet die rotatorische Verdrehung \mathbf{R}_{LP} der Koordinatensysteme zueinander sowie eine translatorische Ursprungsverschiebung \mathbf{t}_{LP} und kann, wie in Kap 2.1 beschrieben, als 4×4 Matrix angeschrieben werden:

$$\mathbf{T}_{extr,LP} = \begin{pmatrix} +0.945313 & -0.0251966 & +0.325189 & +69.8954 \\ -0.325971 & -0.0386625 & +0.944589 & -5.20467 \\ -0.011227 & -0.9989350 & -0.044762 & -41.7904 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Eine Aufteilung in Rotation und Translation ist Anhang A.4 zu entnehmen.

Kapitel 4

Software

Ziel der Softwareentwicklung ist die integrierte Berechnung der notwendigen Transformationen und Grafikoperationen auf dem im picop-development kit enthaltenem Beagleboard, welches mit dem OMAP (R) Prozessor über integrierte 3D Grafikprozessierung verfügt. Vorteile sind ein kompaktes und integriertes Komplettsystem und damit der Wegfall eines externen PC zur Grafikberechnung, sowie die leichte Einbindung in Applikationen über Ethernet. Programmierumgebung auf dem Beagleboard ist ein Ubuntu Linux auf ARM Architektur mit OpenGL ES, die Ethernetverbindung zum Remote PC und Tracking System erfolgt über USB (usbnet).

4.1 OpenGL ES Grundlagen

Wie in Kap. 3 erwähnt, handelt es sich bei OpenGL um eine plattformunabhängigen Programmierschnittstelle zur Ansteuerung von Grafikhardware. Sie findet Anwendung in fast allen Plattformen mit 3D-Unterstützung und wird bei embedded systems, wie z.B. iPhone, Playstation 3 oder dem hier verwendeten Beagleboard in der Version OpenGL ES benutzt. [13]

Die embedded systems-2.0 Version unterstützt ausschließlich die "vertex-array-routine", welche hinsichtlich high-performance Anwendungen optimiert wurde. Zur Visualisierung von Koordinaten, sog. "vertices" werden nicht die Punkte einzeln definiert, sondern in einem 1-dimensionalen array gespeichert. [14]

Da OpenGL ES ausschließlich Punkte, Linien und Dreiecke unterstützt, muss also ein Quadrat, wie es zur Erstellung des pattern nötig war, mit einem 1×12 -Array definiert werden.

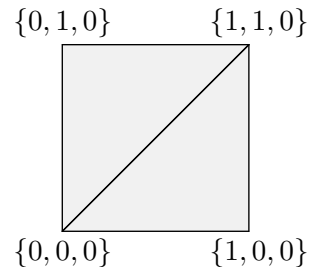


Abbildung 4.1: Erstellung eines Quadrats mit Hilfe von OpenGL ES 2.0

Die Visualisierung des in Abbildung 4.1 gezeigten Quadrats im Folgenden kurz anhand eines Listings dargestellt:

```

GLfloat quad[] = {0,0,0,0,1,0,1,1,0,0,0,0,1,1,0,1,0,0}
while (quit != 1)
{
    glClear( GL_COLOR_BUFFER_BIT);
    glEnableClientState( GL_VERTEX_ARRAY);

    glColor4f(0.5f, 0.5f, 0.5f, 0.1f);
    glVertexPointer(3, GL_FLOAT, 0, quad);
    glDrawArrays(GL_TRIANGLES, 0, 12);

    glDisableClientState(GL_VERTEX_ARRAY);

    eglSwapBuffers(*eglDisplay, *eglSurface);
}

```

Nach der Initialisierung des arrays “quad“ werden in einer while-Schleife verschiedene OpenGL und EGL Befehle ausgeführt, welche in den Tabellen 4.1 und 4.2 erklärt und z.T. im Folgenden näher beschrieben werden.

Mit `glVertexPointer(3, GL_FLOAT, 0, quad)` lässt sich die Lage (0) und das Datenformat (GL_FLOAT) des Arrays (quad) definieren. Die erste Variable gibt an, wieviele Koordinaten ein Vertex, also ein Eckpunkt, besitzt.

Die Funktion `glDrawArrays(GL_TRIANGLES, 0, 12)` definiert die Primiti-

ve, also die vertices hinsichtlich Art(GL_TRIANGLES), Position im array (0) und Anzahl (12).

Tabelle 4.1: Die wichtigsten OpenGL ES Befehle nach [14]

OpenGL ES Befehle	
<code>glClear()</code>	Initialisierung des Buffers
<code>glColor4(<i>r, g, b, α</i>)</code>	Definition der Rot-, Grün- und Blauanteile sowie der Intensität
<code>glDisableClientState()</code>	Deaktivierung der Schreibzugriffe des Vertex-Arrays
<code>glDrawArrays()</code>	Definition der Primitiven, die im Anschluss konstituiert werden
<code>glEnableClientState()</code>	Aktivierung der Schreibzugriffe des Vertex-Arrays für die Aufrufe von <code>glDrawArrays</code>
<code>glRotate(<i>angle, x, y, z</i>)</code>	Multiplikation einer Matrix mit einer Rotationsmatrix / Es werden alle Punkte um den Winkel <i>angle</i> in mathematisch positive Richtung um die durch <i>x,y</i> und <i>z</i> definierte Rotationsachse gedreht
<code>glTranslate(<i>x, y, z</i>)</code>	Translatorische Verschiebung des Koordinatenursprungs in den Punkt (<i>x, y, z</i>)
<code>glVertexPointer()</code>	Beschreibung von Lage und Datenformat des Arrays, das gerendert werden soll

4.2 Simulation des Autoprojektors

Um das Verhalten des Autoprojektors besser untersuchen zu können und mathematische Überlegungen hinsichtlich Abbildungsverhalten und perspektivischer Verzeichnung zu validieren, wurde der Versuchsstand unter Verwendung von OpenGL simuliert.

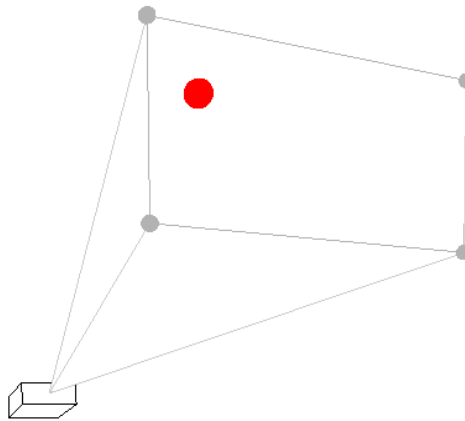


Abbildung 4.2: Simulation des Autoprojeter aus Sicht der transparenten Leinwand

Der Laserprojektor wird mit einem Quader symbolisiert, der sich mit einer keyboard-function um x -, y -, und z - Achse drehen lässt, wobei x der optischen Achse des Projektors entspricht und z der vertikalen Hochachse. Die Projektionspyramide ist über den Koordinatenursprung, welcher im optischen Zentrum liegt, sowie den 4 Schnittpunkte mit einer senkrechten Ebene (der Leinwand) im WKS definiert. Sobald sich der im WKS gegebene Punkt innerhalb der 4 grauen Schnittpunkte von Projektionspyramide und Leinwand befindet, wird er rot markiert.

Das Listing zur Berechnung der 4 Schnittpunkte befindet sich in Anhang A.7. Es ist zu beachten, dass der code OpenGL Befehle enthält, die in OpenGL ES nicht zulässig sind.

4.3 Software des Autoprojektors

Das Programm zur Umsetzung des Laserprojektors als Autoprojektor besteht aus folgenden 3 Schritten, die im Anschluss näher beschrieben werden:

- Interface Trackingsystem und Beagleboard:
Die Lagedaten des Trackingsystems werden vom ART System zum Remote-PC geschickt und von dort in Form einer 4×4 Transformationsmatrix an das Beagleboard weitergeleitet.
- Koordinatentransformation:
Berechnung der 2-D Position der gegebenen Koordinaten in Abhängigkeit von der empfangenen Trackingdaten.
- Bilineare Interpolation:
Ausgleich der Verzeichnungserscheinungen unter Verwendung einer LUT.

4.3.1 Interfaces Trackingsystem und Beagleboard

Abbildung 4.3 zeigt die grafische Benutzeroberfläche der Trackingsoftware *DTrack2* inklusive Visualisierung der erfassten Markerkugeln beider Kameras. Die Werte der 6 Freiheitsgrade des TKS werden direkt angezeigt und können mit der in Anhang A.5 hinterlegten Subroutine und unter Verwendung der DLR-internen Klasse "ARTCom" zur Kommunikation mit dem Trackingsystem ausgelesen werden. Die Lage des TKS relativ zum WKS wird in einer 4×4 Matrix via Ethernet übergeben, wobei sich Trackingsystem, Remote-PC und Beagleboard im gleichen Subnetz befinden. Die relevanten IP-Adressen sind Anhang A.6 zu entnehmen.



Abbildung 4.3: Benutzeroberfläche der DTrack-Software inklusive Visualisierung der erfassten Markerkugeln und Anzeige der Lagedaten

4.3.2 Koordinatentransformation

Um einen Punkt im WKS bei sich bewegendem Laserprojektor zu markieren, ist es nötig, die im 3-dimensionalen Raum gegebenen Koordinaten auf die 2-D BE zu transformieren.

Die Umrechnung erfolgt in 3 Subroutinen welche in der draw-funktion in folgender Reihenfolge aufgerufen werden:

- Weltkoordinatensystem \rightarrow Trackinkoordinatensystem ($\mathbf{T}_{WKS \rightarrow TKS}$)
- Trackingkoordinatensystem \rightarrow Projektorkoordinatensystem ($\mathbf{T}_{TKS \rightarrow PKS}$)
- Projektorkoordinatensystem \rightarrow Bildebene ($\mathbf{T}_{PKS \rightarrow BE}$)

Als erstes ist das WKS in der Software des Trackingsystems mit einem Kalibrationswinkel (vgl. Abb. 4.4) zu definieren. Das in Kapitel 4.3.1 beschriebene Interface von DTrack liefert kontinuierlich eine homogene 4×4 Transformationsmatrix $\mathbf{T}_{WKS \rightarrow TKS}$ womit die Position des TKS im WKS erfasst wird. Das TKS wird mit 4 Markern aufgespannt und ist auf der Grundplatte

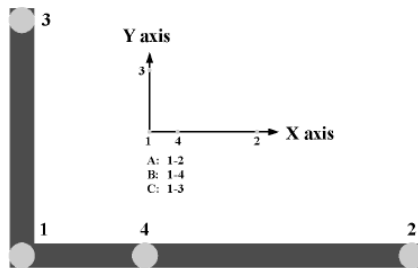


Abbildung 4.4: Kalibrationswinkel zur Definition des Weltkoordinatensystems

neben dem Laserprojektor montiert.

Die Transformationsmatrix $\mathbf{T}_{TKS \rightarrow PKS}$ wurde im Zuge der extrinsischen Kalibration ermittelt und bleibt konstant, ebenso wie die Umrechnung des im WKS gegebenen und anzuzeigenden Punktes auf der BE. Berechnet werden muss die Transformation zwischen PKS und des beleuchteten Punktes P $\mathbf{T}_{TKS \rightarrow P}$.

Abbildung 4.5 zeigt die Koordinatensysteme inkl. Visualisierung der Transformationen. Konstante Matrizen sind grün dargestellt, variable rot.

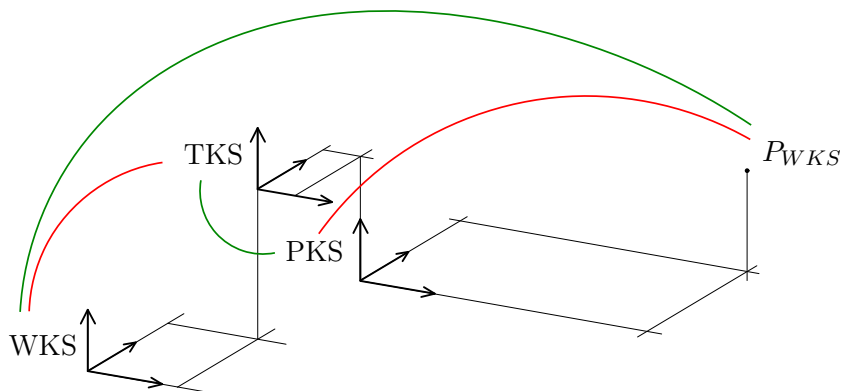


Abbildung 4.5: Koordinatensysteme und deren Transformation. Konstanten Matrizen wurden grün dargestellt, variable rot.

4.3.3 Bilineare Interpolation

Im letzten Schritt müssen die in der Kalibrierung ermittelten Verzerrungserscheinungen ausgeglichen werden. Wie in Kapitel 3.5 beschrieben, lassen sie sich nicht durch ein einfaches radiales Verzerrungsmodell nivellieren. Sie können jedoch mit Hilfe einer LUT, bestehend aus den idealen Koordinaten, also den Stützpunkten (Pixelkoordinaten / gespeichert in Spalte 1 und 2 des arrays "residuals"), und den ermittelten Residuen (Pixelkoordinaten / gespeichert in Spalte 3 und 4 des arrays "residuals") angenähert werden. Um die Verschiebung der einzelnen Pixel zu beschreiben wird zwischen den Stützpunkten interpoliert. Aus Gründen der Übersichtlichkeit visualisieren die Abbildungen 4.6 und 4.7 nur die Verzeichnung in horizontale Richtung. Als erstes werden die Stützpunkte und die Residuen, welche im Zuge der Laserprojektorkalibrierung ermittelt wurden, aus Textfiles eingelesen und in ein 375×4 array gespeichert (vgl. Listing 4.1). Die 375 Punkte ergeben sich aus 24 horizontalen \times 15 vertikalen Eckpunkten und sind spaltenweise von links oben nach rechts unten gespeichert.

```
float p, residuals [375][4];
FILE * pFile;
pFile = fopen("resid1.txt", "r");
rewind(pFile);
for (int i = 0; i < 375; i++)
{
    fscanf(pFile, "%f\n", &p);
    (*residuals)[i][0] = p;
}
fclose(pFile);
```

Listing 4.1: Subroutine read_residuals zur Initialisierung der ersten Spalte des Arrays residuals

Der Interpolationsalgorithmus ermittelt, basierend auf gegebenen Pixelkoordinaten $p(u, v)$ die 4 nächstgelegenen Stützpunkte $Sp_{1...4}$ (vgl. Listing 4.2) sowie die zugehörigen Residuen $R_{1...4}$ und interpoliert die Verzerrungsvektoren linear (siehe Abb. 4.6).

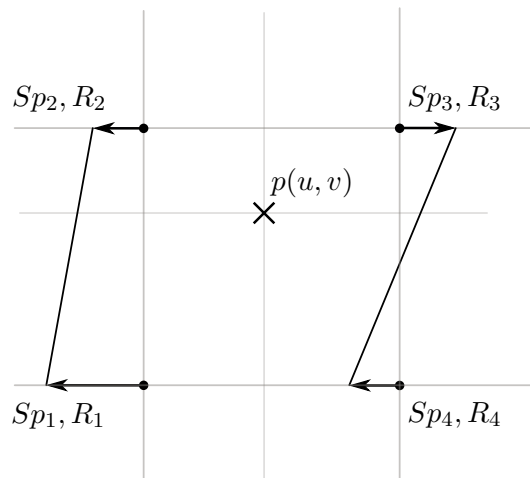


Abbildung 4.6: Interpolation der Verzerrungen in horizontale Richtung:
Ermittlung der nächstgelegenen Stützpunkte mit zugehörigen
Residuen

```

for (int i = 0; i < 375; i++) {
if (u - 32 < (*residuals)[i][0] and u >= (*residuals)[i][0] ...
and v - 32 < (*residuals)[i][1] and v >= (*residuals)[i][1])
{
    Sp_1    = (*residuals)[i][0];
    Sp_2    = (*residuals)[i + 15][0];
    Sp_3    = (*residuals)[i + 14][0];
    Sp_4    = (*residuals)[i - 1][0];
    R_1     = (*residuals)[i][2];
    R_2     = (*residuals)[i + 15][2];
    R_3     = (*residuals)[i + 14][2];
    R_4     = (*residuals)[i - 1][2];
}
}

```

Listing 4.2: Ermittlung der nächstgelegenen Stützpunkte mit zugehörigen
Residuen in horizontale Richtung

Als nächstes erfolgt die Gewichtung der interpolierten Residuen in Abhängigkeit von ihrer Entfernung zum Bildpunkt. Bei der Visualisierung in Abb. 4.7 wurden negative horizontale Verschiebungen nach unten angetragen und positive nach oben.

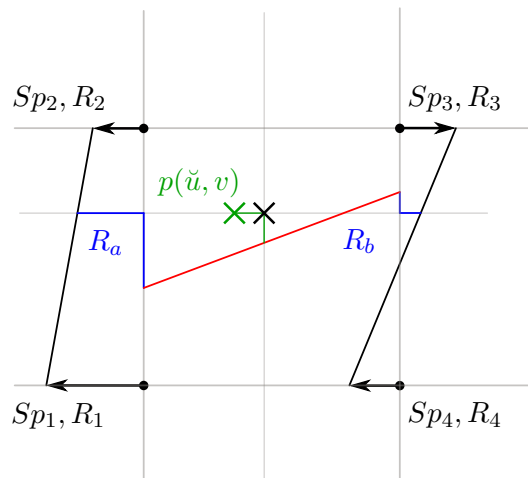


Abbildung 4.7: Gewichtung der interpolierten Residuen in Abhängigkeit von ihrer Entfernung zum Bildpunkt

Bei einer Stützpunktentfernung von 32 px in horizontale und vertikale Richtung kann mit dem Listing 4.3 der Punkt um den ermittelten Betrag verschoben werden.

```

{
    Ra = R_1 + ((R_4 - R_1) / 32) * (v - Ry_1);
    Rb = R_2 + ((R_3 - R_2) / 32) * (v - Ry_1);
    R_interp = (Ra + ((Rb - Ra) / 32) * (u - Sp1));

    coords->x_interp = u - rx_interp;
}

```

Listing 4.3: bilineare Interpolation der ermittelten Residuen

4.3.4 Window System, Interface und Devilspie

Als Anzeigeprotokoll wird das X11-Window System verwendet mit dem xfce-Window Manager für ubuntu Linux. Des weiteren dient EGL als Interface zwischen dem X11 und OpenGL ES. Es übernimmt Funktionen wie Rendering, Synchronisation, etc. Eine kurze Beschreibung der Verwendeten X11 und EGL Befehle, welche zum Verständnis der Listings nötig sind, ist Tabelle 4.3 und 4.2 zu entnehmen.

Tabelle 4.2: Die wichtigsten EGL Funktionen [14]

EGL Funktionen	
eglGetDisplay	Erfassung der EGL display-connection des Anzeigeprotokolls
eglInitialize	Initialisierung der erfassten display-connection
eglChooseConfig	Definition der Framebuffereinstellungen
eglCreateWindowSurface	Erstellung eines EGL-Fensters
eglSwapBuffers	Beschreibung des display mit dem aktuellen EGL buffer
eglTerminate	Abbruch der EGL display-connection

Tabelle 4.3: Die wichtigsten X-11 Funktionen [12]

X-11 Funktionen	
XOpenDisplay	Herstellung der Verbindung zum X-Server
XCreateWindow	Erstellung des X-11 Fensters
XStoreName	Definition des Fensternamens

Kapitel 5

Genauigkeitsuntersuchung

Um Aussagen über die Genauigkeit des Autopointers machen zu können wird ein Eckpunkt eines pattern projiziert, welcher mit einer Messkamera aufgenommen und CalDe detektiert wird. Über den homographischen Zusammenhang wird der Punkt in Weltkoordinaten umgerechnet.

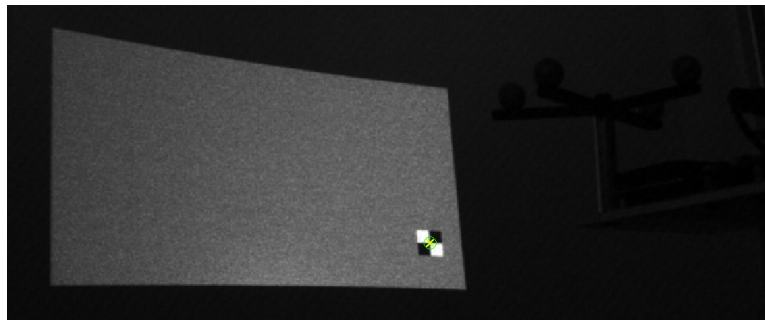


Abbildung 5.1: Detektion des projizierten Eckpunkts via CalDe

Wie in Abbildung 5.5 zu erkennen, ist die grundsätzliche Funktion des Autopointers gewährleistet. Bei sich bewegendem Laserprojektor verändert sich die Position des Eckpunkts nur minimal. Als Gründe für die Abweichung sind verschiedene Einflussfaktoren zu nennen, welche sich in statische und dynamische Faktoren unterteilen und gewichten lassen. Es ist zu beachten, dass auf Basis der folgenden Messreihen Tendenzen angegeben und Einflüsse in Relation zueinander betrachtet werden können. Um jedoch eine exaktere Aussagen über die Abbildungsgenauigkeit des Autopointers zu machen

wären weitere Messungen und ein umfangreicherer Versuchsstand erforderlich.

5.1 Statische Genauigkeit

Um die statische Abbildungsgenauigkeit zu untersuchen wurden Projektionen aus unterschiedlichen Perspektiven vermessen. Sie befinden sich in einem Arbeitsabstand von ca. 40 cm und beinhalten sowohl translatorische, als auch rotatorische Veränderungen. Es wurden 6 Messreihen á 6 Aufnahmen durchgeführt und ausgewertet.

Abbildung 5.2 zeigt die bidirektionale Abweichung (blau) von den gewünschten Projektionskoordinaten (rot). Der Fehler beträgt horizontal ca. $\pm 1,5$ mm und vertikal ca. $\pm 2,5$ mm.

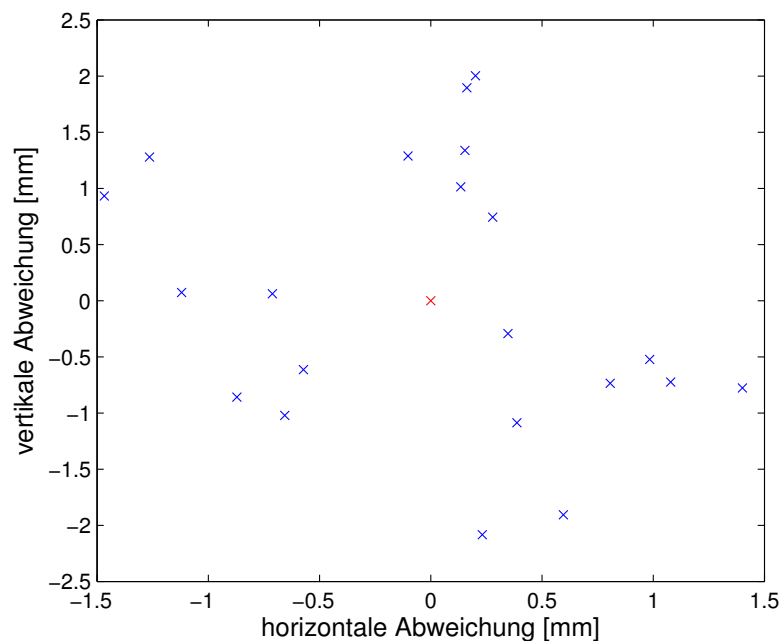


Abbildung 5.2: Gewünschte Projektionskoordinaten (rot) und Abweichungen (blau)

Einfluss des Trackingsystems

Den größten Einfluss auf die erreichte Abbildungsgenauigkeit des Autopointers hat das Trackingsystem. Laut Herstellerangaben können die Marker mit einer Abweichung von $\pm 0,5$ mm detektiert werden. Da allerdings keine Aussagen über Markerzustand oder -größe gemacht werden, ist bei dem verwendeten Target mit sehr kleinen Markerkugeln eine zusätzliche Toleranz von $\pm 0,1$ mm zu berücksichtigen, sodass sich ein theoretischer Gesamtfehler von ca. $\pm 0,6$ mm ergibt. Zur Veranschaulichung des Fehlers wird eine 1-dimensionale rotatorische Bewegung betrachtet, da sie sich deutlich stärker auf die Genauigkeit der Abbildung auswirkt als eine translatorische Veränderung.

Ein Versatz von $\pm 0,6$ mm pro Marker führt bei einem Markerabstand von 90 mm mit $\arctan(\frac{1,2}{90})$ zu einem maximalen Fehler von $\pm 0,76^\circ$ um die Rotationsachse, ohne Berücksichtigung des Abstands zwischen TKS und PKS, welcher ca. 50 mm beträgt und daher in guter Näherung vernachlässigt werden kann (vgl. Abb. 5.3). In der Bildebene kommt es so mit $\tan(0,76^\circ) \cdot 400$ mm zu einem theoretischen maximalen Fehler im BHP von $\pm 5,3$ mm. Es ist zu beachten, dass es sich um einen theoretischen Maximalwert handelt, der mit einer sinnvollen (siehe Abb. 5.4) Anordnung von Trackingkameras und Targetstern stark reduziert werden kann.

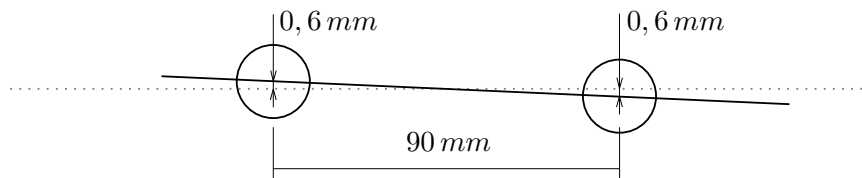


Abbildung 5.3: Seitenansicht von 2 Markerkugeln des Trackingtargets mit Visualisierung der maximalen Abweichung von der Horizontalen

Abbildung 5.4 zeigt die Untersuchung des statistischen Rauschens. Es sind die Abweichungen vom Mittelwert bei einem Winkel zwischen Trackingkamera und TKS von ca. 0° (oben) und 45° (unten) dargestellt. Wie zu erwarten verschlechtert sich das Tracking deutlich sobald die Marker in der gleichen Ebene wie die beiden Kameras liegen.

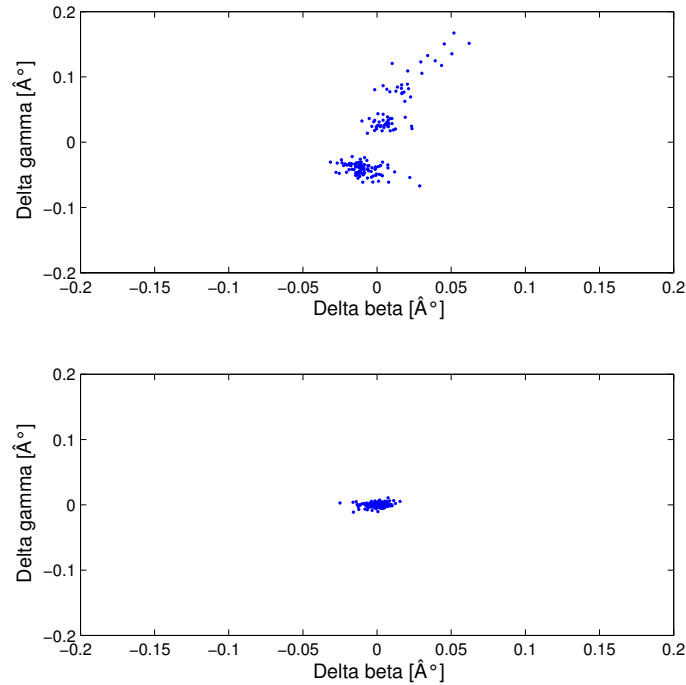


Abbildung 5.4: Gegenüberstellung der Abweichungen vom Mittelwert bei einem Winkel zwischen Trackingkamera und TKS von ca. 0° (oben) und 45° (unten)

Verwendung der Homographie

Die Verwendung der Messkamera und des homographischen Zusammenhangs ist eine schnelle und sehr genaue Methode zur Vermessung der Projektion, allerdings ist ein geringer systematischer Restfehler nur sehr schwer zu erkennen. Die 4 Punkte auf der BE wurden unter Verwendung von Wasserwaage, Messschieber und Schwerkraftsloot orthogonal zueinander positioniert und können so mit einer Genauigkeit von ca. $\pm 0,2$ mm definiert werden. Da sich ein möglicher Fehler über die Breite bzw. Höhe der Projektion erstreckt, hat er kaum Auswirkungen auf die Untersuchungen.

Aus den symmetrischen Ergebnissen der Laserprojektorkalibrierung kann des weiteren gefolgert werden, dass die Punkte keinen signifikanten trapez- oder parallelogrammförmigen Versatz vorweisen.

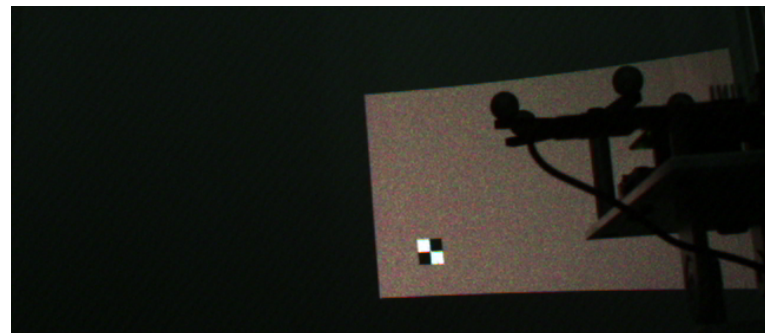
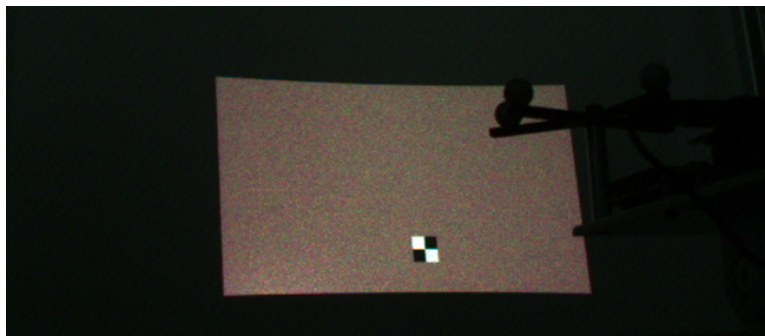
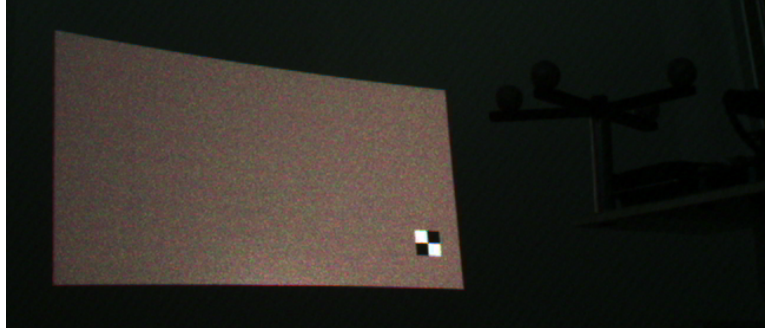


Abbildung 5.5: Projektionen des Autopointers bei einem Arbeitsabstand von 40 cm

5.2 Dynamische Genauigkeit

Da sich mit dem in Abbildung 3.1 dargestellten Versuchstand keine dynamischen Betriebszustände simulieren lassen, also keine Antriebe o.ä. vorhanden sind, können ausschließlich qualitative Aussagen über das Abbildungsverhalten des Autopointers gemacht werden.

Die Einflüsse der einzelnen Komponenten auf die "Trägheit" des Systems ist in Tabelle 5.1 dargestellt. Die Framerate des Laserprojektors ist mit

Tabelle 5.1: wichtige Frequenzen zur Untersuchung des dynamischen Abbildungsverhaltens

Komponente/Programmteil	
Framerate Laserprojektor	60 Hz
Framerate Messkamera	48 Hz
Frequenz Trackingsystem	max. 60 Hz
Frequenz draw-function	ca. 40 Hz
tatsächlich erreichte redraw-Frequenz	ca. 20 Hz

60 Hz gegeben, was mit der Messkamera überprüft wurde. Eine Erhöhung der Projektor-framerate wäre laut Handbuch möglich, indem die Strahlungsintensität gesenkt würde. Allerdings lässt sich das dynamische Verhalten auf diese Weise nicht verbessern, da mit 60 Hz die maximale Trackingsystemfrequenz erreicht ist.

Um die programminternen Abläufe untersuchen zu können wurden verschiedene timer definiert und ausgewertet. Die Frequenz der draw-function, in die alle Rechenschritte und Transformationen implementiert wurden, beträgt 40 Hz. Trotzdem wird die Position des Punktes nur alle 0,05s aktualisiert, also mit einer Bildrate von 20 Hz. Die Verzögerung lässt sich vermutlich auf die egl-Funktion *swapbuffer* zurückführen, in der der Buffer aktualisiert wird.

Die Latenz des Gesamtsystems ist nur sehr schwer zu untersuchen, da die nötigen Messungen eine konstante Rotationsgeschwindigkeit erfordern und vom statischen Fehler überlagert werden. In grober Näherung verschiebt sich die Projektion bei einer manuellen Rotation von ca. $10 \frac{\circ}{s}$ bei einem Arbeits-

abstand von 400 mm um ca. 8 mm. Mit $\arctan \frac{8}{400} = 1,15^\circ$ und $\frac{1,15^\circ}{10 \frac{\text{cm}}{\text{s}}}$ ergibt sich eine Latenz von ca. 0,1 s.

Kapitel 6

Zusammenfassung

In der vorliegenden Arbeit wurde ein Pico Laserprojektor als Autopointer kalibriert, programmiert und hinsichtlich Projektionsgenauigkeit untersucht. Die Ergebnisse zeigen, dass das Ziel, nämlich die ortsfeste Projektion eines Punktes bei sich bewegendem Laserprojektor, erreicht wurde.

Mit der Abbildungsgenauigkeit von ca. $\pm 2,5$ mm in einem Arbeitsabstand 40 cm wäre ein Einsatz als Montageassistenzsystem möglich. So könnten, unter der Annahme, dass Trackingsystem und Laserprojektor in ausreichend großem Winkel zueinander positioniert wurden, Werkstücke markiert oder nummeriert werden. Für eine Verwendung im operativen Bereich sind verschiedene Maßnahmen zur Erhöhung der Genauigkeit nötig. Eine sehr einfache Methode ist die Veränderung des Trackingtargets. Ein größerer Stern mit Markern, die sich nicht in einer Ebene befinden verringert v.a. das in Abbildung 5.4 visualisierte Fehlverhalten. Da sich Abweichungen beim Tracking sehr stark auf das Projektionsverhalten des Autopointers auswirken, sollte das Trackingsystem um mehrere Kameras erweitert werden.

Es hat sich gezeigt, dass die implementierte Messkamera, zusammen mit der Detektion via CalDe und der Umrechnung mittels homographischem Zusammenhang ein sehr effektives Mittel zur Untersuchung der Abbildungseigenschaften ist. Die so gewonnenen Kalibrationsergebnisse sind Grundlage für jede weitere Verwendung des Projektors. Ein mathematisches Ersatzmodell könnte aufgrund der systematischen Fehlerverteilung ermittelt werden. Allerdings ist es sehr unwahrscheinlich, dass sich durch das Ersetzen der

LUT inkl. bilinearer Interpolation das statische oder dynamische Verhalten signifikant verbessert.

Der Versuchsaufbau ist für die vorangegangenen Messungen angemessen, sollte aber zur Untersuchung der Dynamik des Autopointers um Antriebe und einer Kamera mit höherer Bildrate erweitert werden. Weitere Schritte wären die Erweiterung der Autopointerfunktion um komplexe Grafik, sowie Projektionen auf gekrümmten (bekannt) Oberflächen. Eine Beschleunigung der Grafikprozessierung, ggf. auf schnellerer Hardware, würde das dynamische Verhalten verbessern.

Anhang A

A.1 Radiale Linsenverzeichnung

```
function undist = undistortion(distorted ,A,k1,k2)
und_tmp = distorted;
% inits
distortion_error = [ones(size(distorted))];
i=1;
optim_done=0;

while (~optim_done)
    u = und_tmp(:,1) - A(1,3);
    v = und_tmp(:,2) - A(2,3);
    y = v./A(2,2);
    x2 = ( (u-A(1,2).*y) ./ A(1,1) ).^2;
    y2 = y.^2;
    dis_tmp = und_tmp + [ k1.*u.*(x2+y2) + k2.*u.*(x2+y2).^2 ...
                        k1.*v.*(x2+y2) + k2.*v.*(x2+y2).^2];
    distortion_error = dis_tmp - distorted;
    und_tmp = und_tmp - distortion_error;

% Addion der maximalen Absolutwerte von x und y
% dis_error < 1e-1 pixels
if (sum(max(abs(distortion_error)).^2)<1e-2);
    optim_done=1;
    res_dis_error=sqrt(sum(max(distortion_error).^2));
```

```

    else
        %i=i+1 % 4 iterations should suffice
    end
end

```

```

assignin('base', 'undistorted', und_tmp);
undist = und_tmp;

```

A.2 Ermittlung der Homographiematrix

```

% Quellcode zur Ermittlung der Homographiematrix

```

```

function h = calculation_homography()

```

```

% Manuell vermessene Leinwandkoordinaten

```

```

P_lw=[0,245; 0,0; 672,0; 672,245];

```

```

% Via Calde ermittelte intrinsische Kameraparameter

```

```

A =[
        991.5740      0  381.5470
           0  991.5740  285.9140
           0          0    1.0000
    ];

```

```

k1=-0.1224;

```

```

k2= 0.1445;

```

```

% Korrespondierende Pixelkoordinaten

```

```

P=[
        80.2822      339.515
        99.6634      103.219
        694.825      162.930
        724.678      365.574
    ];

```

```

assignin('base', 'A', A);

```

```

assignin('base', 'k1', k1);
assignin('base', 'k2', k2);

% undistortion der 4 Pixelkoordinaten
% siehe "Radiale Linsenverzeichnung"

P_px=und_tmp

% Homographiematrix – siehe "calculation homography.m"
L=[...]

% Singulaerwertdivision
k=null(L);
H=[k(1) k(2) k(3); k(4) k(5) k(6); k(7) k(8) k(9)];
assignin('base', 'H', H);

% Ermittelte Homographiematrix H
H=[
    -0.0066736512    -0.0000531591    +0.6951482335;
    +0.0007898954    -0.0078740187    +0.7187573186;
    +0.0000020195    -0.0000025714    -0.0070289401]

```

A.3 Ergebnisse der Kalibrierung der Messkamera

% Intrinsischer Transformationsmatrix:

```
Ti(cam)=[991.574      0 381.547;  
          0 991.574 285.914;  
          0          0      1]
```

% Radiale Verzeichnungskoeffizienten

```
k1= -0.122418  
k2= +0.144535
```

A.4 Ergebnisse der Kalibrierung des Laserprojektors

% Intrinsischen Kalibrierungsergebnisse

Siehe Resid_1...4.txt

% Extrinsische Kalibrierungsergebnisse

```
R(LP)= [+0.945313  -0.025196  0.325189;  
        -0.325971  -0.038662  0.944589;  
        -0.011227  -0.998935  -0.04476]
```

```
t(LP)= [69.8954;  
        -5.2047;  
        -41.7904]
```

```
T(LP)= [+0.945313  -0.025196  0.325189 69.8954;  
        -0.325971  -0.038662  0.944589 -5.2047;  
        -0.011227  -0.998935  -0.04476 -41.790;  
          0          0          0      1]
```


A.5 ARTCom

```
#include <ART/artcom.h>
#include <iostream>
#include <stdio.h>
#include <math.h>
#include "prg.h"

using namespace std;

//main
int main(int argc, char **argv) {

    //create sensor object
    ARTCom dtrack;
    ARTCom::TrackingData recv;

    // connect to ART System with udp port 52000
    if (!dtrack.openUDP(5000))
    {
        cerr << "ERROR: _Can't_open_Port_to_ART_\n" <<endl;
        dtrack.closeUDP();
        return -1;
    }

    // acquire 10 frames — 30ms timeout —
    size_t i = 0;

    while (i==0)
    {
        if (dtrack.receiveAscii(recv, 30000)){

            float TM11=recv.body.at(0).rot[0][0];
            float TM12=recv.body.at(0).rot[0][1];
            float TM13=recv.body.at(0).rot[0][2];
```

```

float TM21=recv.body.at(0).rot[1][0];
float TM22=recv.body.at(0).rot[1][1];
float TM23=recv.body.at(0).rot[1][2];

float TM31=recv.body.at(0).rot[2][0];
float TM32=recv.body.at(0).rot[2][1];
float TM33=recv.body.at(0).rot[2][2];

//Speichern der TM-Matrix
FILE * TM_matrix;
TM_matrix= fopen ("trackingvalues_TM_matrix.csv","w");
fprintf(TM_matrix, "%4.8f\t%4.8f\t... ,_TM11, TM12, TM13, x, ... )
fclose(TM_matrix);
}

//close_connection
dtrack.closeUDP();
return_0;
}

```

A.6 Interface Trackingsystem und Beagleboard

IP-Adressen im lokalen Netzwerk

Remote-PC	192.168.132.1
Beagleboard	192.168.132.2
Trackingsystem	192.168.132.100

A.7 Simulation des Autopointers

```
Schnittpunkt_c1 = dx/(cos((beta-6)*f)*cos((gamma_1+3)*f))
                 -(dx/(cos(6*f)*cos(3*f))-dx);
Schnittpunkt_c2 = dx/(cos((beta+6)*f)*cos((gamma_1+3)*f))
                 -(dx/(cos(6*f)*cos(3*f))-dx);
Schnittpunkt_c3 = dx/(cos((beta+6)*f)*cos((gamma_1-3)*f))
                 -(dx/(cos(6*f)*cos(3*f))-dx);
Schnittpunkt_c4 = dx/(cos((beta-6)*f)*cos((gamma_1-3)*f))
                 -(dx/(cos(6*f)*cos(3*f))-dx);
```

```
glPushMatrix();
glColor3f(0.8, 0.8, 0.8);
glBegin(GL_LINES);
glVertex3f( 0.0, 0.0, 0.0);
glVertex3f(Schnittpunkt_c1, Schnittpunkt_c1*tan(3*f),
Schnittpunkt_c1*tan(6*f));
glEnd();
glBegin(GL_LINES);
glVertex3f( 0.0, 0.0, 0.0);
glVertex3f(Schnittpunkt_c2, Schnittpunkt_c2*tan(3*f),
Schnittpunkt_c2*tan(-6*f));
glEnd();
glBegin(GL_LINES);
glVertex3f( 0.0, 0.0, 0.0);
glVertex3f(Schnittpunkt_c3, Schnittpunkt_c3*tan(-3*f),
Schnittpunkt_c3*tan(-6*f));
glEnd();
glBegin(GL_LINES);
glVertex3f( 0.0, 0.0, 0.0);
glVertex3f(Schnittpunkt_c4, Schnittpunkt_c4*tan(-3*f),
Schnittpunkt_c4*tan(6*f));
glEnd();
glPopMatrix();
```

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig, ohne fremde Hilfe angefertigt worden ist. Inhalte und Passagen, die aus fremden Quellen stammen und direkt oder indirekt übernommen worden sind, wurden als solche kenntlich gemacht. Ferner versichere ich, dass ich keine andere, außer der im Literaturverzeichnis angegebenen Literatur verwendet habe. Diese Versicherung bezieht sich sowohl auf Textinhalte sowie alle enthaltenden Abbildungen, Skizzen und Tabellen.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

München, 15. Juni

(Stefan Sollinger)

Literaturverzeichnis

- [1] ART GmbH: System user manual ArtTrack und DTrack, Version 2.2, advanced realtime tracking GmbH,2009.
- [2] Craig, J. J.: Introduction to robotics: mechanics and control, 2.Aufl. Addison Wesley, 1989.
- [3] Freeman, M./ Champion M. / Madhavan S.: Scanned Laser Pico projectors: Seeing the Big Picture (with a small device), Micorvision manual, 2008.
- [4] Hirschmüller, H.: Stereo Vision Based Mapping and Immediate Virtual Walkthroughs, school of computing De Montfort University, 2003
- [5] Leppla, S.: Kalibrierungsverfahren für die Erweiterte Realität, Seminar Intelligente Industrieroboter, TU Karlsruhe, 2007.
- [6] Rodehorst, V.: Photogrammetrische 3D-Rekonstruktion im Nahbereich durch Auto-Kalibrierung mit projektiver Geometrie, Wissenschaftlicher Verlag Berlin, 2004.
- [7] Schröder G./ Treiber H.: Technische Optik, 10.Aufl. Vogel, 2007.
- [8] Strobl, K. H.: On the Issue of Camera Calibration with Narrow Angular Field of View. The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 309-315, 2009.
- [9] Tsai, R. Y.: A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation, vol. 3,no. 4, S. 323-344, 1987.

- [10] Wolf, P. R.: Elements of Photogrammetry, McGraw-Hill book company, 1983.
- [11] CalDe: <http://www.dlr.de/rm/desktopdefault.aspx/tabid-3925/>
- [12] delphigl: <http://wiki.delphigl.com/index.php>
- [13] glProgram: <http://glprogramming.com/red/chapter01.html>
- [14] khronos: <http://www.khronos.org/opengles/sdk/docs/man/>
- [15] metaio: <http://www.metaio.de/presse/press-release/2011/>
- [16] microvision: <http://www.microvision.com>