



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Nico Manske

Kamerabasierte Präzisionsnavigation mobiler Systeme im
Indoor-Bereich

Nico Manske

Kamerabasierte Präzisionsnavigation mobiler Systeme im
Indoor-Bereich

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter: Prof. Dr. rer. nat. Stephan Pareigis

Abgegeben am 5. Dezember 2008

Nico Manske

Thema der Masterarbeit

Kamerabasierte Präzisionsnavigation mobiler Systeme im Indoor-Bereich

Stichworte

Pose, Posenbestimmung, Indoor Navigation, Hand-Eye, Kamerakalibrierung, Tool Center Point, Vehicle Center Point, 3D-Kameramodell, digitale Bildverarbeitung, 6 Freiheitsgrade

Kurzzusammenfassung

Die Grundlage der Navigation von mobilen Systemen bildet die Positionsbestimmung. Für die Navigation im Innenraum spielt die Präzision eine sehr große Rolle, weil sie oft auf engem Raum statt findet. Diese Arbeit gibt einen Überblick vorhandener Techniken zur Positionsbestimmung im Indoor-Bereich und bewertet diese anhand von aufgestellten Anforderungen der Navigation in Räumen bis hin zur Größe von Lagern. Es wird eine kamerabasierte Lösung der Posenbestimmung aufbauend auf einem 13 Parameter 3D-Kameramodell entwickelt, welche sich durch eine Genauigkeit im Bereich von Millimetern und eine leichte Kalibrierbarkeit auszeichnet. Des Weiteren ist die Lösung aufgrund der sechs Freiheitsgrade auf jede Art von Plattform portierbar. Zusätzlich wird das Konzept auf einer mobilen Plattform als Beleg der Konzeption realisiert. Abschließend erfolgt eine Auswertung der Ergebnisse bezüglich der Genauigkeit und der Abtastrate gemessen auf der mobilen Plattform.

Title of the paper

Camera-based Indoor Precision Navigation of mobile Systems

Keywords

Pose, Pose Estimation, Indoor Navigation, Hand-Eye, Camera Calibration, Tool Center Point, Vehicle Center Point, 3D Camera Model, digital Image Processing, 6 Degrees of Freedom

Abstract

Pose estimation provides the basis of mobile systems navigation. Especially for indoor navigation precision is highly important due to small distances. This thesis gives an overview of existing positioning techniques in the indoor area and evaluates these with respect to the requirements of navigation in areas ranging from small rooms up to areas of the size of warehouses. A camera-based solution employing a thirteen parameters 3D camera model is being developed, featuring a system which is easy to calibrate while having an accuracy of a few millimeters. The solution presented in this thesis is portable to any kind of platform due to the six degrees of freedom (6DoF) in translation and rotation. As a proof of concept it is being realized on a mobile platform. Finally the results which have been measured on the mobile platform will be evaluated with regard to precision and sampling rate.

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben.

In erster Linie danke ich meiner Frau Wiebke, die mir während des gesamten Studiums besonders in der Phase der Abschlussarbeit verständnisvoll zur Seite stand. Zudem danke ich ihr für das Korrekturlesen der Arbeit und die Tipps zur äußeren Form. An dieser Stelle danke ich ebenso meinem Schwager Marco Dahlem.

Besonderen Dank richte ich an meinen betreuenden Prüfer Prof. Dr.-Ing. Andreas Meisel, der mir dieses interessante Thema zur Verfügung stellte und während unserer Gespräche immer wieder inspirierende Ideen und Vorschläge äußerte.

Des Weiteren danke ich meinen Kommilitonen Dirk Ewerlin und Eike Jenning, die stets für anregende Diskussionen bereitstanden.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Anwendungsszenario der kamerabasierten Indoor-Navigation	2
1.2	Zielsetzung der Masterarbeit	3
1.3	Aufbau der Arbeit	4
2	Grundlagen der kamerabasierten Navigation	5
2.1	Photogrammetrisches 3D-Kameramodell zur Ermittlung der Kamera-Pose	5
2.2	Kalibrierung der internen Parameter mit dem Photomodeler	9
2.3	Kalibrierung der externen Parameter	10
2.4	Linsenverzeichnungskorrektur zur Erhöhung der Genauigkeit	11
2.5	Plattform-Pose zur Navigation der mobilen Plattform	12
3	Übersicht zu Prinzipien der Positionsbestimmung	14
3.1	Prinzipien der Indoor-Positionsbestimmung	14
3.2	Ermittlung der Plattform-Pose aus der Kamera-Pose	20
4	Mathematische und konzeptionelle Realisierung	25
4.1	Bestimmung der Kamera-Pose in Bezug zum Weltkoordinatensystem	26
4.2	Linsenverzeichnungskorrektur zur Erlangung von Sub-Pixelgenauigkeit	33
4.3	Bestimmung der Plattform-Pose	34
4.4	Raummodell zur Identifikation und Zuordnung von Passmarkenkoordinatenpaaren	43
4.5	Vereinfachte Bahnplanung für die Navigation ohne Hindernisse	45
5	Implementation des gesamten Konzeptes	48
5.1	Verwendung der Linsenverzeichnungskorrektur als Modul	51
5.2	Kamerabasierte Posenbestimmung der mobilen Plattform	51
5.3	Planung der Plattformbewegung zur Ansteuerung von Zielen	55
5.4	Verwendung von Impresario zur Verbindung der Module	57
6	Messtechnische Auswertung der realisierten Lösung	59
6.1	Vermessung der Testumgebung mittels Photogrammetrie	61
6.2	Präzision der VCP-Bestimmung	63
6.3	Zeitverhalten unter realen Bedingungen	66
6.4	Berücksichtigung der nicht messbaren Anforderungen	67
7	Ausblick auf Verbesserungen und Erweiterungen	69
7.1	Auswertung von Odometriedaten zur Ergänzung der Pose	69
7.2	Verbesserung der Präzision bis in den Millimeterbereich	70
7.3	Verringerung der Ausführungszeiten zur Erhöhung der Abtastrate	70
7.4	Verbesserung der Anwendbarkeit der Navigation	71
7.5	Portierung der Lösung in den FAUST-Kontext	71
8	Zusammenfassung	73

Abbildungsverzeichnis	74
Tabellenverzeichnis	76
Quellcodeverzeichnis	77
Glossar	78
Abkürzungsverzeichnis	79
Literaturverzeichnis	80
A Kalibrierung der Kamera des mobilen Systems	83
B Berechnungsansätze	85
B.1 Nichtlineare Ausgleichung mit der Newtonschen Nullstellensuche	85
B.2 Gleichungen und Ableitungen für die Berechnung der Pose	86
C Linsenverzeichnungskorrektur eines kompletten Bildes	89
C.1 Anwendung der nichtlinearen Ausgleichung zur Korrektur kompletter Bilder . .	90
C.2 Quellcodeausschnitt der Linsenverzeichnungskorrektur gesamter Bilder	91
D Daten zur Testumgebung	93
D.1 Visualisierung des Mess-Rasters der Testumgebung	93
D.2 Visualisierung der Messpunktverteilung zur Messung der Präzision	95
D.3 Messergebnisse der Präzision und der Ausführungszeiten	96

1 Einleitung

Im Bereich der Technischen Informatik an der Hochschule für Angewandte Wissenschaften (HAW) Hamburg wird im Kontext des FAUST-Projektes Technologie für Fahrerassistenz- und Autonome Systeme entwickelt. Schwerpunkte in diesen Projekten bilden die Sensorik, Telemetrie und digitale Bildverarbeitung, Echtzeitsysteme und Bussysteme, Software- und Hardwarearchitekturen sowie Algorithmen und Steuerung. Die FAUST-Projekte dienen der Durchführung von Entwicklungs- und Forschungsprojekten im Bereich von verteilten Hard Real-Time Systemen.

Es wurden bereits automatische Brems-, Ausweich- und Einparkassistenten mit Laserscannerbasierten Objekterkennungsmodellen für das Sensor Controlled Vehicle (SCV) entwickelt und auf verteilten PC-104 Rechnerkomponenten implementiert.

Eine weitere notwendige Funktion für ein autonomes Fahrzeug ist ein präzises und zuverlässiges Navigationsmodul für virtuelle Fahrspuren und zum autonomen Abfahren vorgegebener Strecken. Zur Realisierung eines Navigationssystems ist zunächst eine Bestimmung der Pose¹ notwendig, weil Informationen über die Position und Lage des Objektes für Entscheidungen bezüglich der Wegfindung vorausgesetzt werden. Dieses Modul zur Navigation soll zur Verwendung auf nahezu allen mobilen Plattformen geeignet sein. Ein vorstellbares Einsatzszenario eines solchen Moduls wäre die Nutzung auf Gabelstaplern in Lagern. Die Navigation muss hier sehr präzise und zuverlässig arbeiten. Zudem unterliegen vorhandene funkbasierte Lösungen in Lagern oft Störungen durch bestehende Metallkonstruktionen, wie Hochregalen. Neben den bekannten Hindernissen und Raumbeschaffenheiten, können in Lagern die für Kamerasysteme wichtigen Lichtverhältnisse kontrolliert werden. Aufgrund der bekannten Situation kann zusätzlich auf bestimmte Einflüsse der Umgebung reagiert werden.

In dieser Arbeit werden verschiedene Prinzipien der Positionsbestimmung innerhalb von Räumen vorgestellt und deren Eigenschaften anhand aufgestellter Anforderungen für den Einsatz in Lagerräumen verglichen. Insbesondere wird geprüft, ob eine kamerabasierte Variante Nachteile mit sich bringt, denn aus der Vorlesung AW1 (TI) entstand die Idee die digitale Bildverarbeitung als Grundlage für die Umsetzung dieses Moduls zu nutzen. Um eine kamerabasierte Navigation präzise umsetzen zu können, wird das 13 Parameter 3D-Kameramodell aus der Vermessungstechnik für die Entwicklung eines prototypischen Posenbestimmung-Moduls im Rahmen dieser Masterarbeit verwendet. Des Weiteren findet eine Auswertung der gewonnenen Ergebnisse in Form einer Genauigkeits- und Laufzeitmessung statt.

Um die Masterarbeit in einem realisierbaren Umfang zu halten, fand die Entwicklung zunächst auf einer eigenständigen Basis ohne Kooperation zum FAUST-Projekt statt. Die spätere Portierung der zu entwickelnden Lösung ist nicht Bestandteil dieser Arbeit.

¹Pose - die Position und Orientierung eines Objektes in einem vorgegebenen Weltkoordinatensystem

1.1 Anwendungsszenario der kamerabasierten Indoor-Navigation

Das zu navigierende mobile System ist mit einem Notebook als Recheneinheit und einem daran angeschlossenen optischen Sensor ausgestattet. Der optische Sensor zur digitalen Aufnahme der Umgebung kann durch eine handelsübliche Charge Coupled Device (CCD) Kamera bereitgestellt werden. Die Navigation soll - ähnlich wie bei Schiffen - anhand bekannter Landmarken wie Leuchttürme oder Tonnen, mittels geeigneter Passmarken mit festen 3D-Koordinaten als Orientierungspunkte realisiert werden. Zur Verwendung der Passmarken müssen diese zuvor durch digitale Bildverarbeitung im 2D-Abbild der Umgebung gefunden und eindeutig identifiziert werden. Die Posenbestimmung wird unter Verwendung des 13 Parameter 3D-Kameramodells aus [Meisel (1994)] vorgenommen.

Es wird zunächst die Pose der installierten Kamera ermittelt. Mit der ermittelten Pose wird anschließend die Pose der Plattform berechnet, welche als Grundlage für die Navigation der Plattform dient. Durch die ständige Posenbestimmung ist die Pose der Plattform zu jeder Zeit bekannt und bildet die Voraussetzung für die Planung sowie die Verfolgung der Route zum Zielpunkt. Ebenso kann eine aus mehreren Streckenpunkten bestehende Route definiert werden, um virtuelle Spuren abfahren zu können.

Die Abbildung 1.1 zeigt das Szenario für die kamerabasierte Indoor-Navigation in dieser Arbeit als Ausgangspunkt für das entwickelte Konzept. Auf dem mobilen System befindet sich ein Notebook als Recheneinheit. Dieser Rechner dient zur Ausführung der entwickelten und vorhandenen Software-Komponenten für die Posenbestimmung der Kamera und der darauf folgenden Bestimmung der Plattform-Pose. Zudem steuert der Rechner den Plattformantrieb. Des Weiteren ist in der Abbildung eine Kamera erkennbar, die zur Erkennung von Passmarken im Raum dient.

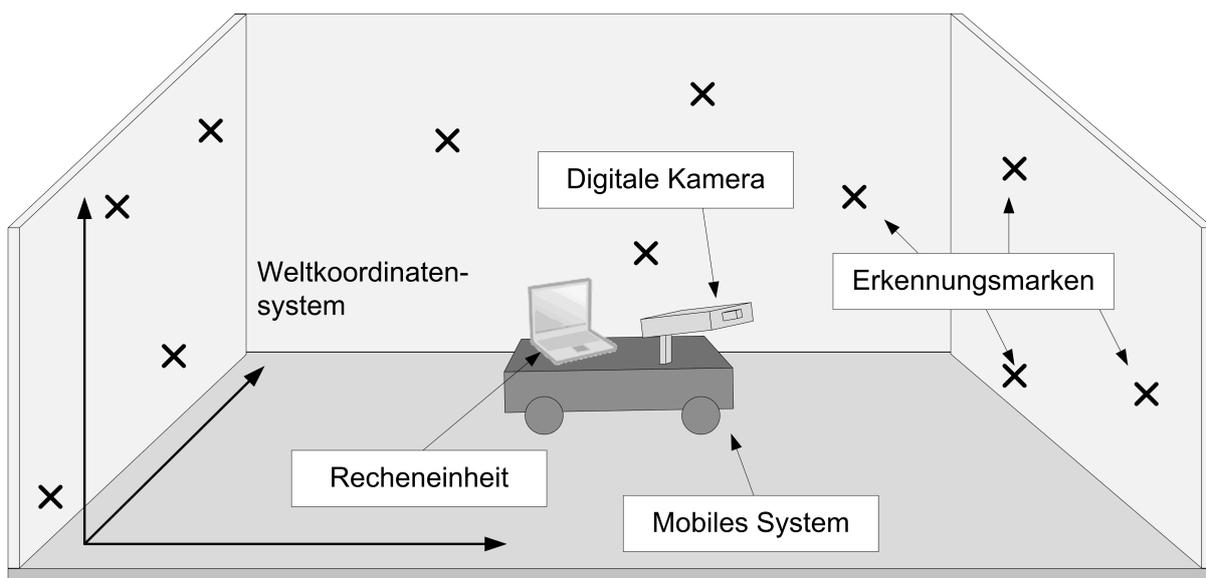


Abb. 1.1: Szenario der Navigation eines mobilen Systems im Indoor-Bereich

1.2 Zielsetzung der Masterarbeit

Das primäre Ziel dieser Masterarbeit ist die Entwicklung eines prototypischen Moduls zur kamerabasierten Navigation eines mobilen Systems. Bei der Konzeption ist auf die Portierbarkeit der Lösung auf jede Art mobiler Systeme zu achten, insbesondere die Bereitstellung von sechs Freiheitsgraden in Bezug auf die Translation und Rotation der Zielplattform.

Die praktische Umsetzung der kamerabasierten Navigation auf einem eigenen mobilen System außerhalb des Fahrerassistenz- und Autonome Systeme (FAUST) Kontext soll den Beleg des Konzeptes liefern, bei dem die Präzision und die Geschwindigkeit vorerst eine untergeordnete Rolle einnehmen. In diesem Zusammenhang wird eine stark vereinfachte Bahnplanung und -verfolgung als erste Anwendung des Navigations-Moduls dienen. Die Planung und Verfolgung der Bahn liegt nicht im Fokus der Arbeit, aus diesem Grund findet zu diesem Thema keine Recherche statt. Statt dessen wird zu Demonstrationszwecken der direkte Weg zum Zielpunkt berechnet. Zudem ist eine Auswertung der gewonnenen Ergebnisse in Bezug auf die Präzision und Ausführungsgeschwindigkeit vorgesehen. Abschließend wird ein Ausblick auf Verbesserungen und Erweiterungen des Konzeptes gegeben.

Folgende Rahmenbedingungen sind bei der Umsetzung der kamerabasierten Navigation zu berücksichtigen, um den Einsatz der Lösung unter realen Bedingungen zu gewährleisten:

- *skalierbar von kleinen Räumen bis auf Lagergröße* - Je nach Szenario soll die Anwendbarkeit der kamerabasierten Navigation in kleinen Räumen oder auch in großen Lagern gegeben sein.
- *genau bis in den Millimeterbereich* - In Lagern müssen Kollisionen mit Gegenständen ausgeschlossen werden, was auf engem Raum eine hohe Präzision erfordert.
- *robust gegen Umgebungseinflüsse* - Einflüsse wie Licht, Akustik, Hitze, Magnetfelder oder Radiowellen sind die häufigsten Störquellen im Bereich der Sensorik. Die Navigation soll durch diese Einflüsse nicht abgelenkt werden.
- *geringes Gewicht und Größe* - Für die Installation auf einem kleinen mobilen System muss die angewandte Technik klein und leicht sein. Auf mobilen Systemen ist aufgrund ihrer Konzeption für bestimmte Zwecke oft nur wenig Platz und die Zuladung ist aus Gründen des Gewichtersparnisses oft auf ein Minimum reduziert.
- *Abstrakte von mindestens 10 Hz* - Mobile Plattformen befinden sich in Bewegung, je nach Fortbewegungsgeschwindigkeit ist daher eine zügige Aktualisierung der Position notwendig. Für die Bewegung innerhalb von Räumen und Lagern wird hier eine Abstrakte von 10 Hz angestrebt.
- *günstig (Hardware-Kosten)* - Autonome Fahrzeuge sind bereits ohne zusätzliche Sensorik kostspielig. Je nach Art der Sensors können weitere Sensorikbausteine ebenfalls hohe Kosten verursachen. Deshalb, und aufgrund der begrenzten zur Verfügung stehenden Mittel, muss das Konzept eine günstige Variante wählen.

- *sparsam (Energie)* - Mobile Plattformen verfügen aufgrund ihrer Mobilität über begrenzte Energieressourcen. Bei der Wahl des Sensors und der weiteren Komponenten ist diesbezüglich auf Sparsamkeit zu achten.
- *flexibel, günstig änderbar* - In Räumen oder in Lagern können Umbaumaßnahmen zur Veränderung der Umgebung und vor allem zu neuen Hindernissen und somit zu neuen Routen führen. Der notwendige Kalibrierungsaufwand nach einem solchen Umbau soll sehr gering gehalten werden.
- *frei beweglich* - Für die freie Bewegung des mobilen Systems müssen sechs Degrees of Freedom, Freiheitsgrade (DOF) für die freie Veränderung der Orientierung und der Position im Raum zur Verfügung stehen. Eine feste Verbindung oder der ständige Kontakt zu äußeren Komponenten darf daher nicht bestehen.

1.3 Aufbau der Arbeit

Im Anschluss an diese Einleitung werden in Kapitel 2 einige Grundlagen erläutert, die zum besseren Verständnis dieser Arbeit notwendig sind. Es handelt sich dabei unter anderem um Grundlagen aus dem Bereich der Photogrammetrie. In diesem Kapitel wird das verwendete Kameramodell, die benötigten sowie die zu ermittelnden Parameter und die eingesetzten Koordinatensysteme erläutert. Ein weiterer Punkt der Grundlagen ist die Einführung in die Thematik zur Bestimmung der Pose des Mobilensystems.

Eine Übersicht der existierenden Prinzipien zur Positionsbestimmung wird in Kapitel 3 vorgestellt. Da es viele verschiedene Techniken zur Lösung dieser Aufgabe gibt, findet bei der Analyse eine Einschränkung der Prinzipien auf die meist verbreiteten Lösungen statt. Zusätzlich erfolgt eine Recherche zu Varianten der Posenbestimmung von Plattformen oder Robotergreifern, aufbauend auf der Kamera-Pose.

Das erarbeitete Konzept sowie die mathematische Realisierung des Konzeptes werden in Kapitel 4 vorgestellt. Des Weiteren erfolgt die Beschreibung der Funktionsweise der Markenerkennung und der direkten Fahrt zu einem definierten Zielpunkt.

In Kapitel 5 wird die Implementation thematisiert. Die Schnittstellen der entstandenen Module werden offen gelegt und beschrieben. Zudem findet eine Beschreibung der Modulverbindung mit Hilfe der grafischen Oberfläche Impresario statt.

Die messtechnische Auswertung der erzielten Ergebnisse erfolgt in Kapitel 6. Es wird die Messung der Präzision sowie die Gegenüberstellung zu den Soll-Werten dargestellt. Die Messung der Ausführungszeit und die resultierende durchschnittliche Abtastrate wird ebenfalls präsentiert.

In Kapitel 7 erfolgt ein Ausblick auf das Potential der erarbeiteten kamerabasierten Navigation aus dieser Arbeit. Es werden Vorschläge zu Verbesserungen der Präzision und der Ausführungszeit unterbreitet. Die notwendigen Schritte für die Portierung in den FAUST Kontext am Beispiel des SCV werden ebenfalls genannt.

Abschließend werden die Ergebnisse dieser Arbeit in Kapitel 8 zusammenfassend dargestellt.

2 Grundlagen der kamerabasierten Navigation

In diesem Kapitel werden die für das Verständnis notwendigen Grundlagen der in dieser Arbeit enthaltenen Lösungen vorgestellt. Dies umfasst auch die Einführung der mathematischen und physikalischen Modelle für die realisierten Lösung. Dazu gehört die

- Beschreibung des verwendeten Kameramodells
- Einführung in die Kalibrierung der internen und externen Kameraparameter
- Linsenverzeichnungskorrektur zur Erhöhung der Genauigkeit, sowie
- Bestimmung der Plattform-Pose unter Verwendung der Kamera-Pose.

Die für die der Posenbestimmung benötigte nichtlineare Ausgleichung wird im Anhang B.1 erläutert.

2.1 Photogrammetrisches 3D-Kameramodell zur Ermittlung der Kamera-Pose

In diesem Abschnitt wird das zur Posenbestimmung verwendete photogrammetrische 3D-Kameramodell beschrieben. Diesem Modell liegt die bildbasierte Bestimmung von Punktkoordinaten zugrunde, mit dessen Hilfe die Vermessung der Lage, der Größe und der Form von Objekten realisiert wird. Durch dieses Modell wird die mathematische Beschreibung des Abbildungsvorgangs eines Raumpunktes auf die Bildebene ausgedrückt. Das verwendete Kameramodell hat sich in der Vermessungstechnik und in weiteren Arbeiten auf diesem Gebiet (vgl. [Föhr (1990)]) bewährt. Es zeichnet sich durch eine Genauigkeit im Millimeterbereich und eine leichte Kalibrierbarkeit aus (vgl. [Meisel (1994)]). Die 13 charakteristischen Parameter werden in den folgenden Abschnitten erläutert. Als Grundlage nutzt dieses Modell die im geodätischen Bereich übliche Messkammer, ähnlich wie das Lochkameramodell (vgl. [Hartley und Zisserman (2003)] S.153-156 und [Jähne (2005)] S.203), welches eine ideale Abbildung der Zentralprojektion liefert. Da reale, nichtideale optische Systeme aber Verzeichnungsfehler aufweisen, werden diese Fehler im 13 Parameter 3D-Kameramodell berücksichtigt. Folglich ist eine Linsenverzeichnungskorrektur der genutzten Bildpunkte zur Erhöhung der Genauigkeit durchführbar.

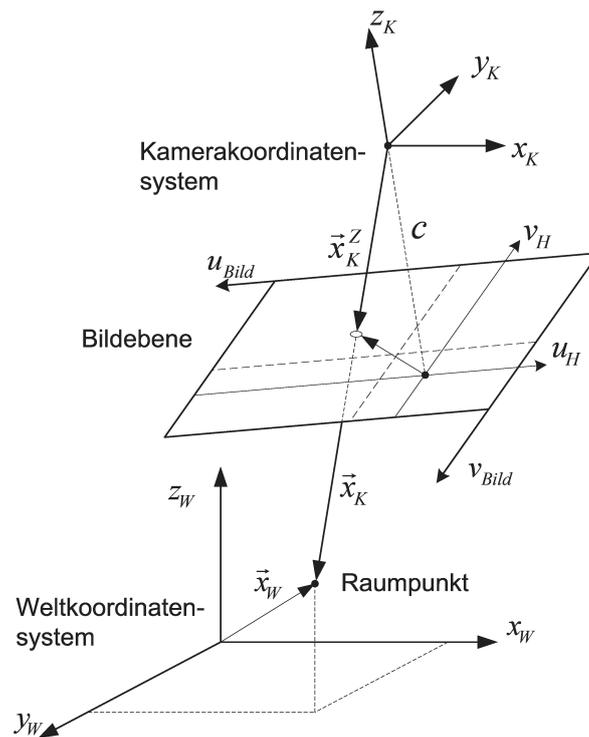


Abb. 2.1: Koordinatensysteme des 3D-Kameramodells

Zum Verständnis der nächsten Abschnitte werden die folgenden vier Koordinatensysteme vorausgesetzt. In Abbildung 2.1 sind die vier Koordinatensysteme und deren Beziehung zueinander abgebildet.

Weltkoordinatensystem $\vec{x}_W = (x_W, y_W, z_W)^T$

Das Weltkoordinatensystem bezeichnet das reale Koordinatensystem (KS), dessen Ursprung und Ausrichtung festgelegt werden. Von diesem System ausgehend erfolgt die Kalibrierung des Gesamtsystems. Die Angabe der Koordinaten wird in 3D-Vektoren vorgenommen. In Abbildung 2.2 erfolgt eine Veranschaulichung des Koordinatensystems.

Kamerakoordinatensystem $\vec{x}_K = (x_K, y_K, z_K)^T$

Dieses Koordinatensystem wird auch als Projektionszentrum-KS bezeichnet, in dem das Projektionszentrum der Kamera den Koordinatenursprung darstellt (Abbildung 2.2). Die optische Achse der Kamera ist die z-Achse dieses Koordinatensystems. Die Angabe der Koordinaten erfolgt ebenfalls durch einen 3D-Vektor.

Bildkoordinatensystem $\vec{u}_{Bild} = (u_{Bild}, v_{Bild})^T$

Dieses zweidimensionale KS liegt im 2D-Abbild der Umgebung, in dem die Koordinaten in 2D-Koordinaten vorliegen. Die Angabe der Koordinaten erfolgt in Pixel, wobei der maximale Pixelwert in x-Richtung der Bildbreite und in y-Richtung der Bildhöhe entspricht. Der Koordinatenursprung liegt in der oberen linken Ecke (siehe Abbildung 2.3).

Bildhauptpunktkoordinatensystem $\vec{u}_H = (u_H, v_H)^T$

In diesem KS liegen die Koordinaten ebenfalls in 2D-Koordinaten vor, jedoch erfolgt die Angabe in einer metrischen Einheit (mm). Der Koordinatenursprung ist in diesem Fall der Bildhauptpunkt, welcher den Durchstoßpunkt der optischen Achse durch die Bildebene repräsentiert. Die Abbildung 2.3 zeigt den Zusammenhang zwischen dem Bild- und Bildhauptpunkt-KS.

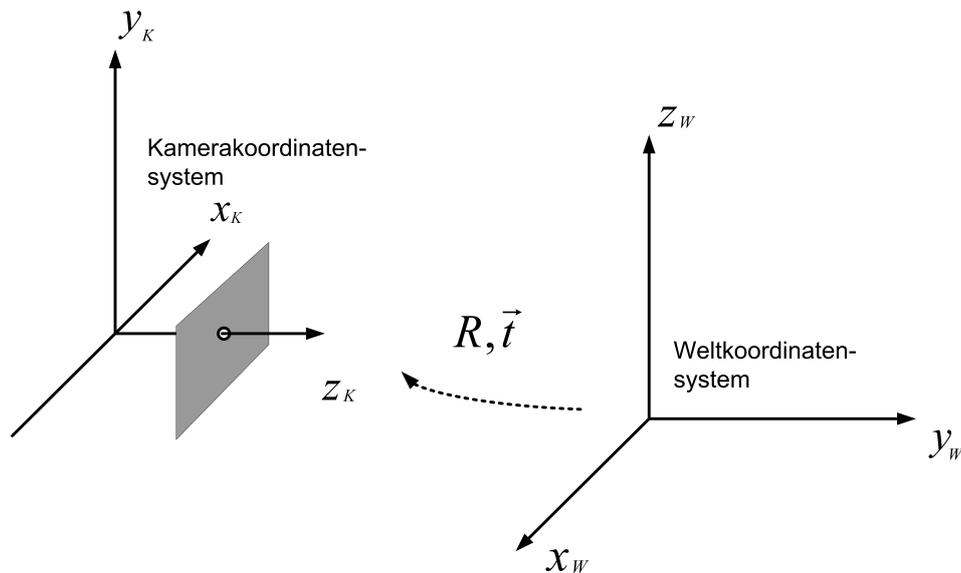


Abb. 2.2: Transformation zwischen Welt- und Kamerakoordinatensystem

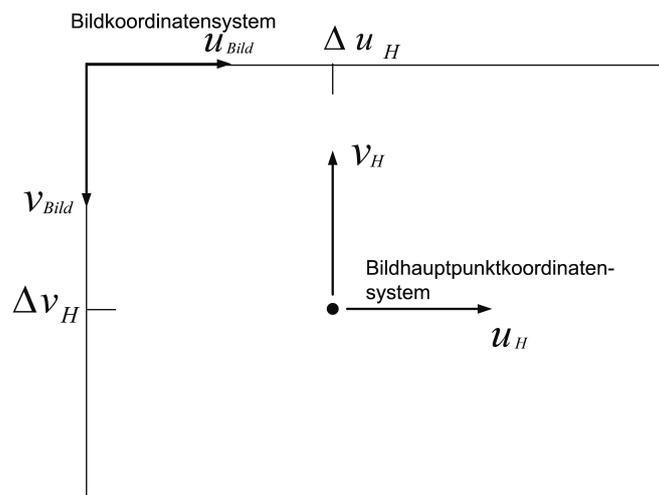


Abb. 2.3: Zusammenhang zwischen Bild- und Bildhauptpunktkoordinatensystem

2.1.1 Kameraparameter des Modells

Die 13 Parameter des Modells setzen sich aus sechs externen und sieben internen Parametern zusammen. Sowohl die externen als auch die internen Parameter lassen sich nicht direkt messen, sie müssen berechnet werden.

Externe Parameter

Die sechs externen Kameraparameter geben die Position und die Orientierung (Pose) der Kamera in Bezug zum Welt-KS an. Zur Angabe der Position werden die 3D-Koordinaten t_x, t_y, t_z verwendet. Sie werden auch durch den folgenden Translationsvektor ausgedrückt.

$$\vec{t} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

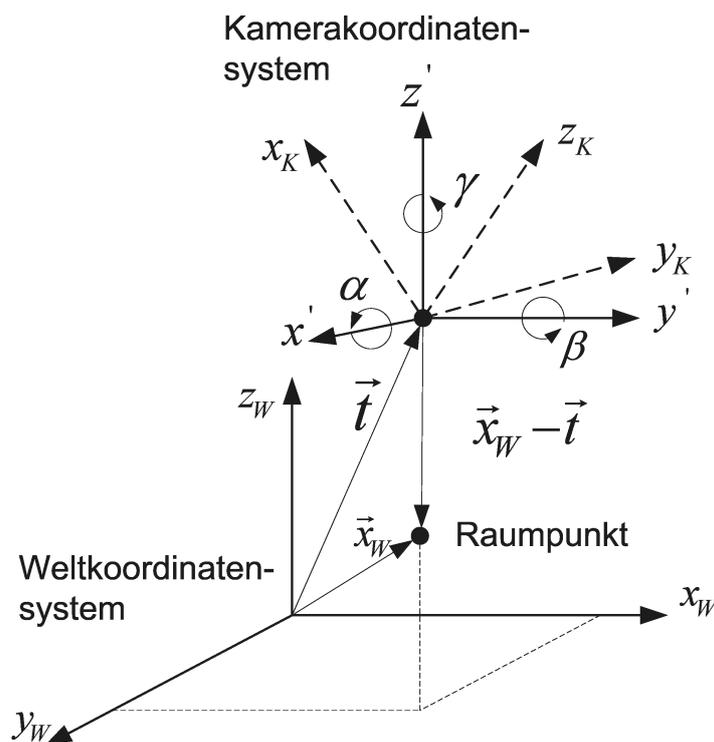


Abb. 2.4: Kameraparameter des Modells im Weltkoordinatensystem

Zur Angabe der Orientierung der Kamera werden die Winkel α , β , γ genutzt. In Abbildung 2.4 werden die externen Parameter und deren Lage im Welt-KS veranschaulicht. Die Winkelbezeichnungen und deren Bedeutung werden nachfolgend aufgelistet.

- α Winkel um die x-Achse (Nickwinkel)
- β Winkel um die y-Achse (Rollwinkel)
- γ Winkel um die z-Achse (Gierwinkel)

Jeder der Parameter entspricht einem Freiheitsgrad, weil die Parameter nicht von einander abhängig sind. Jedoch darf keiner der Freiheitsgrade fehlen, da die Kamera dann nicht jede Pose einnehmen könnte. Grundsätzlich ist es jedoch gestattet, das System um Freiheitsgrade zu minimieren, mit der Auswirkung, dass der Parameter des entfernten Freiheitsgrades nicht mehr variabel ist. Allgemein gilt, das System ist umso flexibler, je mehr Freiheitsgrade existieren. Hierbei ist zu beachten, dass sich die mathematische Komplexität mit jedem weiteren Freiheitsgrad erhöht (Anzahl der Unbekannten).

Interne Parameter

Die folgenden internen Kameraparameter werden einmalig bestimmt und als konstant vorausgesetzt.

- Bildhauptpunktverschiebung, angegeben durch eine 2D-Koordinate ($\Delta u_H \Delta v_H$). Durch die Bildhauptpunktverschiebung wird der Durchstoßpunkt der optischen Achse durch die Bildebene beschrieben.
- Kammerkonstante c , welche die Bildweite darstellt
- Linsenverzeichnungparameter, angegeben durch K_1, K_2 (radial) und P_1, P_2 (tangential)

Für die Bestimmung der internen Kameraparameter muss kein Aufwand betrieben werden, sie können durch die initiale Kalibrierung der verwendeten Kamera bestimmt werden. Die photogrammetrische Vermessungssoftware *Photodeler* stellt Funktionen zur initialen Kalibrierung mit Hilfe einer ausgedruckten Kalibrieranordnung bereit (vgl. [Eos Systems (2008)]). Anschließend können die internen Parameter in der Software ausgelesen werden. Eine Beschreibung der initialen Kalibrierung zur Bestimmung der internen Parameter findet im folgenden Abschnitt statt.

2.2 Kalibrierung der internen Parameter mit dem Photodeler

Die zuvor beschriebenen internen Kameraparameter werden für das weitere Vorgehen benötigt. Sie werden sowohl bei der Linsenverzeichnungskorrektur (vgl. Abschnitt 2.4) als auch bei der Posenbestimmung (vgl. Abschnitt 4.1) genutzt. Somit werden die internen Parameter für das Projekt grundlegend vorausgesetzt. Diese Parameter können durch die Kalibrierung mit der photogrammetrischen Vermessungssoftware *Photodeler* bestimmt werden. Die initiale Kalibrierung muss einmal für die jeweils verwendete Kamera durchgeführt werden. Das Verändern der Fokus- oder Zoomeinstellungen der Kamera nach der initialen Kalibrierung führt jedoch zu anderen internen Parameter. Wenn verschiedene Fokus- oder Zoomeinstellungen genutzt werden, müssen für die jeweiligen Einstellungen separat die internen Parameter bestimmt werden. In dieser Arbeit wird davon ausgegangen, dass die Fokus- und Zoomeinstellungen während des gesamten Posenbestimmungs-Ablaufes beibehalten werden. Es ist somit nur eine einmalige Bestimmung der internen Parameter notwendig.

Die Kalibrierung unter Verwendung von *Photodeler* erfordert mehrere Schritte, welche in dem Bericht zum vorangegangenen Projekt erläutert werden (vgl. [Manske und Jost (2008)] S. 6-7).

Grundlagen der Kalibrierung der internen Parameter

In dieser Arbeit wurde eine handelsübliche Digitalkamera verwendet, um die Abbilder der Umgebung aufzunehmen. Für die Kalibrierung der internen Parameter ist eine Kalibrieranordnung zur Orientierung notwendig. Die Kalibrieranordnung muss über eindeutige Punkte für die Referenzierung verfügen. Auf diese Weise wird sichergestellt, dass die Anordnung der verschiedenen Punkte aus jeder Perspektive eindeutig ist. Die in Abbildung 2.5 dargestellte Kalibrieranordnung wird in dieser Arbeit zur Kalibrierung der internen Parameter genutzt, sie wird zusammen mit dem *Photodeler* bereitgestellt und ist für die Kalibrierung von Kameras empfohlen.

Mindestens sechs Bilder der Anordnung werden für die Kalibrierung benötigt, es ist jedoch zu beachten, dass das Ergebnis genauer wird umso mehr Bilder zur Verfügung stehen. Wenn diese Voraussetzungen erfüllt sind, kann ein Projekt zur Kalibrierung einer Kamera erstellt werden.

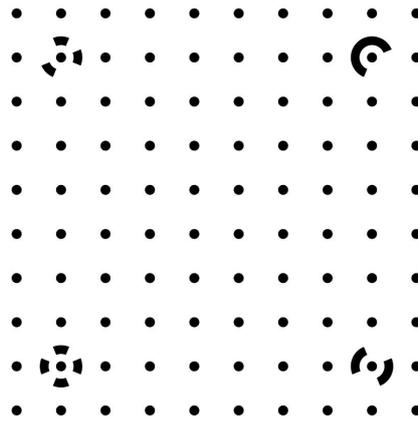
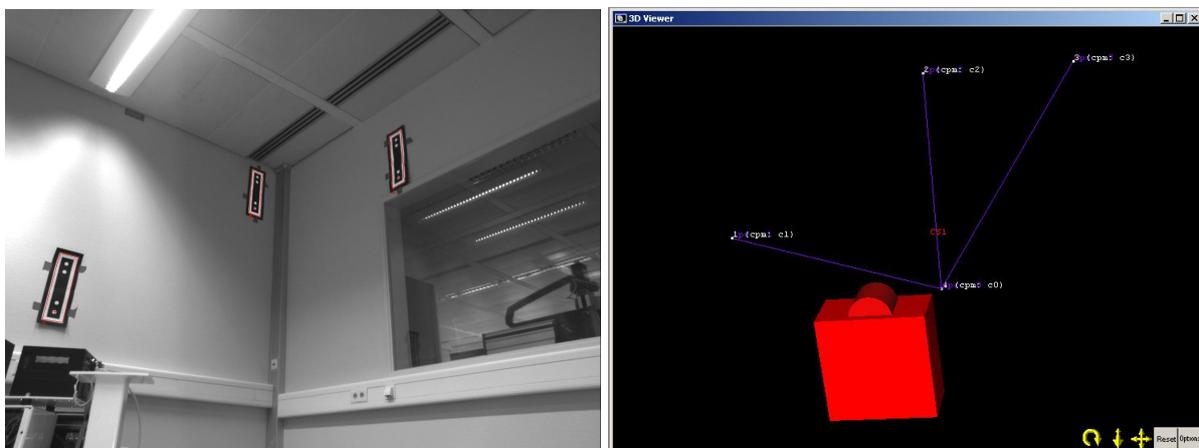


Abb. 2.5: Kalibrieranordnung für die Kalibrierung einer Kamera (vgl. [Eos Systems (2008)])

2.3 Kalibrierung der externen Parameter

Die Bestimmung der externen Kameraparameter ist gleichbedeutend mit der Bestimmung der Kamera-Pose im Bezug zum Welt-KS, was wiederum der Navigation entspricht. Für die Navigation muss die Posenbestimmung ständig durchgeführt werden, weil sich mobile Plattformen in Bewegung befinden.

Die rechnerische Bestimmung der externen Parameter ist eine Variante der Posenbestimmung. Für diesen Zweck werden die sieben internen Parameter vorausgesetzt, um diese im 13 Parameter Kameramodell einsetzen zu können. Bei der vorgestellten Posenbestimmung werden jedoch zur mathematischen Berechnung der externen Parameter Startwerte benötigt. Die Startwerte können geschätzt oder mit dem *Photomodeler* mit der Durchführung der inversen Kamera bestimmt werden. Der *Photomodeler* kann für jedes aufgenommene Bild die externen Parameter (äußere Orientierung) bestimmen. In den Abbildungen 2.6 werden die Aufnahmen der Passmarken aus Kamerasicht (Abbildung 2.6a) und aus Sicht des *Photomodeler* (Abbildung 2.6b) dargestellt, während Abbildung 2.7 die resultierenden externen Parameter im *Photomodeler* zeigt.



(a) Bildaufnahme einer bestimmten Pose

(b) 3D-Viewer zur Ermittlung der externen Parameter

Abb. 2.6: Bildaufnahme aus Kamera- und Photomodeler-Ansicht

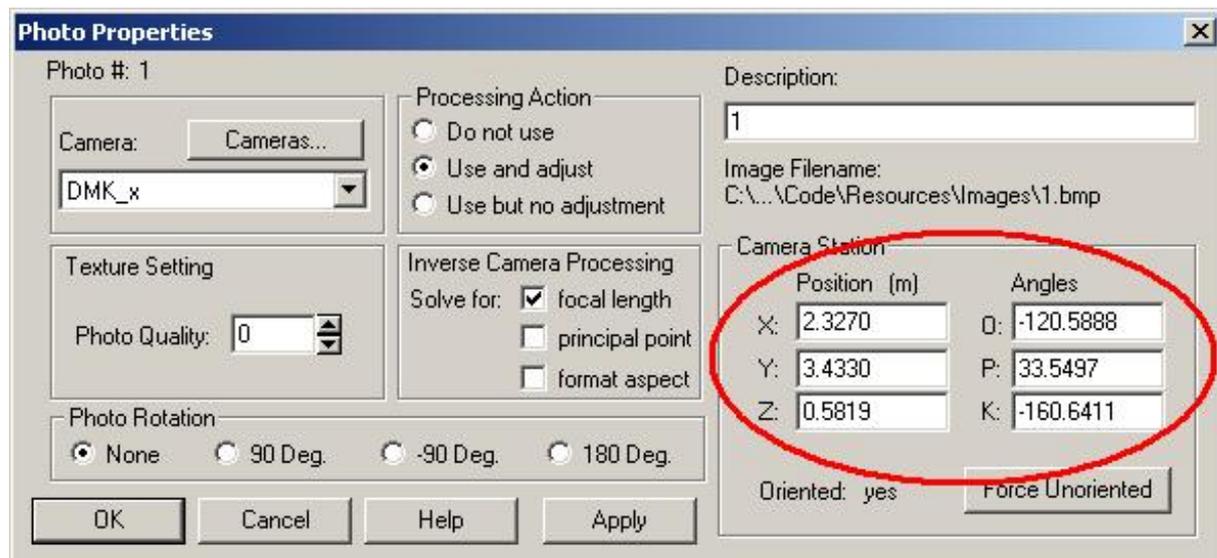


Abb. 2.7: Darstellung der externen Parameter einer Kamera-Pose im Photomodeler

2.4 Linsenverzeichnungskorrektur zur Erhöhung der Genauigkeit

Handelsübliche Objektive weisen meist bedingt durch die Krümmung der Linse eine radialsymmetrische und tangentielle Linsenverzeichnung auf. Die Auswirkung der Verzeichnung ist umso stärker, je weitwinkliger das Objektiv ist und nimmt vom Bildhauptpunkt ausgehend nach außen zu. Als Folge der radialen Verzeichnung treten tonnen- oder kissenförmige Verzeichnungen der Bilder auf. Die tangentielle Verzeichnung dagegen führt zu einer Verdrehung der Bildpunkte um den Verzeichnungsmittelpunkt. Eine Korrektur der Verzeichnung ist notwendig, um die Genauigkeit zu erhöhen, da in dieser Arbeit eine Genauigkeit im Sub-Pixelbereich angestrebt wird. Das Modul zur Linsenverzeichnungskorrektur wurde im vorangegangenen Projekt prototypisch implementiert und im Rahmen dieser Arbeit als Bibliothek zur einfachen Verwendung bereitgestellt. Im Projekt wurde zunächst ein komplettes Bild korrigiert. Dieser Vorgang ist sehr zeitaufwendig, wobei die Zeit von der Größe (Pixelanzahl) des zu korrigierenden Bildes beeinflusst wird. Bei der Posenbestimmung genügt es jedoch die Bildpunkte der angegebenen Bezugspunkte zu korrigieren, weil nur diese von Belang sind. Dieser Fakt beschleunigt die Korrektur massiv, da lediglich die Bildpunkte der angegebenen Passmarken korrigiert werden, während zuvor jeder Bildpunkt des Bildes korrigiert wurde.

Die Korrektur erfolgt mit Hilfe der bei der Kalibrierung der internen Parameter der Kamera (vgl. Abschnitt 2.2) bestimmten Abbildungsfehler. Eine detaillierte Erläuterung der Linsenverzeichnungskorrektur wird in Abschnitt 4.2 vorgenommen. In der nachfolgenden Abbildung 2.8 werden die Auswirkungen der radialen Linsenverzeichnung verdeutlicht.

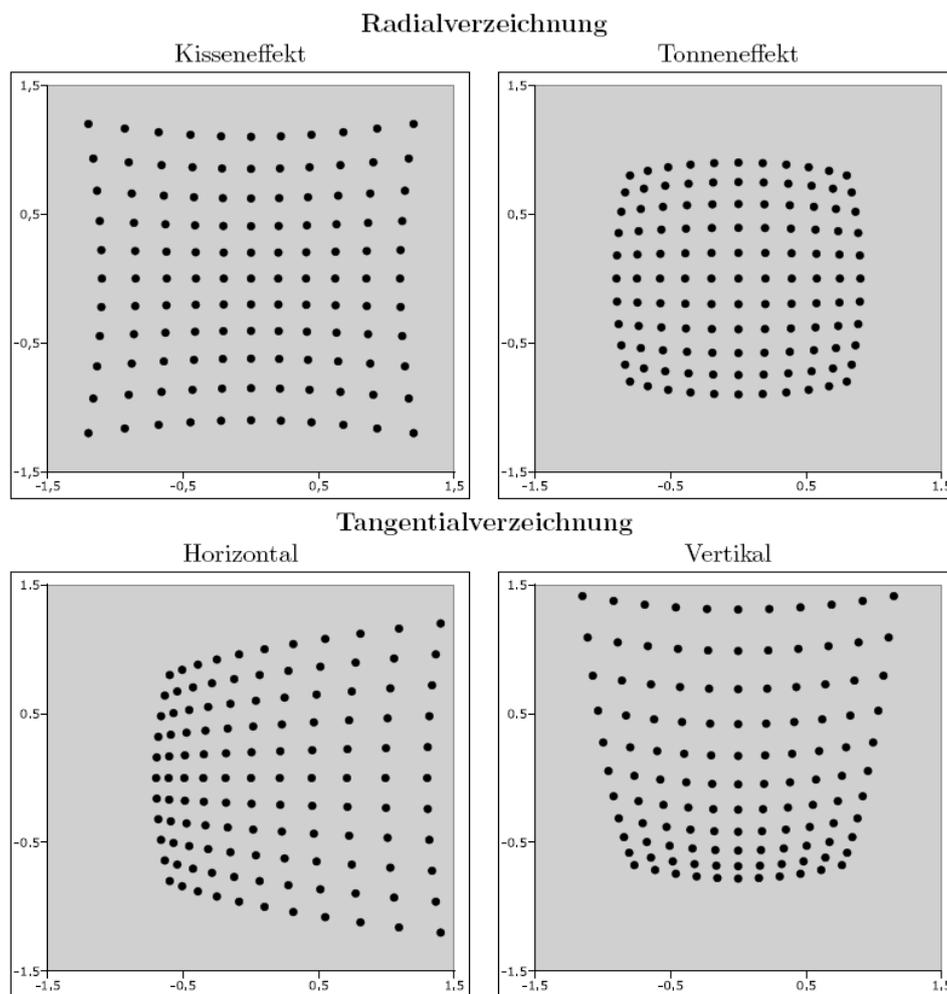


Abb. 2.8: Radiale und tangentielle Auswirkungen der Linsenverzeichnung (vgl. [Eschenburg (2006)])

2.5 Plattform-Pose zur Navigation der mobilen Plattform

Bei der in dieser Arbeit entwickelten Navigation dient eine Kamera als optischer Sensor für die Ermittlung der Pose im Indoor-Bereich. Dabei wird zunächst lediglich die Pose der auf der Plattform installierten Kamera ermittelt. Diese kann in jeder beliebigen Orientierung, je nach Anwendungsfall, ausgerichtet werden. Für die Navigation der Plattform muss bekannt sein, in welcher Weise die Kamera auf der Plattform installiert ist, weil zur Steuerung der Plattform dessen Pose bekannt sein muss, nicht die der Kamera. Für die Ermittlung der Plattform-Pose aus der Kamera-Pose gibt es verschiedene Ansätze. Ein weit verbreiteter Ansatz ist die Bestimmung der relativen Lage der Kamera zum Plattformkoordinatensystem. Dieses Verfahren wird auch als Hand-Eye-Calibration (HEC) bezeichnet und bietet wiederum verschiedene Lösungsansätze zur Bestimmung der relativen Lage der Kamera. In Abschnitt 3.2 werden einige der bekannten HEC-Verfahren erläutert und auf Anwendbarkeit in dieser Arbeit geprüft.

In dem hier vorgestellten Szenario wird die Kamera fest auf der Plattform installiert und nach der initialen Posenbestimmung bis zum Ende der Fahrt nicht mehr verändert. Diese Bedingung erleichtert die Lösung dahingehend, dass die Verschiebung zwischen dem Kamera- und dem Plattformursprung muss nur einmal vor Beginn der Fahrt ermittelt werden muss. Der resultie-

rende Translationsvektor ($\Delta\vec{t}$) und die Rotationsmatrix ($\Delta\mathbf{R}$) werden bei jeder Posenbestimmung wiederverwendet, um die Plattform-Pose zu ermitteln.

In Abbildung 2.9 werden die Posen des Plattformkoordinaten- und des Kamerakoordinatensystemursprungs im Bezug zum Welt-KS dargestellt, um das beschriebene Problem zu verdeutlichen. Es wird sichtbar, dass die Kamera durchaus eine komplett abweichende Position gegenüber der mobilen Plattform aufweisen kann. Die Navigation, beispielsweise in Lagerräumen erfordert jedoch eine sehr hohe Präzision, um Kollisionen zu verhindern und Zielpunkte genau anfahren zu können. Noch deutlicher wird die Notwendigkeit der Bestimmung der Plattform-Pose bei der Orientierung, denn wie in Abbildung 2.9 zu sehen ist, wird diese meist stark von der Orientierung der Kamera abweichen. Dies hat zur Folge, dass eine Bahnplanung ohne die vorherige Bestimmung der Plattform-Pose nicht zuverlässig realisierbar ist.

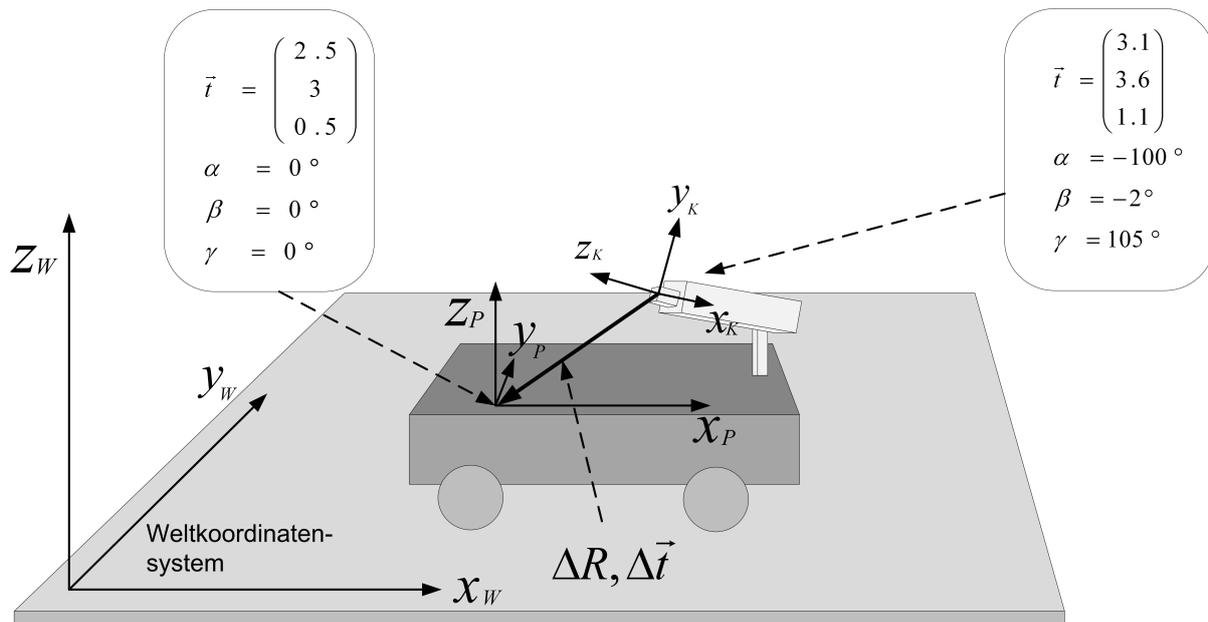


Abb. 2.9: Zusammenhang der Plattform- und Kamera-Pose

3 Übersicht zu Prinzipien der Positionsbestimmung

Die kamerabasierte Navigation ist nicht die einzige Variante zur Posenbestimmung. Hier wird eine Untersuchung und anschließende Bewertung vorhandener Techniken zur Positionbestimmung vorgenommen. Dabei erfolgt die Bewertung an den im Szenario in Abschnitt 1.1 aufgestellten Anforderungen für die Realisierung der kamerabasierten Präzisionsnavigation im Indoor-Bereich. Des Weiteren werden Prinzipien zur Bestimmung der Pose der mobilen Plattform aus der Kamera-Pose untersucht, da für die Navigation des mobilen Systems die Kenntnis über die Pose der genutzten Kamera alleine nicht ausreichend.

3.1 Prinzipien der Indoor-Positionsbestimmung

In diesem Abschnitt werden existierende Prinzipien vorgestellt und ihre Vor- und Nachteile genannt (vgl. [Welch und Foxlin (2002)]). Es werden keine kompletten Lösungen vorgestellt, sondern nur die Grundlagen und Funktionsweisen der verschiedenen Sensoren beschrieben. Diese Recherche soll einen Überblick der vorhandenen Techniken geben und zur Bewertung der vorgeschlagenen Technik für diese Arbeit beitragen. Am Ende dieses Abschnittes werden die einzelnen Prinzipien und deren Eignung für dieses Szenario abschließend übersichtlich in der Tabelle 3.1 dargestellt und an den aufgestellten Anforderungen gemessen. Ein Teil dieser Recherche wurde bereits in der vorangegangenen Seminararbeit vorgenommen, und in dieser Arbeit erweitert (vgl.[Manske (2008)]).

3.1.1 Trägheitssensoren am Objekt

Inertiale Navigationssysteme werden häufig in Schiffen, U-Booten und Flugzeugen eingesetzt. Schon seit den 1950er Jahren werden diese Systeme verwendet, aber der Einzug in die computergestützte Navigation erfolgte erst nach den noch folgenden Prinzipien. In Abbildung 3.1 ist ein beispielhafter Aufbau eines solchen Systems dargestellt.

Ein Inertiales Navigationssystem (INS), in diesem Fall mit Trägheitssensoren ausgestattet, ist in der Lage die eigene Position und Geschwindigkeit ohne Bezug zur Umgebung zu bestimmen. Bei diesem Prinzip werden ständig die Beschleunigungen der, in den drei orthogonalen Achsen installierten, Beschleunigungssensoren (Accelerometers) gemessen. Die einfache zeitliche Integration der Beschleunigungen $a_{x,y,z}(t)$ liefert die Geschwindigkeit $v_{x,y,z}(t)$ und deren Integration ergibt die zurückgelegten Strecken $x(t), y(t)$ und $z(t)$. Um die Genauigkeit zu erhöhen, muss

jedoch der Gravitationseffekt rechnerisch von den Werten der vertikalen Beschleunigungssensoren abgezogen werden (vgl. [King (2005)], [Welch und Foxlin (2002)], [Barbour und Elwell (2005)]).

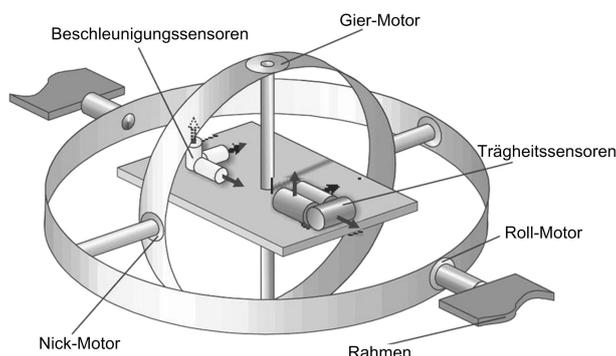


Abb. 3.1: Kardanisch aufgehängte Trägheits- Bewegungseinheit (vgl. [King (2005)])

Kreisel sind im Idealfall absolut Raum stabil und somit sehr gut für die Bestimmung der Orientierung geeignet. Die Kreiselplattform liefert Orientierungsdaten durch die Integration der Kreiselwerte und registriert somit Lageveränderungen des Objektes. Das in Abbildung 3.2 dargestellte Blockschaltbild visualisiert die Abfolge der Positionsbestimmung. In Verbindung der Orientierungsdaten des Kreisels und der Daten der Beschleunigungssensoren wird schließlich die Lage und die Position des Objektes bestimmt. Bei dieser Variante handelt es sich um eine relative Bestimmung der Pose, denn die Bestimmung der Pose kann nur relativ zum Ausgangspunkt erfolgen. Der Ausgangspunkt entspricht dabei der Startgeschwindigkeit in den drei Richtungen. Die Koordinaten im Welt-KS müssen durch Umrechnung der ermittelten Daten in Verbindung mit Startwerten berechnet werden. In der Praxis werden diese Systeme oft in Verbindung mit anderen Navigationssystemen wie Global Positioning System (GPS) eingesetzt, was zu einer Genauigkeit im Meter-Bereich unabhängig von der vergangenen Zeit führt. Gyroskope existieren aber auch schon als Mikro-Elektromechanisches System (MEMS). Diese Systeme zeichnen sich durch die sehr geringe Größe sowie die Leistungsfähigkeit aus und sind dabei kostengünstig. In Bezug auf die in Abschnitt 1.1 aufgeführten Rahmenbedingungen entstehen folgende Vor- und Nachteile.

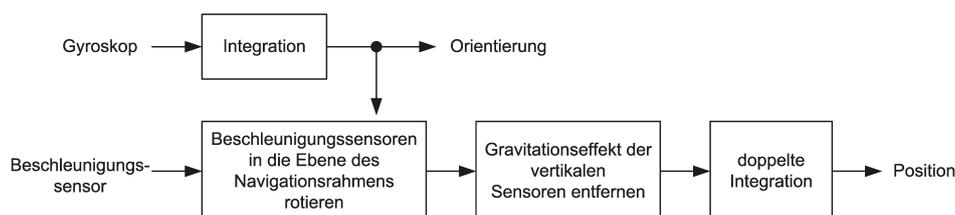


Abb. 3.2: Blockschaltbild der Posenbestimmung mit Trägheitssensoren (vgl. [Welch und Foxlin (2002)])

Vorteile

- Für die Inbetriebnahme ist kein Einrichtaufwand erforderlich.
- Bei dieser Technik liegt ein abgeschlossenes und vollständig eigenständiges System vor.
- Das System ist gegen die in den Anforderungen in Kapitel 1.1 genannten äußeren Einflüsse unempfindlich, eine Ausnahme bilden mechanische Erschütterungen.

- Es liegen sehr geringe Verzögerungszeiten vor (nur wenige ms).
- Das Verfahren kann mit einer sehr hohen Abtastrate eingesetzt werden.
- Gyroskope in Form von MEMS sind sehr klein.
- Bestimmung der Position und der Orientierung des Objektes.

Nachteile

- Fehler in der Messung (oder bei Kalibrierung) werden durch die zeitliche Integration mit zunehmender Messzeit ständig größer.
- Die Messung der Position erfolgt *relativ* zum Startpunkt.

3.1.2 Senden und Empfangen von Schallwellen

Schallwellen können verwendet werden, um die Position zu bestimmen, wobei die bekanntesten Methoden der Einsatz von Ultraschallwellen in Form von Echolot und Sonar auf U-Booten, Schiffen und in der Medizin sind.

Bei diesem Prinzip wird durch die Messung der Signallaufzeit von Ultraschallimpulsen die Entfernung zu Gegenständen bestimmt. Die Signallaufzeit kann für reflektierte Signale oder von anderen Sendern gesendete Signale ermittelt werden. Im Verbund mit mehreren Ultraschallsendern und des mobilen Systems als Empfänger kann die Position des Empfängers trianguliert werden, wenn die Positionen der Sender bekannt sind. Zusätzlich ist bei dieser Variante aber ein weiterer Mechanismus notwendig, der signalisiert wann der Impuls gesendet wird. Ein ähnliches Verfahren wird von dem an der HAW entwickelten Indoor Distance Measurement and Positioning System (IMAPS) genutzt, es kombiniert Funk- und Ultraschalltechnik (vgl. [Gregor (2006)]). IMAPS wies im letzten bekannten Entwicklungsstand jedoch Probleme mit den Ultraschallsendern in Bezug auf die Genauigkeit auf. Des Weiteren muss hoher Installations- und Verwaltungsaufwand für die Inbetriebnahme des Systems betrieben werden.

Durch den Einsatz von Schallwellen können Multipfad Reflektionen² entstehen, die zu unberechenbaren Abweichungen führen können. In neueren Varianten werden Reflektionen durch das Ausblenden bereits empfangener Signale verhindert, somit werden Reflektionen nicht registriert. Ultraschallwellen liegen im Bereich von 20 kHz - 1 GHz und sind somit vom Menschen nicht wahrnehmbar. Meistens werden Frequenzen im Bereich von 40 - 80 kHz eingesetzt, um Umgebungsschallwellen und Interferenzen auszublenden. Des Weiteren ist durch die hohe Frequenz eine höhere Auflösung realisierbar. Im Folgenden werden die Nachteile dieser Variante genannt, hervorzuhebenden Vorteile sind nicht vorhanden.

Nachteile

- Die Geschwindigkeit von Schallwellen ist stark abhängig von Luftströmen, der Temperatur und der Luftfeuchtigkeit. Die Geschwindigkeit ändert sich um 0.1 % pro °C Tempe-

²Multipfad Reflektionen - Wände und Gegenstände in Räumen sorgen für starke Reflektionen, Multipfad Reflektion sind die Summe von einer oder mehrerer Signalreflektionen

raturberänderung. Daraus ergibt sich ein Fehler von 1 mm pro Meter pro °C Temperaturänderung.

- Die Abtastrate liegt bei 10 Hz. Diese relativ langsame Abtastrate entsteht durch die langsame Ausbreitung der Schallwellen (ca. 3 ms pro Meter, 343 m/s).
- Es wird nur die Position ermittelt, die Ermittlung der Orientierung ist ohne weiteres nicht möglich.
- Für die Anwendung der Triangulation werden mehrere verteilte Sender benötigt.

3.1.3 Radio- und Mikrowellen

Radio- und Mikrowellen sind weit verbreitet in Navigationssystemen, Flughafen-Landehilfen (Landfunkfeuer) sowie in der bekanntesten Anwendung dem Radar. Im Outdoor-Bereich ist diese Technik neben dem GPS die meist verwendete Technik zur Positionsbestimmung. Eine direkte Bestimmung der Position ist auch hier nicht durchführbar. Mit Hilfe der Entfernungsmessung zu bekannten Punkten, wie zum Beispiel Leuchttürmen, ist jedoch die Positionsbestimmung realisierbar. Während diese Technik zur Zeit meist in größeren Systemen eingesetzt wird, wird diese in Zukunft, wenn die Technologie präziser und günstiger ist, auch in kleineren Tracking-Systemen eingesetzt werden können.

Ebenso wie bei Schallwellen wird hier die Signallaufzeit (Echos) zur Feststellung der Position gemessen. Radiowellen sind eine Millionen mal schneller als Schallwellen, dadurch sind die Abstände zwischen den einzelnen Messungen wesentlich geringer als bei der Messung mit Ultraschallwellen (vgl. [Göbel (2001)] und [Welch und Foxlin (2002)]). Die hohe Geschwindigkeit der Radiowellen (Lichtgeschwindigkeit, 299.792.458 m/s) erschwert jedoch die Ermittlung der Position im Millimeter-Bereich, denn für diese Genauigkeit wäre eine Messung der Signale mit einem Abstand von ca. 3 Pikosekunden (333 GHz) nötig. Die Vor- und Nachteile beim Einsatz von Radio- und Mikrowellen werden im Folgenden genannt.

Vorteile

- Radio- und Mikrowellen zeichnen sich durch eine sehr geringe Absorptionen aus.
- Sie werden nicht beeinflusst von Luftströmen, Luftfeuchtigkeit und Lufttemperatur.
- Diese Technik stellt hohe Abtastraten bei den Messungen zur Verfügung.

Nachteile

- Bei Nutzung von sinusförmigen Signalen mit großer Wellenlänge entsteht die Gefahr von bereits erwähnten Multipath Reflektionen, im Indoor-Bereich sind viele Reflektionen nicht verhinderbar.
- Eine Genauigkeit im Millimeter-Bereich erfordert zu geringe Abstände zwischen den Messungen (3 ps).

- Bei dieser Technik kann die Orientierung eines Objektes nicht bestimmt werden.

3.1.4 Kamerabasierte Posenbestimmung

Der Einsatz von optischen Sensoren in Form von CCD Kameras bietet verschiedene Varianten zur Ermittlung der Pose. Die in dieser Arbeit entwickelte Lösung nutzt das 13 Parameter 3D Kameramodell aus Abschnitt 2.1. Das aus der Vermessungstechnik bekannte Modell wird in der Regel genutzt, um aus der Kenntnis der Kamera-Pose die absolute Position von Weltpunkten zu berechnen. Zur Ermittlung der Kamera-Pose wird hier jedoch das inverse 13 Parameter Modell angewendet. Erläuterungen zu den Details und der Funktionsweise dieser Lösung erfolgen in Abschnitt 4.1.

Vorteile

- Es ist nur eine einmalige Kalibrierung des Systems notwendig.
- Sehr hohe Genauigkeit der Technik bis in den Millimeterbereich.
- Das Prinzip ist gegen die in den Anforderungen in Kapitel 1.1 genannten äußeren Einflüsse unempfindlich, eine Ausnahme bilden jedoch starke Sonnenlichteinstrahlungen.
- Die Größe von CCD-Kameras ist sehr gering, ebenso wie das Gewicht.
- CCD-Kameras sind günstig und benötigen nicht viel Energie.
- Das System ist bei Veränderung der Raumbeschaffenheit flexibel änder- und anpassbar.
- Bestimmung der Position und der Orientierung des Objektes.

Nachteile

- Einschränkungen bei der Abtastrate durch die notwendige Bildverarbeitung für die Raumpunkterkennung. Es wird eine Abtastrate von 10 Hz angestrebt.

3.1.5 Nicht berücksichtigte Prinzipien

Bei der Analyse der existierenden Technologien zur Positionsbestimmung wurden einige Technologien nicht berücksichtigt, weil sie aufgrund ihrer Eigenschaften oder Funktionsweisen nicht für dieses Szenario genutzt werden können. Dieser Abschnitt gibt einen kurzen Überblick dieser Technologien und enthält die Ausschlusskriterien dieser Varianten.

Mechanische Kopplung - Diese Technik setzt eine physikalische Verbindung zwischen der Umgebung und der mobilen Plattform voraus, womit der Aktionsradius sehr stark eingeschränkt und für Lagerräume ungeeignet ist. Eine komplette Bewegungsfreiheit ist hier nicht gegeben (vgl. [Welch und Foxlin (2002)]).

Magnetsensoren - In Magnetfeldern treten starke Störungen durch ferromagnetische oder leitende Teile auf, besonders in Lagern würden diese Eigenschaften zu großen Problemen führen (vgl. [Welch und Foxlin (2002)]).

Drucksensoren - Drucksensoren im Boden, die Objekte durch ihr Gewicht detektieren können, sind für eine sehr genaue Positionsbestimmung nutzbar. Diese Variante erfordert jedoch einen sehr hohen Installationsaufwand und ist nicht flexibel veränderbar (vgl. [Lemelson (2005)]).

GPS - Die GPS Signale werden durch Wände stark gedämpft, wodurch der Einsatz im Indoor-Bereich unzuverlässig ist. Das Projekt *INDOOR* beschäftigt sich mit dem Ausbau von GPS im Indoor-Bereich, jedoch liegen noch keine veröffentlichten Ergebnisse vor (vgl. [INDOOR (2008)]).

Zusammenfassender Überblick der Kriterienmatrix

Das in Abschnitt 1.1 beschriebene Szenario setzt die Erfüllung einiger Anforderungen in Bezug auf die Posenbestimmung und die damit verbundene Navigation voraus. Dabei handelt es sich überwiegend um Bedingungen, die zum Einsatz in der Praxis erfüllt werden sollten. Die nachfolgend erneut aufgeführten Anforderungen habe ich durch Überlegungen zur Umsetzung des Szenarios zum Beispiel in Lagerräumen entwickelt.

Anforderungen

- skalierbar von kleinen Räumen bis auf Lagergröße
- genau bis in den Millimeterbereich - in Lagern müssen Kollisionen mit Gegenständen ausgeschlossen werden
- robust gegen Umgebungseinflüsse - Einflüsse wie Licht, Akustik, Hitze, Magnetfelder oder Radiowellen
- geringes Gewicht und Größe - zur Installation auf einer kleinen mobilen Plattform
- Abtastrate von mindestens 10 Hz - Plattform befindet sich in Bewegung
- günstig (Hardware-Kosten)
- sparsam (Energie) - mobile Plattformen verfügen über begrenzte Energieressourcen
- flexibel, günstig änderbar - der Aufwand nach Umbaumaßnahmen im Lager ist zu berücksichtigen (Kalibrierungsaufwand)
- frei beweglich - sechs Freiheitsgrade für die freie Veränderung der Orientierung und der Position

Jedes der analysierten Prinzipien hat Vor- und Nachteile. In der Praxis werden Prinzipien oft kombiniert (hybride Lösung), um die Nachteile der einzelnen Lösung auszugleichen (vgl. [Choi u. a. (2006)]). In den meisten Fällen entscheiden die Rahmenbedingungen in der Umgebung und die Anforderungen des Projektes über die zu verwendenden Prinzipien. In nahezu allen Projekten müssen bei der Auswahl der angewandten Technik vorgegebene Bedingungen und

Einschränkungen berücksichtigt werden. Grundsätzlich ist jedoch immer die absolute der relativen Positionsbestimmung vorzuziehen, weil sie von dem Startzustand unabhängig ist.

Um eine Bewertung der vorgestellten Prinzipien in Bezug zur Verwendung in diesem Szenario vornehmen zu können, sind die aufgeführten Anforderungen zu beachten. In der Tabelle 3.1 werden die Bewertungen der einzelnen Technologien zusammenfassend dargestellt. Die Bewertung erfolgte anhand der oben aufgestellten Anforderungen und bezieht sich auf die Eignung für den Einsatz in diesem Szenario.

Tabelle 3.1: Bewertung der Prinzipien gegenüber den Anforderungen

Kriterien	existierende Prinzipien zur Positionsbestimmung					
	mechanisch	Trägheits-sensoren	akustisch	magnetsich	Radio-/Mikrowellen	kamerabasiert
frei beweglich	--	++	++	++	++	++
Skalierbarkeit	--	++	o	o	++	++
Kosten / Hardwareaufwand	--	+	o	+	--	++
Genauigkeit	+	+	+	+	++	++
Orientierung	+	+	-	-	-	+
absolute Messung	+	-	+	+	+	+
Anfälligkeit	++	o	--	--	--	o
Größe/Gewicht der Sensoren	++	++	o	++	-	++
Abtastrate	o	++	-	+	++	-
flexibel änderbar	--	++	o	o	++	+
Energieverbrauch	+	+	+	+	o	+
Kalibrierlaufwand	--	--	-	o	++	+

nach Eignung für Einsatz -- sehr schlecht - schlecht o ausgeglichen + gut ++ sehr gut

Die Zusammenfassung aus der vorangegangenen Seminararbeit zeigt, dass keine der vorgestellten Techniken aufgrund der Eigenschaften priorisiert werden kann (vgl. [Manske (2008)]), wobei die kamerabasierte Posenbestimmung die Anfälligkeit durch Lichteinflüsse durch entsprechende Filter ausgleichen kann. Zudem können Probleme der mit 10 Hz relativ geringen Abtastrate durch die Kombination der Posenbestimmung mit Odometriedaten kompensiert werden. Da die kamerabasierte Technik absolute Messungen vornimmt und auch eine Bestimmung der Orientierung umsetzt, folgt der Schluss diese Variante ohne Bedenken einsetzen zu können.

3.2 Ermittlung der Plattform-Pose aus der Kamera-Pose

Die prototypisch im vorangegangenen Projekt implementierte Posenbestimmung einer Kamera genügt nicht zur Navigation eines mobilen Systems, denn die Kamera kann in beliebiger Weise auf der Plattform installiert werden. Für die Navigation muss jedoch die Lage und Orientierung der mobilen Plattform und nicht der Kamera bekannt sein.

Hier werden bestehende Varianten zur relativen Bestimmung der Plattform-Pose mit Hilfe der Kamera-Pose analysiert und auf die Einsetzbarkeit in dieser Arbeit geprüft. Bei einer Variante kann die Plattform-Pose durch die Kenntnis der relativen Beziehung zwischen der Plattform und der darauf installierten Kamera bestimmt werden. Dieses Verfahren wird als HEC bezeichnet. Die in Abschnitt 3.2.2 folgende Analyse der Techniken zur HEC zeigt, dass die Variante zur Bestimmung der Plattform-Pose sehr komplex und aufwendig ist. Zum einen müssen mindestens zwei Positionen während der initialen Kalibrierung angefahren werden, die sich in Translation und Rotation unterscheiden. Zum anderen wird die exakte Kenntnis der genutzten Posen vorausgesetzt, um eine genaue Kalibrierung durchführen zu können. Aufgrund der Anforderung der *einfachen Kalibrierbarkeit* scheidet die Methode der HEC für die Anwendung in dieser Arbeit aus. Statt dessen wird die im folgenden Abschnitt erläuterte absolute Bestimmung der Plattform-Pose für den Einsatz in dieser Arbeit gewählt. Mit dieser Methode wurde ein Weg gefunden die komplizierte HEC zu vermeiden. Die genutzte Variante hat den Vorteil, dass bereits entwickelte Komponenten genutzt werden und die Lösung auf bestätigten Fakten aufbaut. Die Pose der Plattform kann dabei absolut bestimmt werden. Es handelt sich um die direkte Bestimmung des sogenannten Tool Center Point.

3.2.1 Absolute Tool Center Point Bestimmung

Die komplizierte und aufwendige HEC wird überflüssig wenn der sogenannte Tool Center Point (TCP) absolut bestimmt werden kann. Es wird aber dennoch die Kamera-Pose benötigt, um den TCP zu bestimmen. Die Voraussetzung für die absolute TCP-Bestimmung ist die Kenntnis der Plattform-Pose zum Startzeitpunkt und die starre Installation der Kamera auf der Plattform. Soll die Kamera während der Fahrt unabhängig von der Plattform bewegt werden ist dieses Verfahren nicht zulässig.

Der Kerngedanke dieser Variante besteht in der Bedingung, dass der Koordinatenursprung eines starr mit der Plattform verbundenen KS zum Startzeitpunkt deckungsgleich mit dem Weltkoordinatensystemursprung ist und während der Fahrt mitbewegt wird. So wird eine Modellierung vorgenommen, die die Bewegung des Welt-KS in dem mitbewegten KS wiedergibt. Aufbauend auf dieser Idee kann durch weitere Operationen der TCP des Fahrzeuges bestimmt werden.

Die Idee zu diesem Prinzip basiert auf der Nachkalibrierung des 13-Parameter-Modells (vgl. [Meisel (1994)] S. 43-49). Das vollständige Konzept wird in Abschnitt 4.3 erläutert.

Ein anderes Beispiel für die absolute Bestimmung des TCP wird in einer analysierten Arbeit zu diesem Thema beschrieben (vgl. [Hallenberg (2007)]). Dort wird jedoch eine externe Kamera verwendet womit es sich nicht um ein abgeschlossenes System handelt und somit nicht für mobile Systeme geeignet ist. Bei diesem Ansatz wird ein Roboter mit einer Kamera beobachtet wodurch die relative Beziehung zwischen Kamera und TCP feststellbar ist.

3.2.2 Methoden zur Hand-Eye-Kalibrierung

Hier werden verschiedene Methoden zur HEC untersucht. Diese Thematik weist sehr viele und vielseitige Lösungen auf, wobei sich das Konzept der meisten Lösungen stark ähnelt. In dieser Recherche wird aus diesem Grund nur die grundsätzliche Technik der HEC analysiert. Obwohl die HEC Methode in dieser Arbeit vermieden wird, soll ein Eindruck von der Komplexität der HEC vermittelt werden.

Die folgende Erläuterung der Technik erfolgt auf Grundlage verschiedener Arbeiten zum Thema HEC (vgl. [Zhuang (Aug 1998)], [Wei u. a. (Feb 1998)], [Tsai und Lenz (Jun 1989)], [Ma (Feb 1996)] und [Shiu und Ahmad (Feb 1989)]). Hervorzuhebende Besonderheiten der einzelnen Ansätze werden explizit genannt. Die HEC stammt hauptsächlich aus dem Bereich der Robotik, dadurch ist oft das Ziel, die Pose eines Greifers (Roboterarm) zu bestimmen und nicht die einer Plattform. Das Prinzip lässt sich jedoch analog dazu auf das in dieser Arbeit behandelte Thema anwenden.

Das Ergebnis jeder HEC ist die relative Rotation und Translation zwischen dem KS der Kamera und dem der Plattform im Weltkoordinatensystem. Durch die Transformationsparameter kann mit der Kenntnis der Kamera-Pose die Plattform-Pose berechnet werden. Als Voraussetzung für dieses Vorgehen gilt, dass die Kamera fest mit dem relativen Bezugspunkt (Greifer oder Plattform) verbunden ist. Zudem darf die Position der Kamera bezüglich des Greifers oder der Plattform nach der Kalibrierung nicht geändert werden. In diesem Fall würde die relative Beziehung zueinander nicht mehr mit der im Vorfeld bestimmten Beziehung übereinstimmen.

Alle untersuchten Ansätze weisen die folgenden Gemeinsamkeiten auf:

- Es werden mehrere Positionen (mindestens zwei) angefahren, von denen die exakten Koordinaten im Bezugssystem (meist die Roboterplattform) bekannt sind. Diese Positionen müssen sich durch Translation und Rotation unterscheiden und sollten möglichst große Unterschiede aufweisen, um eine hohe Genauigkeit bei der Kalibrierung erzielen zu können (vgl. Abbildung 3.3).
- Bei jeder Position wird die Kamera-Pose bezüglich des Kalibrierobjektes ermittelt. Somit entstehen Wertepaare aus 3D-Koordinaten die sich zum einen aus den Koordinaten des Greifers im Roboter-KS und zum anderen aus den Koordinaten der Kamera bezüglich des Kalibrier-KS zusammensetzen.
- Die Berechnung der Transformationsparameter zur Beschreibung der Beziehung zwischen Plattform und Kamera erfolgt über das aus den Koordinatenpaaren aufgestellte Gleichungssystem.

Es folgt eine Erläuterung der jeweiligen Besonderheiten der analysierten Methoden zur HEC.

[Tsai und Lenz (Jun 1989)] Dieser Ansatz zählt zu den schnellsten und genauesten Techniken der HEC. Hier werden mehr als zwei Stationen angefahren. Zudem werden Positionspaare gebildet, die weit auseinander lokalisiert sind. Im Folgenden werden die Koordinatentransformationen in den Gleichungen 3.1 bis 3.5 definiert und später in einem Gleichungssystem zusammengeführt. Die einzelnen Transformationsgleichungen beziehen sich auf die in Abbildung 3.4 mit Pfeilen dargestellten Transformationen. Die Indizes

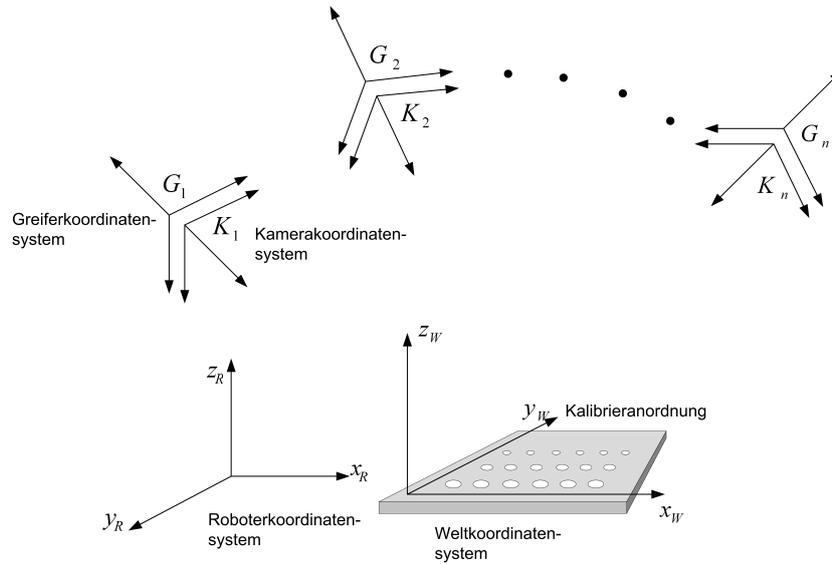


Abb. 3.3: Basis Setup für die Hand-Eye-Kalibrierung

i und j stehen für die verschiedenen Kamera- und Greiferpositionen, während K , G und W die Koordinatensysteme der Kamera, des Greifers und der Welt symbolisieren.

\mathbf{H}_{Gi} Transformation von Greiferposition G_i in das Roboter-KS

$$\mathbf{H}_{Gi} \equiv \mathbf{R}_{Gi} \cdot \begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix} + \vec{t}_{Gi} \quad (3.1)$$

\mathbf{H}_{Kj} Transformation vom Welt-KS zur Kameraposition K_j

$$\mathbf{H}_{Kj} \equiv \mathbf{R}_{Kj} \cdot \begin{pmatrix} x_K \\ y_K \\ z_K \end{pmatrix} + \vec{t}_{Kj} \quad (3.2)$$

\mathbf{H}_{Gij} Transformation von Greiferposition G_i zu Greiferposition G_j

$$\mathbf{H}_{Gij} \equiv \mathbf{R}_{Gij} \cdot \begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix} + \vec{t}_{Gij} \quad (3.3)$$

\mathbf{H}_{Kij} Transformation von Kameraposition K_i zu Kameraposition K_j

$$\mathbf{H}_{Kij} \equiv \mathbf{R}_{Kij} \cdot \begin{pmatrix} x_K \\ y_K \\ z_K \end{pmatrix} + \vec{t}_{Kij} \quad (3.4)$$

\mathbf{H}_{KG} Transformation von Kameraposition K_i zu Greiferposition G_i

$$\mathbf{H}_{KG} \equiv \mathbf{R}_{KG} \cdot \begin{pmatrix} x_K \\ y_K \\ z_K \end{pmatrix} + \vec{t}_{KG} \quad (3.5)$$

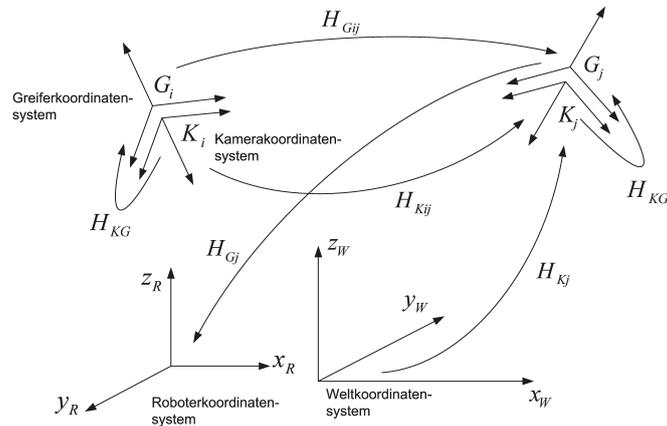


Abb. 3.4: Beziehung der Transformationen zu den Koordinatensystemen

[Zhuang (Aug 1998)] In dieser Arbeit wird ein linearer und ein iterativer Ansatz vorgestellt. Der iterative Ansatz ist aufgrund seiner höheren Genauigkeit von größerer Bedeutung und wird daher näher betrachtet. Diese Lösung baut grundlegend auf den Erkenntnissen von [Tsai und Lenz (Jun 1989)] auf, jedoch wird davon ausgegangen, dass die Umsetzung der HEC ebenso wie die Lösung eines Gleichungssystems aus Matrizen, durch die Gleichung

$$\mathbf{A}_i \mathbf{X} = \mathbf{X} \mathbf{B}_i \quad i = 1, 2. \quad (3.6)$$

abbildbar ist. Dabei entspricht \mathbf{A} der Änderung der Greiferposition, \mathbf{B} der absoluten Kameraposition im Welt-KS und \mathbf{X} der Kameraposition relativ zum Robotergreifer. Das Besondere an dieser Lösung ist, dass bei mindestens zwei anzufahrenden Positionen nur eine Rotation um eine Achse notwendig ist. Das heißt, dass bei der vorzunehmenden Rotation nur ein Freiheitsgrad benötigt wird. Dieser Fakt vereinfacht den Kalibrierprozess für den Anwender dieser Technik.

[Ma (Feb 1996)] Diese Methode sticht durch die komplette HEC und Bestimmung der internen und externen Parameter in einem Prozess hervor. Demzufolge müssen im Vorwege keine Kameraparameter bestimmt oder bereitgestellt werden. Es ist ebenso die einzelne Kalibrierung zur Erlangung der Kameraparameter durchführbar. Für die HEC werden wiederholt zwei Positionen inklusive Rotation benötigt, während für die Kalibrierung der Kamera mindestens sechs Translationen und Rotationen notwendig sind. Bei der Ermittlung der externen Kameraparameter wird das Lochkameramodell verwendet, ähnlich wie das in dieser Arbeit angewandte Messkammermodell.

[Wei u. a. (Feb 1998)] Bei dieser Variante handelt es sich um eine komplett eigenständige Lösung für die Selbst-Kalibrierung des kompletten Systems. Hervorzuheben ist die Eigenschaft, dass keine metrischen Kontrollpunkte notwendig sind. Zudem werden die Startwerte für das resultierende nicht-lineare Gleichungssystem automatisch gefunden. Außerdem wird die Linsenverzeichnung der installierten Kamera im Modell berücksichtigt, wodurch die Genauigkeit gegenüber den anderen Methoden verbessert wird. Wie auch schon bei einem anderen Ansatz werden keine Kameraparameter vorausgesetzt (vgl. [Ma (Feb 1996)]), um die Anwendung der Technik in der Praxis zu vereinfachen.

An dieser Stelle wird noch einmal darauf hingewiesen, dass die HEC grundsätzlich nicht verwendet wird, weil alle untersuchten Ansätze dieses Prinzipes ein aufwendiges Kalibrierverfahren mit mehreren Translationen und Rotationen voraussetzen. Die Anforderung der **einfachen Kalibrierbarkeit** wird nicht erfüllt.

4 Mathematische und konzeptionelle Realisierung

Nach der Einführung in die mathematischen und physikalischen Modelle der kamerabasierten Präzisionsnavigation in Kapitel 2 erfolgt in diesem Kapitel die Beschreibung der mathematischen Realisierung und der Konzeption. Dazu gehört

- die Bestimmung der Kamera-Pose,
- eine detaillierte Beschreibung der durchgeführten Linsenverzeichnungskorrektur,
- die absolute Bestimmung der Plattform-Pose,
- das Konzept des Raummodells in Zusammenhang mit der Passmarken-Identifikation
- sowie die Planung und Verfolgung von Bahnen zum autonomen Abfahren einer definierten Strecke.

Eine Gesamtübersicht der verwendeten Teilkonzepte und deren Zusammenhang wird in Abbildung 4.1 dargestellt.

Eine handelsübliche Digitalkamera, die auf der mobilen Plattform installiert ist, dient als *Bildquelle* und liefert periodisch 2D-Abbilder der Umgebung. Der *Tracker*³ hat die Aufgabe Passmarken in dem gelieferten Abbild zu finden und diese zu identifizieren. Die 2D-Bildkoordinaten der Passmarken werden in Kombination mit der korrespondierenden Identifikationsnummer (ID) an die *Posenbestimmung* weitergegeben. Die *Posenbestimmung* nimmt unter Verwendung des *Raummodells* eine Zuordnung der 2D-Koordinaten der Passmarken zu deren 3D-Koordinaten im Welt-KS vor. Zudem werden die verzeichneten 2D-Bildkoordinaten mit Hilfe der *Linsenverzeichnungskorrektur* korrigiert, so dass die verzeichnungsfreien Bildpunkte der Passmarken weiter verwendet werden können. Während der ersten *Posenbestimmung* ist eine *Initialisierung* des Systems notwendig. In dieser Phase wird die Pose der Kamera näherungsweise als Startwert angegeben, die genaue Pose der Kamera hingegen wird berechnet. Zudem wird während der initialen *Posenbestimmung* die zuvor eingegebene Plattform-Pose in Zusammenhang mit der errechneten Kamera-Pose gespeichert und somit die Grundlage für die absolute *Vehicle Center Point (VCP)-Pose* gelegt. Das Ergebnis jedes Zyklus der *Posenbestimmung* ist die *Pose* der Plattform. Wenn zusätzlich eine Zielliste im Bahnplanungs- und Verfolgungsmodul angegeben wurde, werden die Ziel-Posen nacheinander angefahren.

³Tracker - Bildverarbeitungs-Modul zur Erkennung und Identifikation von Passmarken in Bildern, eingebunden als Drittanbieter-Software-Komponente

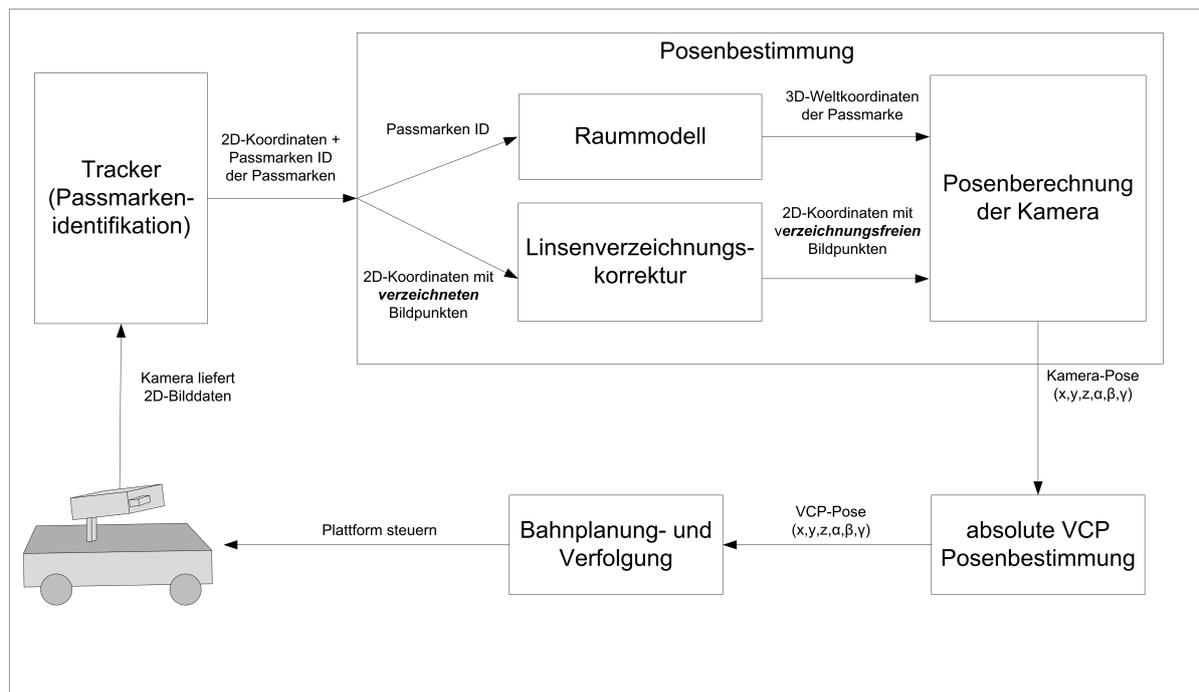


Abb. 4.1: Gesamtübersicht zum Konzept der kamerabasierten Präzisionsnavigation

4.1 Bestimmung der Kamera-Pose in Bezug zum Weltkoordinatensystem

Die Kamera-Pose bezeichnet die Position und die Orientierung des Projektionszentrums der installierten Kamera in Bezug zum Welt-KS. Die Posenbestimmung erfolgt durch 3D-Bildverarbeitung mit mittels Photogrammetrie, was die Verwendung eines geeigneten Kameramodells voraussetzt. Das in Abschnitt 2.1 eingeführte Kameramodell zur mathematischen Beschreibung der Abbildung von Raumpunkten auf die Bildebene hat sich bereits bewährt (vgl. [Föhr (1990)] und [Meisel (1994)]) und wird aufgrund der guten Erfahrungen in dieser Arbeit verwendet. In diesem Abschnitt werden das Konzept der Posenbestimmung einer handelsüblichen CCD-Kamera, die durchzuführenden Schritte sowie die mathematischen Hintergründe zur Posenbestimmung thematisiert.

Aufgrund der essentiellen Bedeutung des Kameramodells für die Posenbestimmung, wird das in [Meisel (1994)] detailliert beschriebene Modell hier noch einmal zusammengefasst. Gleichzeitig wird in diesem Zusammenhang das Messkammermodell erläutert, da das Kameramodell auf dem Messkammermodell aufbaut.

Bei der Vermessung von Raumpunkten, mit Hilfe von photogrammetrischen Mitteln, werden die Koordinaten der Raumpunkte in Koordinaten des Welt-KS bestimmt. Das Welt-KS stimmt im allgemeinen nicht mit dem Kamerakoordinatensystem (Projektionszentrum-KS) überein. In Abbildung 4.2 und 4.3 wird der Zusammenhang zwischen den KS beschrieben mit Hilfe von:

\vec{x}_W	Raumpunkt, beschrieben in Weltkoordinaten
\vec{x}_K	Raumpunkt, beschrieben in Kamerakoordinaten
\vec{t}	<i>Translationsvektor</i> : Position des Projektionszentrums P in Weltkoordinaten
\mathbf{R}	<i>Rotationsmatrix</i> : Verdrehung der beiden Koordinatensysteme gegeneinander

Mit der Gleichung 4.1 wird die vorzunehmende perspektivische Projektion, mit der die Koordinaten \vec{x}_W zunächst in Koordinaten des Projektionszentrum-KS \vec{x}_K überführt werden, beschrieben.

$$\vec{x}_W = \mathbf{R}(\vec{x}_W - \vec{t}) = \mathbf{R} \cdot \vec{x}'_W \quad (4.1)$$

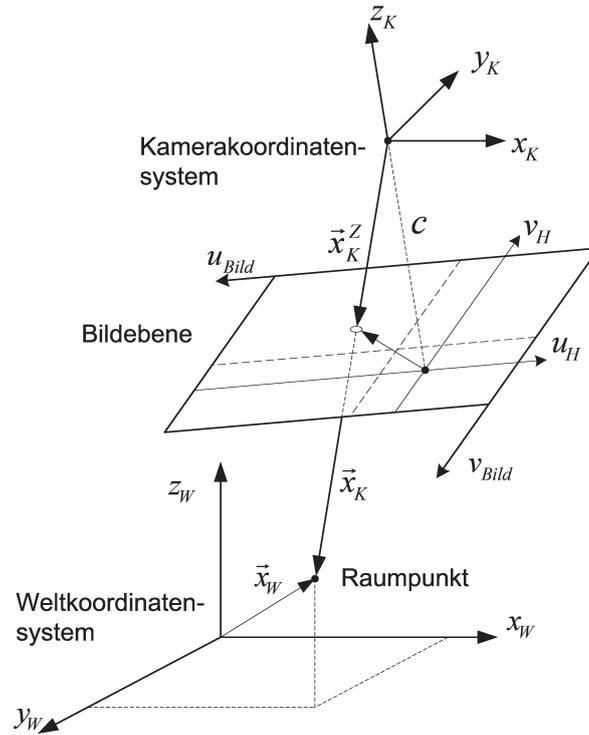


Abb. 4.2: Zusammenhang von Welt- und Kamerakoordinatensystem

Die gesamte Rotation von einem KS in ein anderes erfolgt in drei Einzelschritten, jeweils ein Schritt pro Koordinatenachse. In diesem Fall stellen die folgenden drei Gleichungen 4.2 bis 4.4 die Rotation dar. Eine Visualisierung der Rotation wird in Abbildung 4.3 vorgenommen.

1. Koordinaten um die x' -Achse gedreht (α)

$$\vec{x}'' = (x'', y'', z'')^T = \mathbf{R}_\alpha \cdot \vec{x}' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix} \cdot (x', y', z')^T \quad (4.2)$$

2. Koordinaten um die y'' -Achse gedreht (β)

$$\vec{x}''' = (x''', y''', z''')^T = \mathbf{R}_\beta \cdot \vec{x}'' = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & \sin\beta \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \cdot (x'', y'', z'')^T \quad (4.3)$$

3. Koordinaten um die z''' -Achse gedreht (γ)

$$\vec{x}_K = (x_K, y_K, z_K)^T = \mathbf{R}_\gamma \cdot \vec{x}''' = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot (x''', y''', z''')^T \quad (4.4)$$

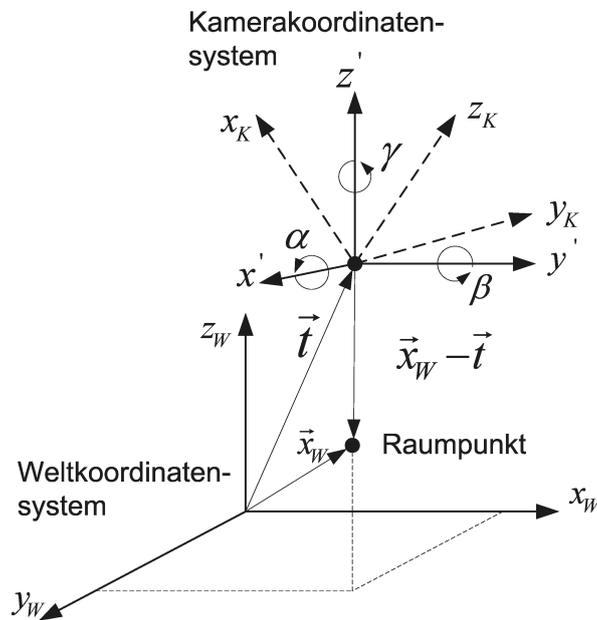


Abb. 4.3: Kamerallage im Weltkoordinatensystem

Zusammenfassend ergibt sich aus den drei Teildrehungen die Gesamttrotationsmatrix (Gl. 4.5), welche orthogonal ist und die Zeilen und Spaltenvektoren ein Orthonormalsystem bilden.

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_\gamma \cdot \mathbf{R}_\beta \cdot \mathbf{R}_\alpha = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \\ &= \begin{pmatrix} \cos\beta \cos\gamma & \cos\alpha \sin\gamma + \sin\alpha \sin\beta \cos\gamma & \sin\alpha \sin\gamma - \cos\alpha \sin\beta \cos\gamma \\ -\cos\beta \sin\gamma & \cos\alpha \cos\gamma - \sin\alpha \sin\beta \sin\gamma & \sin\alpha \cos\gamma + \cos\alpha \sin\beta \sin\gamma \\ \sin\beta & -\sin\alpha \cos\beta & \cos\alpha \cos\beta \end{pmatrix} \end{aligned} \quad (4.5)$$

Die perspektivische Transformation von Raumpunkten auf die Bildebene wird durch die Zentralprojektion beschrieben. Dabei werden die Koordinaten bezüglich des Projektionszentrum-KS der Kamera angegeben. In Abbildung 4.4 wird der Abbildungsvorgang eines Raumpunktes dargestellt. Die Kammerkonstante c beschreibt den Abstand vom Projektionszentrum zur Bildebene, während die optische Achse den senkrecht auf die Bildebene auftreffenden Abbildungsstrahl bezeichnet. Der aus Abbildung 4.4 resultierende Strahlensatz wird genutzt, um die Projektionszentrumkoordinaten \vec{x}_K^Z mit der Gleichung 4.6 zu berechnen.

$$\vec{x}_K^Z = -c \cdot \frac{(x_K, y_K, z_K)}{z_K} = -c \cdot \frac{\vec{x}_K}{(0, 0, 1) \cdot \vec{x}_K} \quad (4.6)$$

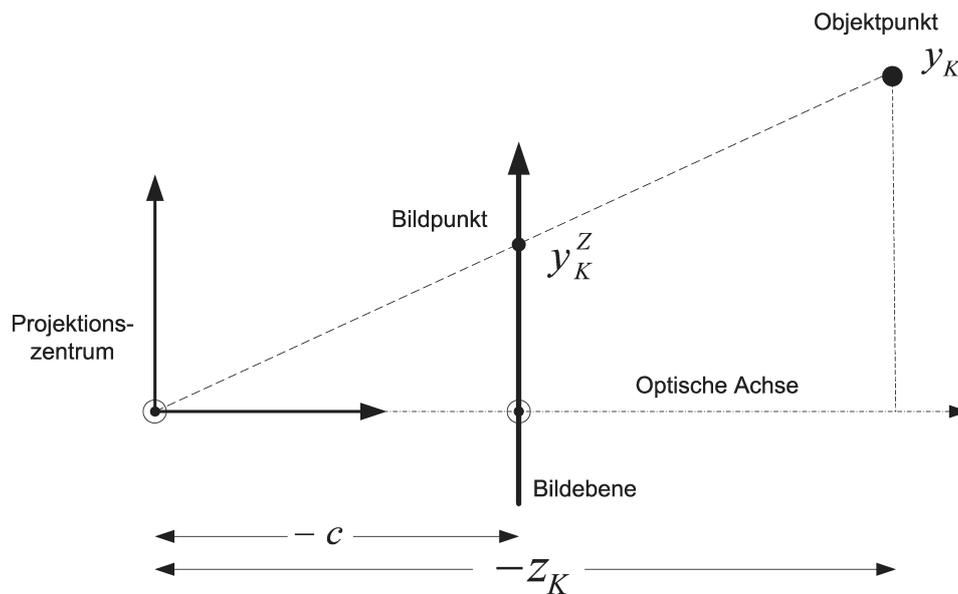


Abb. 4.4: Zentralprojektion eines Raumpunktes auf die Bildebene

Die Bildkoordinaten \vec{u}_H ergeben sich somit aus

$$u_H = x_K^Z \quad v_H = y_K^Z. \quad (4.7)$$

4.1.1 Gesamtmodell der perspektivischen Abbildung

Die Gleichung 4.8 beschreibt den gesamten Abbildungsvorgang einer Messkammer.

$$\vec{x}_K^Z = -c \cdot \frac{\mathbf{R} \cdot (\vec{x}_W - \vec{t})}{(0,0,1) \cdot \mathbf{R} \cdot (\vec{x}_W - \vec{t})} \quad (4.8)$$

mit $\vec{x}_K^Z = (u_H, v_H, -c)$ Projektionszentrumkoordinaten eines in die Bildebene projizierten Punktes

Zur Beschreibung des aus 13 Parametern bestehenden Gesamtkameramodells werden die sechs äußeren Kameraparameter und die sieben inneren Parameter benötigt.

- $\alpha, \beta, \gamma, t_x, t_y, t_z$ (äußere Parameter)
- $c, \Delta u_H, \Delta v_H, K_1, K_2, P_1, P_2$ (innere Parameter)

Diese Parameter fließen in die resultierenden Gleichungen 4.9 und 4.10 zur Berechnung der Bildkoordinaten ein.

$$u_i = \Delta u_H - c \cdot \frac{r_{11}(x - t_x) + r_{12}(y - t_y) + r_{13}(z - t_z)}{r_{31}(x - t_x) + r_{32}(y - t_y) + r_{33}(z - t_z)} \quad (4.9)$$

$$v_i = \Delta v_H + c \cdot \frac{r_{21}(x - t_x) + r_{22}(y - t_y) + r_{23}(z - t_z)}{r_{31}(x - t_x) + r_{32}(y - t_y) + r_{33}(z - t_z)} \quad (4.10)$$

Da in dieser Arbeit nicht die Bildkoordinaten, sondern die Kameraposition und Orientierung zu bestimmen sind, ist eine Umstellung der Gleichung 4.9 und 4.10 zur Berechnung der äußeren Kameraparameter notwendig. Es sind sechs Unbekannte in jeder der Gleichung enthalten, was eine direkte Umstellung nicht erlaubt, stattdessen ist zur Berechnung der gesuchten Parameter die nichtlineare Ausgleichsrechnung anzuwenden. Details zur mathematischen Lösung sind dem Unterabschnitt 4.1.3 zu entnehmen.

In diesen Gleichungen wird zunächst von einer verzeichnungsfreien Abbildung der Raumpunkte ausgegangen. Die in der Realität auftretende Linsenverzeichnung wird durch die in Abschnitt 4.2 beschriebene Linsenverzeichnungskorrektur ausgeglichen.

4.1.2 Vorgehensweise bei der Posenbestimmung der Kamera

Die Bestimmung der Pose erfolgt in fünf Schritten. Vorausgesetzt wird dabei die vorherige initiale Kalibrierung der verwendeten Kamera zur Bestimmung der internen Parameter und deren Eingabe in das System.

1. Zunächst muss von der Umwelt ein 2D-Abbild mit Hilfe einer Kamera erzeugt werden. Dieses Abbild enthält Passmarken, die zur Bestimmung der Pose herangezogen werden.
2. Erkennung von mindestens drei Passmarken im 2D-Abbild, von denen die Weltkoordinaten bekannt sind. Es ist essentiell, dass immer ein Paar aus 2D-Bildkoordinaten und 3D-Weltkoordinaten für jede Passmarke bereit gestellt wird. Eine Beschreibung der Identifikation und Zuordnung der Koordinatenpaare erfolgt in Abschnitt 4.4.
3. Das Kameramodell arbeitet mit metrischen Werten. Daher müssen die Koordinaten, die als Pixelkoordinaten vorliegen, in das metrische System umgerechnet werden. Hierzu müssen die Abmessungen des lichtempfindlichen Sensors bekannt sein.
4. Linsenverzeichnungskorrektur der mindestens drei korrespondierenden Bildpunkte zur Erlangung einer höheren Präzision. Durch ungenaue Optik an den Bildaufnahmegegeräten treten in der Regel Verzeichnungen auf. Diese führen dazu, dass die Bildpunkte nicht korrekt auf der Bildebene aufgezeichnet werden. Durch die zuvor vorgenommene Kalibrierung sind diese Fehler bekannt und können durch die in Abschnitt 4.2 beschriebene Linsenverzeichnungskorrektur behoben werden. Durch diese Korrektur wird die Genauigkeit der Messung signifikant verbessert.
5. Berechnung des Translationsvektors (Position) mit den Parametern t_x , t_y , t_z und die Berechnung der Rotationswinkel (Orientierung) α , β und γ . Das Ergebnis enthält die Kameraposition- und Orientierung im Welt-KS. Dieser Schritt impliziert die perspektivische Transformation der abgebildeten 2D-Punkte in 3D-Punkte aus Sicht der Kamera (Kamerakoordinatensystem). Anschließend erfolgt die Berechnung der Kamerakoordinaten im Weltkoordinatensystem, durch die Anwendung der perspektivischen Projektion (vgl. [Meisel (1994)], [Haralick u. a. (1989)]).

Diese Schritte werden kontinuierlich angewendet, um zu jedem Zeitpunkt die Pose der Kamera zu berechnen. Die Geschwindigkeit der Berechnung ist vor allem von der Leistungsstärke des Bildverarbeitungsalgorithmus abhängig, welcher die Passmarken identifiziert. Der Zeitaufwand zur Berechnung der Pose ist im Vergleich dazu sehr gering.

In der Abbildung 4.5 ist ein Blockdiagramm zur Visualisierung der Vorgehensweise bei der Posenbestimmung dargestellt.

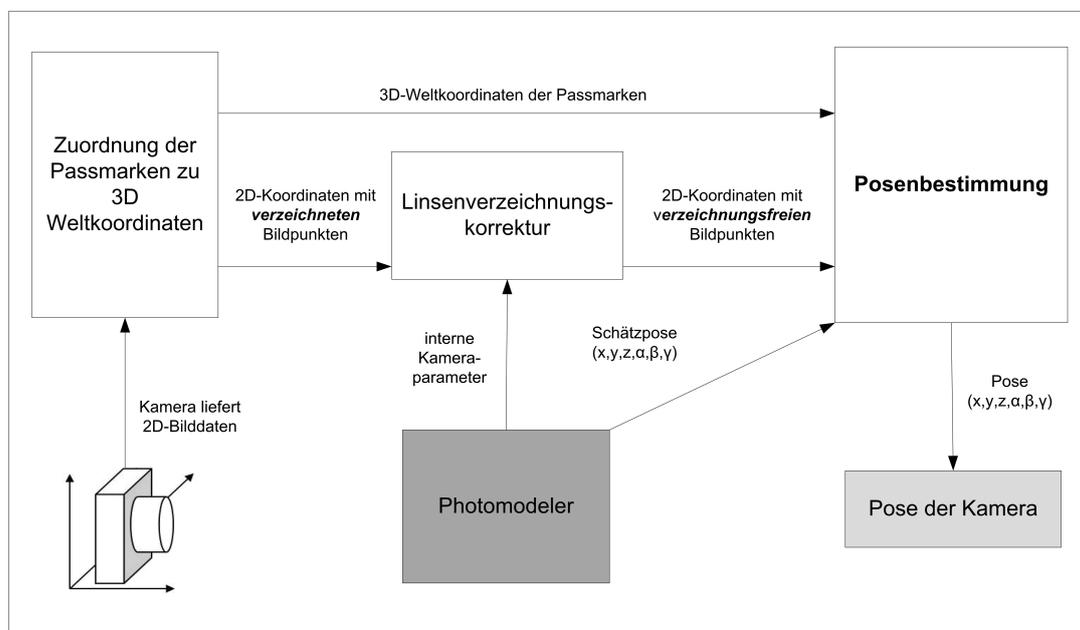


Abb. 4.5: Blockdiagramm zur Vorgehensweise bei der Posenbestimmung der Kamera

4.1.3 Mathematische Lösung der Posenbestimmung

Die Gleichungen 4.9 und 4.10 bilden die Grundlage für die Berechnung der Kamera-Pose. Da die Umstellung dieser Gleichungen nach den gesuchten Parametern nicht direkt durchführbar ist, muss zur Auflösung die nicht lineare Ausgleichung angewendet werden. Hierzu werden die Gleichungen zunächst um den Index i zur Verwendung von Passmarkenkoordinaten erweitert und nach Null umgestellt, um ein Gleichungssystem zur Nullstellensuche nach Newton aufstellen zu können (vgl. Anhang B.1). Für n Passmarken in der Bildebene resultieren jeweils zwei nicht lineare Funktionen:

$$f_i^u(t_x, t_y, t_z, \alpha, \beta, \gamma) = \Delta u_H - c \cdot \frac{r_{11} \cdot (x_i - t_x) + r_{12} \cdot (y_i - t_y) + r_{13} \cdot (z_i - t_z)}{r_{31} \cdot (x_i - t_x) + r_{32} \cdot (y_i - t_y) + r_{33} \cdot (z_i - t_z)} = 0 \quad (4.11)$$

$$f_i^v(t_x, t_y, t_z, \alpha, \beta, \gamma) = \Delta v_H + c \cdot \frac{r_{21} \cdot (x_i - t_x) + r_{22} \cdot (y_i - t_y) + r_{23} \cdot (z_i - t_z)}{r_{31} \cdot (x_i - t_x) + r_{32} \cdot (y_i - t_y) + r_{33} \cdot (z_i - t_z)} = 0 \quad (4.12)$$

Diese Gleichungen bilden die Voraussetzung für die folgende Schritte. Gesucht sind die Variablen t_x, t_y, t_z sowie die Winkel α, β, γ aus der Rotationsmatrix. Diese Variablen stellen die Pose der Kamera dar. Die 3D-Koordinaten (x_i, y_i, z_i) im Welt-KS sowie die 2D-Bildkoordinaten (u_i, v_i) der Raumpunkte sind dagegen bekannt. Aufgrund der sechs Unbekannten im System werden mindestens drei Raumpunkte benötigt, um sechs Gleichungen und somit das Gleichungssystem zur eindeutigen Lösung aufzustellen zu können.

Die gesuchten Parameter werden durch die Anwendung des Standardansatzes (vgl. [Meisel (1994)] Kapitel C.1). berechnet.

$$\vec{\varepsilon} = A \cdot \vec{\xi} - \vec{l} \quad (4.13)$$

Im nächsten Schritt wird die Jacobi-Matrix (Ableitungsmatrix) \mathbf{A} für die näherungsweise Berechnung aufgestellt. Die Jacobi-Matrix enthält alle ersten partiellen Ableitungen der Funktionen f_1^u und f_i^v , jeweils nach den Parametern $t_x, t_y, t_z, \alpha, \beta$ und γ . Diese werden zeilenweise in die Matrix \mathbf{A} eingesetzt (vgl. Gl. 4.14 bis 4.16). Die Ableitungen der jeweiligen Funktionen können zum Beispiel mit der Mathematik-Software *Maple* gebildet werden. In Anhang B.2 sind die komplexen Ableitungen zu finden.

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f_1^u}{\partial t_x} & \frac{\partial f_1^u}{\partial t_y} & \frac{\partial f_1^u}{\partial t_z} & \frac{\partial f_1^u}{\partial \alpha} & \frac{\partial f_1^u}{\partial \beta} & \frac{\partial f_1^u}{\partial \gamma} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n^u}{\partial t_x} & \frac{\partial f_n^u}{\partial t_y} & \frac{\partial f_n^u}{\partial t_z} & \frac{\partial f_n^u}{\partial \alpha} & \frac{\partial f_n^u}{\partial \beta} & \frac{\partial f_n^u}{\partial \gamma} \\ \frac{\partial f_1^v}{\partial t_x} & \frac{\partial f_1^v}{\partial t_y} & \frac{\partial f_1^v}{\partial t_z} & \frac{\partial f_1^v}{\partial \alpha} & \frac{\partial f_1^v}{\partial \beta} & \frac{\partial f_1^v}{\partial \gamma} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n^v}{\partial t_x} & \frac{\partial f_n^v}{\partial t_y} & \frac{\partial f_n^v}{\partial t_z} & \frac{\partial f_n^v}{\partial \alpha} & \frac{\partial f_n^v}{\partial \beta} & \frac{\partial f_n^v}{\partial \gamma} \end{pmatrix} \quad (4.14)$$

Die gesuchten Parameter werden durch den Vektor $\vec{\xi}$ beschrieben:

$$\vec{\xi} = \begin{pmatrix} t_{x_{neu}} - t_x \\ t_{y_{neu}} - t_y \\ t_{z_{neu}} - t_z \\ \alpha_{neu} - \alpha \\ \beta_{neu} - \beta \\ \gamma_{neu} - \gamma \end{pmatrix} \quad (4.15)$$

Der Vektor \vec{l} enthält die umgestellten Gleichungen zur Berechnung der Bildkoordinaten, für jeden Punkt zwei Gleichungen. Das ergibt bei n Punkten den folgenden Vektor.

$$\vec{l} = \begin{pmatrix} -f_1^u(t_x, t_y, t_z, \alpha, \beta, \gamma) \\ -f_n^u(t_x, t_y, t_z, \alpha, \beta, \gamma) \\ \vdots \\ -f_1^v(t_x, t_y, t_z, \alpha, \beta, \gamma) \\ -f_n^v(t_x, t_y, t_z, \alpha, \beta, \gamma) \end{pmatrix} \quad (4.16)$$

Die Lösung des Ausgleichsproblems erfolgt durch die Methode der kleinsten Fehlerquadratsumme in Vektor $\vec{\xi}$. Als Verfahren zur Lösung ist die QR-Zerlegung mit der Householdertransformation oder das Singular Value Decomposition (SVD) Verfahren zur singulären Wertzerlegung anwendbar. Details zu diesen Verfahren sind in Mathematik-Fachbüchern nachzulesen (vgl. [Herrmann (2004)] S.60-64 und [Press u. a. (2007)] S.65-75).

Für die erste Posenbestimmung müssen ein Startwerte für alle sechs gesuchten Parameter festgelegt werden. Für diesen Zweck wird die Kamera-Pose geschätzt oder durch den *Photomodeler* ermittelt.

4.2 Linsenverzeichnungskorrektur zur Erlangung von Sub-Pixelgenauigkeit

Die in Abschnitt 2.4 eingeführte Linsenverzeichnung wird durch eine Korrektur ausgeglichen, um die Genauigkeit bei der Posenbestimmung in den Sub-Pixelbereich und somit in die Millimeter-Genauigkeit zu überführen. In diesem Abschnitt wird die mathematische Lösung und die Vorgehensweise bei der Korrektur der Linsenverzeichnung beschrieben.

Bei der Korrektur der Linsenverzeichnung werden die vorliegenden verzeichneten Bildpunkte in verzeichnungsfreie Bildpunkte transformiert. Für diesen Vorgang werden die internen Kameraparameter ($\Delta u_H, \Delta v_H, K_1, K_2, P_1, P_2$) vorausgesetzt. Zur Herstellung der Verzeichnungsfreiheit sind drei Schritte notwendig. In den dargestellten Gleichungen entsprechen u_d und v_d dem verzeichneten Bildpunkt, u_c und v_c dem korrigierten Bildpunkt in metrischer Einheit, während die hochgestellte [#] der Angabe in Pixelwerten entspricht. Die drei Schritte im Einzelnen sind:

1. Transformation ins Bildhauptpunktkoordinatensystem

Die Pixelkoordinaten der angegebenen Bildpunkte werden in ein metrisches Koordinatensystem überführt, in dem der Koordinatenursprung im Bildhauptpunkt liegt. Dieser Schritt ist aufgrund der vom Bildhauptpunkt nach außen hin zunehmenden Verzeichnung notwendig (vgl. [Meisel (1994)] S.24-25).

$$u_d = \left(\frac{p^{\#}}{s_x} - \Delta u_H \right) \cdot 1000 [mm] \quad (4.17)$$

$$v_d = \left(\frac{q^{\#}}{s_y} - \Delta v_H \right) \cdot 1000 [mm] \quad (4.18)$$

mit s_x, s_y Umrechnung Meter in Pixel = Pixel pro Meter Chiplänge/Chipbreite

Die nachfolgende Berechnung der korrigierten Bildpunkte ist auf eine Berechnung in Millimetern normiert. Die Multiplikation mit 1000 entsteht durch die Angabe der Chipgröße in Meter.

2. Verzeichnungskorrektur

Die Korrektur der Bildpunkte erfolgt unter Verwendung der hier dargestellten Gleichungen (vgl. [Hartley und Zisserman (2003)] S.189-193).

$$u_c = u_d \cdot (1 + K_1 \cdot r^2 + K_2 \cdot r^4) + P_1 \cdot (r^2 + 2u_d^2) + 2P_2 \cdot u_d \cdot v_d \quad (4.19)$$

$$v_c = v_d \cdot (1 + K_1 \cdot r^2 + K_2 \cdot r^4) + P_2 \cdot (r^2 + 2v_d^2) + 2P_1 \cdot u_d \cdot v_d \quad (4.20)$$

mit

$$r = \sqrt{u_d^2 + v_d^2} \quad (4.21)$$

3. Rücktransformation ins Bildkoordinatensystem

Nach der Korrektur der Bildpunkte in verzeichnungsfreie Koordinaten folgt schließlich die Rücktransformation aus den metrischen Koordinaten in Pixelkoordinaten.

$$u_c^\# = \left(\frac{u_c}{1000} + \Delta u_H \right) \cdot s_x \quad (4.22)$$

$$v_c^\# = \left(\frac{v_c}{1000} + \Delta v_H \right) \cdot s_y \quad (4.23)$$

4.3 Bestimmung der Plattform-Pose

In dem vorangegangenen Abschnitt wurde erläutert, wie die Lage und Orientierung der Kamera in Bezug zum Welt-KS ermittelt wird. Die resultierende Pose entspricht jedoch nicht der tatsächlichen Lage und Orientierung der mobilen Plattform, da die Kamera auf der Oberfläche der Plattform installiert ist und die Ausrichtung von der Plattform abweicht. Zur Bestimmung der Pose der Plattform ist eine Nachkalibrierung notwendig, um auf der Grundlage der Kamera-Pose die Pose der Plattform zu bestimmen. Der durch die Plattform-Pose beschriebene Punkt wird auch als TCP bezeichnet. Dieser Punkt stellt den Ursprung des Plattform-KS dar und wird bei der Navigation als Fahrzeugposition betrachtet. In diesem Zusammenhang wird an dieser Stelle die Bezeichnung Vehicle Center Point eingeführt. Der VCP entspricht dem TCP, wird aber in diesem Kapitel verwendet, um den Bezug zum Fahrzeug und die Abgrenzung zu Roboterarmen zu verdeutlichen.

In den Erläuterungen, Abbildungen und Vektoren in diesem und den folgenden Unterabschnitten werden die hier aufgeführten KS verwendet. Um das Verständnis zu erleichtern, wird jeweils das Bezugs-KS einer Koordinatenangabe als tiefgestellter großer Buchstabe angegeben. Die dargestellten Abbildungen dienen dem Verständnis der folgenden Überlegungen und Entscheidungen zur Bestimmung der Plattform-Pose (VCP).

- Kamera- oder Projektionszentrumkoordinatensystem (tiefgestelltes K)
- Weltkoordinatensystem (tiefgestelltes W)
- Fahrzeugkoordinatensystem, VCP (tiefgestelltes VCP)

Durch die starre Verbindung der Kamera mit der Plattform bewegt sich die Kamera und damit das KS der Kamera in der gleichen Weise wie die Plattform und dessen KS mit. Nach der Posenbestimmung der Kamera sind die Weltkoordinaten des Projektionszentrum (Kamerakoor-
dinatenursprung) bekannt, jedoch sind die zur Navigation benötigten Weltkoordinaten des VCP zu diesem Zeitpunkt unbekannt. Nach jeder Bewegung ist infolgedessen der VCP in unbekannter Weise zum Welt-KS positioniert und orientiert. Zur Transformation der Kamera-Pose in die VCP-Pose ist die Kenntnis der Rotationsmatrix ΔR und des Translationsvektors $\Delta \vec{t}$ zwischen dem Koordinatenursprung der Plattform VCP und der Kamera K notwendig (vgl. Abbildung 4.6). Da sich die Ermittlung der Translations- und Rotationsparameter als sehr komplex und aufwendig erweist (vgl. Abschnitt 3.2), wird hier die nachfolgend beschriebene Bestimmung der VCP-Pose angewandt.

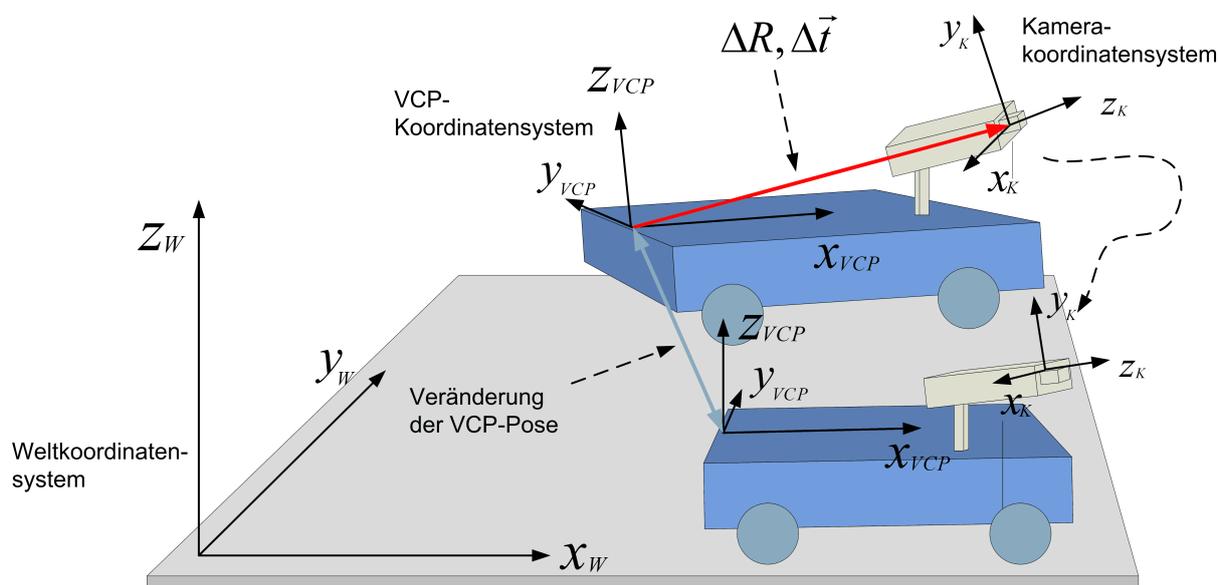


Abb. 4.6: Translations- und Rotationsparameter zwischen Plattform und Kamera

4.3.1 Verschiebung des Weltkoordinatensystem in Bezug zum mitbewegten Koordinatensystem

Die entwickelte Lösung nutzt die starre Kopplung zwischen der Kamera und der mobilen Plattform. Das Kamera-KS und das VCP-KS unterliegen somit den gleichen Translationen und Rotationen. In diesem Zusammenhang wird an dieser Stelle zusätzlich das mitbewegte KS (tiefgestelltes M) eingeführt. Aufgrund der gedachten festen Verbindung zwischen dem mitbewegten KS und der Kamera sowie dem VCP, ist die Pose dieser Komponenten im mitbewegten KS immer konstant. Infolgedessen wird bei der Bewegung der Plattform dieses KS in gleicher Weise mitbewegt.

Die folgenden Abbildungen in diesem Abschnitt (4.7a und 4.7b) dienen zum besseren Verständnis der Lösung und wurden auf zwei Dimensionen reduziert, um das Prinzip vereinfacht darstellen zu können. Die Realisierung erfolgt in drei Dimensionen, um die sechs geforderten Freiheitsgrade bei der Bewegung des Fahrzeuges zu erhalten.

Die Posenbestimmung des VCP basiert auf einem Ansatz, in dem die Plattform zu Beginn der Navigation in eine festgelegte Grundposition gefahren wird. Von dort aus wird die Veränderung der Lage des Welt-KS $_W$ in Bezug zum mitbewegten KS $_M$ nach jeder Posenbestimmung der Kamera ermittelt. Es wird vorausgesetzt, dass der Koordinatenursprung des Welt-KS zum Zeitpunkt einer initialen Posenbestimmung in der Grundposition deckungsgleich mit dem mitbewegten KS ist. Die Veränderung der Translation $\Delta\vec{t}$ und Rotation $\Delta\mathbf{R}$ beschreibt die Bewegung der Plattform. Zur Berechnung der Veränderung sind die Ausgangs-Pose zum Zeitpunkt der initialen Posenbestimmung (\mathbf{R}, \vec{t}) und die aktuelle Pose $(\mathbf{R}^*, \vec{t}^*)$ der Kamera nach der Bewegung notwendig. Unter Verwendung dieser Lösung ist eine Bestimmung der relativen Translations- und Rotationsparameter zwischen der Kamera und der Plattform nicht erforderlich.

Vor dem Beginn der Fahrt des mobilen Systems muss es zunächst in eine definierte Grundposition gebracht werden. Dabei ist auf eine exakte Ausrichtung (Orientierung) des VCP-KS am mitbewegten und zu diesem Zeitpunkt deckungsgleichen Weltkoordinatensystem zu achten. Die Grundposition ist frei wählbar, muss aber bei jeder Berechnung der VCP-Pose angegeben werden. Sie beschreibt die Lage des VCP im mitbewegten KS.

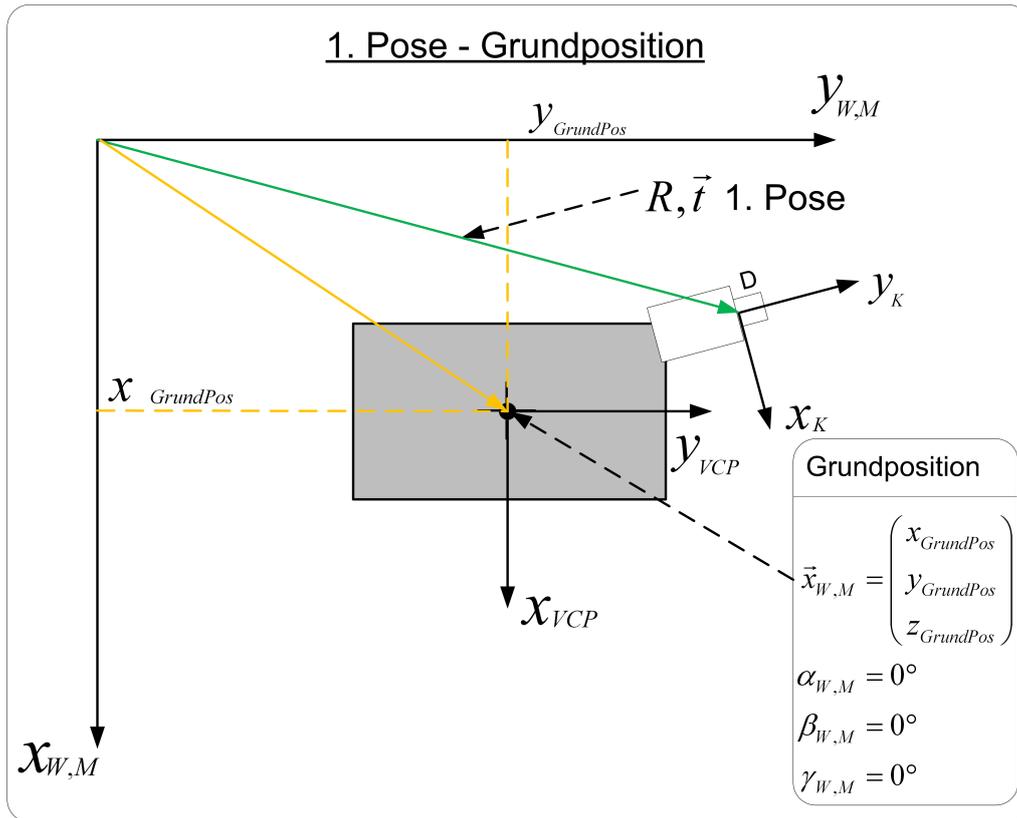
Der beschriebene Ansatz ist nur bei einer starren Kopplung der Kamera mit dem VCP gültig, das heißt dass, die Position und Orientierung der Kamera auf der Plattform nach der initialen Posenbestimmung nicht verändert werden darf. Es folgt eine detaillierte Beschreibung der zweistufigen Posenbestimmung der Plattform.

1. Initiale Posenbestimmung der Kamera in der Grundposition

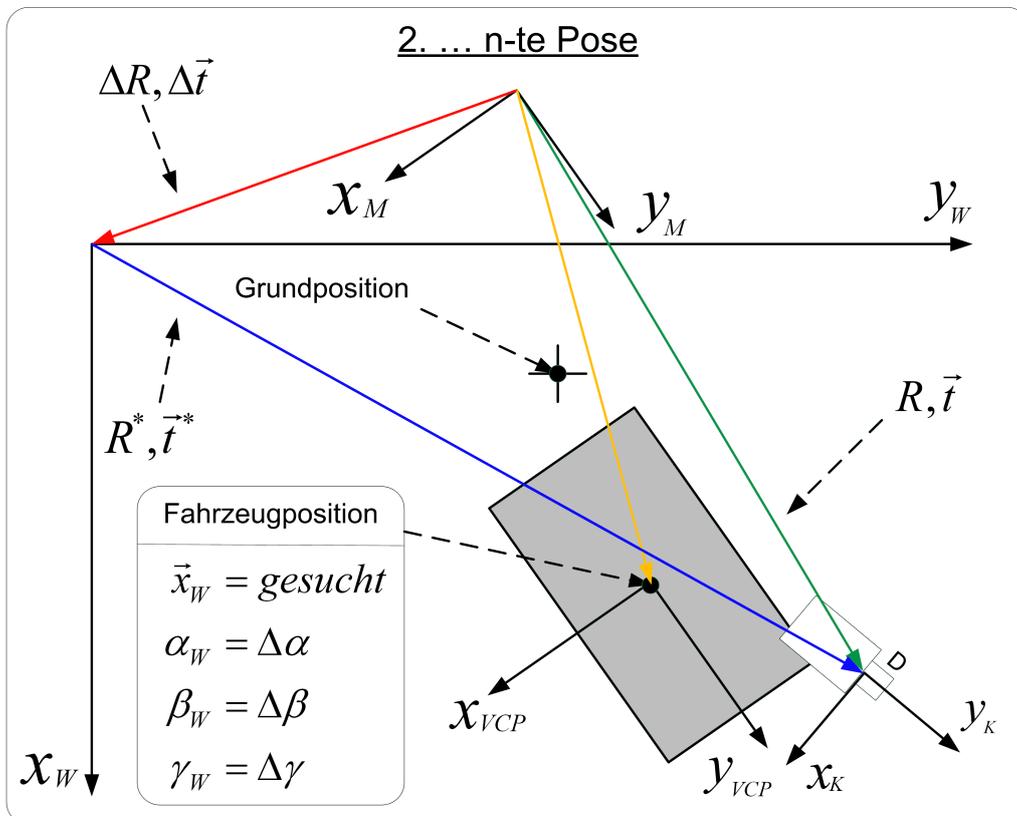
Das Ergebnis der Posenbestimmung in der Grundposition wird durch die Rotationsmatrix \mathbf{R} und den Translationsvektor \vec{t} angegeben, die die Lage und Orientierung der Kamera im Welt-KS beschreiben. Diese Parameter werden bei allen weiteren Posenbestimmungen des Fahrzeuges genutzt, um aus der Kamera-Pose die VCP-Pose zu berechnen. Der VCP der Plattform muss zu diesem Zeitpunkt die Koordinaten der Grundposition einnehmen. In der Grundposition sind der Translationsvektor und die Rotationsmatrix der Verschiebung zwischen Welt- und mitbewegtem KS mit den Ausgangswerten belegt. Während des Zeitpunktes der initialen Posenbestimmung liegen das Welt-KS und das mitbewegte KS exakt übereinander, somit sind die Koordinaten der Grundposition zu diesem Zeitpunkt in beiden Systemen identisch. Durch die starre Verbindung der Kamera mit dem mitbewegten KS und somit auch mit der Plattform, ändert sich die Position und Orientierung der Plattform und der Kamera innerhalb des mitbewegten KS nie. Dieser Vorgang muss immer dann durchgeführt werden, wenn Änderungen an der Kamerainstallation vorgenommen wurden oder der VCP verändert wird. Die Posenbestimmung der Kamera in der Grundposition wird in Abbildung 4.7a visualisiert. Zudem werden im Folgenden die Translationsparameter zum Zeitpunkt der initialen Posenbestimmung dargestellt.

$$\Delta\vec{t} = (0, 0, 0)^T \quad (4.24)$$

$$\Delta\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.25)$$



(a) Initiale Posenbestimmung in der Grundposition



(b) Welt-Rotation und Translation während der Bewegung des Fahrzeuges

Abb. 4.7: Visualisierung der Posenbestimmung des Fahrzeuges

Die Ausrichtung des VCP im Welt- und mitbewegtem KS muss

$$\alpha_{W,M} = \beta_{W,M} = \gamma_{W,M} = 0^\circ$$

entsprechen, während die Position des VCP gleich der Grundposition ist.

$$\vec{x}_{W,M} = \begin{pmatrix} x_{GrundPos} \\ y_{GrundPos} \\ z_{GrundPos} \end{pmatrix}$$

2. Posenbestimmung der Plattform während der Fahrt

Während der Fahrt des Fahrzeuges in eine beliebige Position unabhängig von der Orientierung, wird fortlaufend die Pose der installierten Kamera bestimmt. Die Pose wird durch die ermittelten Transformationsparameter \mathbf{R}^* und \vec{t}^* angegeben. Aus dieser aktuellen Pose der Kamera und der initialen Pose der Kamera in der Grundposition ist die Bestimmung der Verschiebung des Welt-KS zum mitbewegtem KS errechenbar. Die Transformationsparameter $\Delta\mathbf{R}$ und $\Delta\vec{t}$ geben die Lage und Orientierung des Welt-KS im mitbewegtem KS an. Die Rotation des VCP im Welt-KS entspricht der Rotation zwischen dem mitbewegtem und dem Welt-KS, weil die Plattform fest mit dem mitbewegtem KS verbunden ist und somit der gleichen Rotation zum Welt-KS unterliegt. Somit kann die Orientierung des VCP der Rotationsmatrix $\Delta\mathbf{R}$ entnommen werden. Solange die Plattform bei der Bewegung keiner Rotation unterliegt und lediglich Translationen vornimmt, entspricht der Translationsvektor $\Delta\vec{t}$ der Positionsveränderung des VCP bezüglich der Grundposition. Es ist jedoch von einer Rotation der Plattform auszugehen, was einen weiteren Berechnungsschritt zur Darstellung der VCP Koordinaten im Welt-KS erfordert. Die Beschreibung dieses Schrittes wird in Unterabschnitt 4.3.2 vorgenommen. Hier gilt die Gl. 4.26 für die Rotationsmatrix. Die Posenbestimmung der Plattform während der Fahrt wird in der Abbildung 4.7b skizziert. Die Berechnungsschritte und deren Herleitung erfolgt anschließend.

$$\mathbf{R}_W = \Delta\mathbf{R} \quad (4.26)$$

Diese Lösung setzt die Transformation eines in Weltkoordinaten angegebenen Raumpunktes \vec{x}_W in die Koordinaten des mitbewegtem KS, beschrieben durch die Gl. 4.27, voraus (vgl. [Meisel (1994)] S.45-47). Durch die Anwendung dieser Transformation werden die Transformationsparameter für die Bewegung der Welt um das mitbewegte KS ermittelt, während normalerweise von einem festen Welt-KS als Bezug der Bewegungen in der Welt ausgegangen wird.

$$\vec{x}_M = \Delta\mathbf{R} \cdot \vec{x}_W + \Delta\vec{t} \quad (4.27)$$

Für Berechnung der Transformationsparameter wird der Abbildungsvorgang einer Kamera, beschrieben durch die Gleichung (4.28), vorausgesetzt.

$$\vec{x}_K^Z = -c \cdot \frac{\mathbf{R} \cdot (\vec{x}_M - \vec{t})}{(0,0,1) \cdot \mathbf{R} \cdot (\vec{x}_M - \vec{t})} \quad (4.28)$$

Durch die Erweiterung der Gl. 4.28 mit den Transformationsparametern $\Delta\mathbf{R}$ und $\Delta\vec{t}$ (einsetzen der Gl. 4.27 für \vec{x}_M) entsteht die Gleichung:

$$\vec{x}_K^Z = -c \cdot \frac{\overbrace{\mathbf{R} \cdot (\Delta\mathbf{R} \cdot \vec{x}_W + \Delta\vec{t} - \vec{t})}^D}{(0,0,1) \cdot \mathbf{R} \cdot (\Delta\mathbf{R} \cdot \vec{x}_W + \Delta\vec{t} - \vec{t})} \quad (4.29)$$

Es wird nur der Dividend D für die weiteren Schritte betrachtet.

$$D = \mathbf{R} \cdot (\Delta\mathbf{R} \cdot \vec{x}_W + \Delta\vec{t} - \vec{t}) \quad (4.30)$$

Die Rotationsmatrix \mathbf{R} ist orthogonal, so dass

$$\mathbf{R}^T = \mathbf{R}^{-1}, \quad (4.31)$$

die Zeilen- und Spaltenvektoren von \mathbf{R} ein Orthonormalsystem bilden. Es wird der Parameter $\Delta\mathbf{R}$ aus der Gl. 4.30 ausgeklammert, somit wird der Kehrwert von $\Delta\mathbf{R}$ in den zweiten Teil der resultierenden Gleichung eingebettet:

$$\begin{aligned} D &= \underbrace{\mathbf{R} \cdot \Delta\mathbf{R}}_{\mathbf{R}^*} \cdot (\vec{x}_W + \underbrace{\Delta\mathbf{R}^T \cdot (\Delta\vec{t} - \vec{t})}_{-\vec{t}^*}) \\ &= \mathbf{R}^* \cdot (\vec{x}_W - \vec{t}^*) \end{aligned} \quad (4.32)$$

Mit

$$\mathbf{R}^* = \mathbf{R} \cdot \Delta\mathbf{R} \Rightarrow \boxed{\Delta\mathbf{R} = \mathbf{R}^* \cdot \mathbf{R}^T} \quad (4.33)$$

und

$$\underbrace{\vec{t}^*}_{\text{aktuelle Pose}} = (t_x^*, t_y^*, t_z^*)^T = -\Delta\mathbf{R}^T \cdot (\Delta\vec{t} - \underbrace{\vec{t}}_{1.\text{Pose}}) \quad (4.34)$$

werden die Parameter $\Delta\mathbf{R}$ und $\Delta\vec{t}$ im Zuge der Nachbestimmung berechnet. Die Umstellung der Gl. 4.34 nach $\Delta\vec{t}$ erfolgt in den folgenden Schritten:

$$\begin{aligned} \vec{t}^* &= -\Delta\mathbf{R}^T \cdot \Delta\vec{t} + \Delta\mathbf{R}^T \cdot \vec{t} \\ \Delta\vec{t} &= \frac{\Delta\mathbf{R}^T}{\Delta\mathbf{R}^T} \cdot \vec{t} - \frac{\vec{t}^*}{\Delta\mathbf{R}^T} \\ &\boxed{\Delta\vec{t} = \vec{t} - \Delta\mathbf{R} \cdot \vec{t}^*} \end{aligned} \quad (4.35)$$

4.3.2 Abbildung des VCP in Weltkoordinaten

Im Unterabschnitt 4.3.1 wurde die Bewegung des Welt-KS im mitbewegten KS beschrieben. Aufgrund der starren Verbindung des mitbewegten KS mit der mobilen Plattform ist die Translation und Rotation der Plattform in dem Welt-KS somit bekannt. Jedoch ist die zur Navigation benötigte Pose der Plattform (VCP) in Weltkoordinaten nur durch eine zusätzliche Transformation der Koordinaten zur bestimmen. Zur Bestimmung der VCP-Pose wird nun die affine Transformation unter Verwendung der zuvor berechneten Transformationsparameter ($\Delta\vec{t}$ und $\Delta\mathbf{R}$) angewandt.

Eine affine Abbildung umfasst unter anderem elementare Transformationen *Translation* und *Rotation* und beschreibt eine lineare Koordinatentransformation (vgl. [Jähne (2005)]). Sie wird durch die folgende Vektoraddition und Matrizenmultiplikation ausgedrückt und entspricht prinzipiell der bereits in Gl. 4.27 beschriebenen Abbildung eines Weltpunktes in Koordinaten des mitbewegten Systems.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (4.36)$$

Zunächst erfolgt die Erläuterung der affinen Abbildung zur Koordinatentransformation am zuvor bereits verwendeten zweidimensionalen Beispiel aus Abbildung 4.7b, um die Komplexität nicht unnötig zu erhöhen. Die gegebene Grundposition gibt die Lage des VCP im mitbewegten KS an (\vec{t}). Die Orientierung der Plattform muss bei der initialen Kalibrierung exakt der Ausrichtung des Bezugs-KS entsprechen, die Translation dagegen kann frei gewählt werden. Die Transformationsparameter $\Delta\mathbf{R}$ und $\Delta\vec{t}$ werden im Voraus berechnet und sind somit ebenfalls bereits verfügbar. Der Berechnung der gesuchten Weltkoordinaten der Plattform (\vec{t}_W) liegt die hier beschriebene Koordinatentransformation von einem KS in ein anderes zu Grunde. Die hier verwendeten Vektoren und Matrizen entsprechen den im Abschnitt 4.3 eingeführten Komponenten und beziehen sich ebenso auf die dort vorgestellten Abbildungen.

$$\underbrace{\begin{pmatrix} x_M \\ x_M \end{pmatrix}}_{\text{Grundposition VCP}} = \underbrace{\begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}}_{\Delta\mathbf{R}} \cdot \underbrace{\begin{pmatrix} x_W \\ y_W \end{pmatrix}}_{\text{gesuchte VCP-Position in Weltkoordinaten}} + \underbrace{\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}}_{\Delta\vec{t}} \quad (4.37)$$

In diesem Fall sind die Weltkoordinaten des VCP gesucht, was zu einer Umstellung der Gl. 4.37 in die folgende Form führt. Diese Umstellung ist aufgrund der Orthonormalität von \mathbf{R} möglich.

$$\begin{pmatrix} x_W \\ y_W \end{pmatrix} = \begin{pmatrix} x_M \\ y_M \end{pmatrix} - \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \cdot \Delta\mathbf{R}^T \quad (4.38)$$

$$\text{mit (4.31) } \mathbf{R}^T = \mathbf{R}^{-1}$$

Das Ergebnis der Koordinatentransformation ist ein Vektor, der die Position der Plattform im Welt-KS angibt (\vec{t}_W). Des Weiteren werden für die Navigation aber zusätzlich Winkel zur Angabe der Orientierung der Plattform im Welt-KS benötigt. Aufgrund der identischen Ausrichtung der Plattform und des mitbewegten KS entspricht die Orientierung der Plattform der Orientierung des mitbewegten KS in Bezug zum Welt-KS. Die Berechnung der Orientierungswinkel muss separat auf Grundlage der Rotationsmatrix erfolgen und wird in Unterabschnitt 4.3.4 erläutert. Die resultierenden Winkel werden zur Angabe der Orientierung der Plattform verwendet.

Zur Erhaltung der geforderten sechs Freiheitsgrade ist die beschriebene affine Transformation auf drei Dimensionen in folgender Art und Weise anzuwenden:

$$\begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} \quad (4.39)$$

Die zusätzliche Umformung zur Erlangung der Weltkoordinaten der Plattform erfolgt analog zur Gleichung 4.38.

$$\begin{pmatrix} x_W \\ y_W \\ z_W \end{pmatrix} = \left(\begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix} - \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} \right) \cdot \Delta \mathbf{R}^T \quad (4.40)$$

4.3.3 Darstellung von Weltpunkten in VCP-Koordinaten zur Navigation

Im vorherigen Abschnitt wurde die Darstellung der Plattform in Weltkoordinaten erläutert. In diesem Abschnitt wird die Transformation von Weltkoordinaten in Koordinaten des Plattform-KS (VCP) beschrieben, welche ebenfalls mit Hilfe einer affinen Abbildung umgesetzt wird. Mit dieser Darstellungsvariante ist die Angabe von beispielsweise Zielpunkten in Bezug zum VCP realisierbar, was die Navigation zum Zielpunkt vereinfacht.

Zunächst muss die Position des Weltpunktes in Koordinaten des mitbewegten KS berechnet werden (\vec{x}_M). Für diesen Zweck wird die Gleichung 4.37 aus Unterabschnitt 4.3.2 angewendet. Jedoch sind in diesem Fall die Weltkoordinaten (\vec{x}_W) gegeben. Es sind die Koordinaten im mitbewegten KS zu berechnen. In Abbildung 4.8 werden die benötigten Komponenten und deren Zusammenhang noch einmal verdeutlicht.

$$\underbrace{\begin{pmatrix} x_{M_i} \\ y_{M_i} \end{pmatrix}}_{\text{Position im mitb. KS } \vec{x}_M} = \underbrace{\begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}}_{\Delta \mathbf{R}} \cdot \underbrace{\begin{pmatrix} x_{W_i} \\ y_{W_i} \end{pmatrix}}_{\text{Weltpunkt } \vec{x}_W} + \underbrace{\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}}_{\Delta \vec{t}} \quad (4.41)$$

Das Ergebnis der Koordinatentransformation ist die Position des angegebenen Weltpunktes in Koordinaten des mitbewegten KS (\vec{x}_M). Um zu den VCP-Koordinaten des Weltpunktes (\vec{x}_{VCP}) zu gelangen ist ein weiterer Schritt notwendig. Aufgrund der starren Verbindung des mitbewegten KS mit der Plattform (VCP) und der Angabe der Grundposition des VCP im mitbewegten KS ist lediglich die Grundposition des VCP von der errechneten Position abziehen. Voraussetzung für diesen Schritt ist der Bezug der Koordinaten der Grundposition und der errechneten Koordinaten des Weltpunktes (4.42) zum selben KS. Der Orientierungsanteil der Pose des Weltpunktes entspricht dabei wieder den errechneten Winkeln aus der Rotationsmatrix. Dieses Verfahren ist nur anwendbar, wenn das VCP-KS zum Zeitpunkt der initialen Posenbestimmung die identische Ausrichtung wie das mitbewegte KS aufweist. Nur dann entspricht die Rotation bei der Bewegung der Plattform den Rotationen in der Rotationsmatrix.

$$\underbrace{\begin{pmatrix} x_{VCP_i} \\ y_{VCP_i} \end{pmatrix}}_{\vec{x}_{VCP}} = \underbrace{\begin{pmatrix} x_{M_i} \\ y_{M_i} \end{pmatrix}}_{\vec{x}_M} - \underbrace{\begin{pmatrix} x_{Grundpos} \\ y_{GrundPos} \end{pmatrix}}_{\vec{x}_{GrundPos}} \quad (4.42)$$

Diese Berechnung ist wiederum analog auf drei Dimensionen anwendbar, um die geforderten sechs Freiheitsgrade zu erhalten.

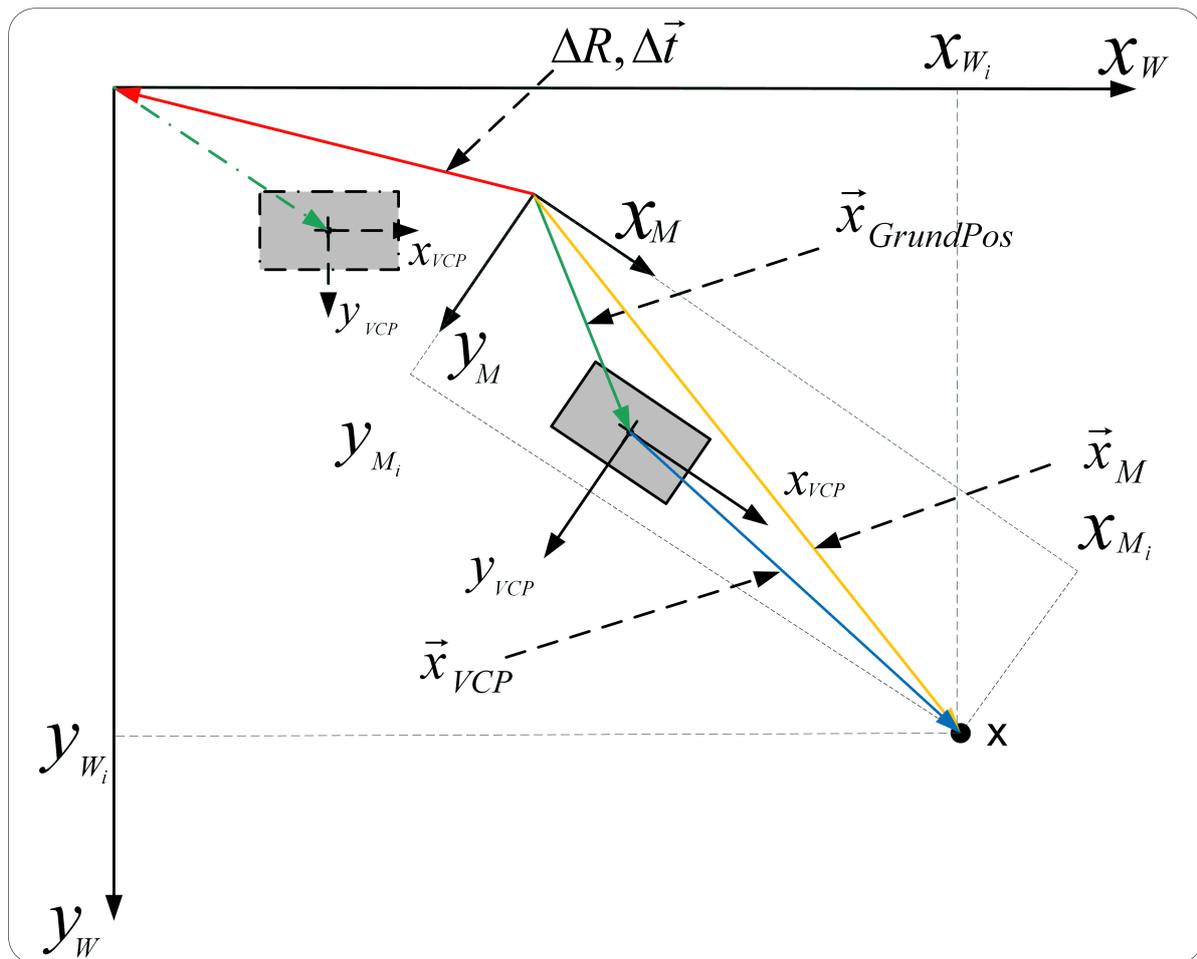


Abb. 4.8: Transformation der Koordinaten eines Weltpunktes in VCP-Koordinaten

4.3.4 Berechnung der Transformationsparameter und der resultierenden VCP-Pose

Zur mathematischen Bestimmung der Transformationsparameter werden die Gleichungen 4.33 und 4.35 aus Unterabschnitt 4.3.1 verwendet, um ein lineares Gleichungssystem aufzustellen. Zunächst wird $\Delta \mathbf{R}$ mit Gl. 4.33 berechnet und in die Gl. 4.35 eingesetzt. Es resultiert ein lineares Gleichungssystem, bestehend aus drei Gleichungen mit drei Unbekannten zur Errechnung des Translationsvektors $\Delta \vec{t}$ 4.43. Die Orientierung der Plattform bezüglich des Welt-KS wird durch den Rotationsparameter $\Delta \mathbf{R}$ in Form einer Rotationsmatrix angegeben, wobei die Winkel α, β und γ nicht direkt aus der Rotationsmatrix ablesbar sind. Für die Darstellung der einzelnen Rotationswinkel um die Koordinatenachsen sind weitere Schritte notwendig, die nachfolgend erläutert werden.

$$\underbrace{\begin{pmatrix} \Delta t_x \\ \Delta t_y \\ \Delta t_z \end{pmatrix}}_{\Delta \vec{t}} = \underbrace{\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}}_{\vec{t}} - \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}}_{\Delta \mathbf{R}^T \cdot \vec{t}^*} \cdot \underbrace{\begin{pmatrix} t_x^* \\ t_y^* \\ t_z^* \end{pmatrix}}_{\vec{t}^*} \quad (4.43)$$

Zur Lösung des Gleichungssystems werden die vorhandenen Parameter \mathbf{R} , \mathbf{R}^* , \vec{t} und \vec{t}^* benötigt. Nach einsetzen dieser Parameter in das Gleichungssystem erfolgt die Lösung durch die Anwendung der Matrizenrechnung. Zur programmiertechnischen Lösung dieser Matrizenrechnung wird wiederum die LTI-Lib⁴ verwendet.

Die Errechnung der Orientierungswinkel zur Angabe der Ausrichtung der Plattform erfolgt unter Verwendung der errechneten Rotationsmatrix $\Delta\mathbf{R}$ durch Gl. 4.5 und der Gl. aus 4.44.

$$\beta = \arcsin(r_{31}), \quad \alpha = -\arctan\left(\frac{r_{32}}{r_{33}}\right), \quad \gamma = -\arctan\left(\frac{r_{21}}{r_{11}}\right), \quad (4.44)$$

Die Arcusfunktionen sind jedoch mehrdeutig wodurch acht mögliche Lösungstriplets $(\alpha_k^*, \beta_l^*, \gamma_m^*)$ mit $k = 1, 2$, $l = 1, 2$ und $m = 1, 2$ entstehen. Dabei weist das richtige Lösungstriplet den geringsten Fehler auf (vgl. [Meisel (1994)] S.48).

Da es sich bei der Rotationsmatrix um ein Ergebnis aus der Ausgleichsrechnung handelt, kann keine exakte orthogonale Matrix erwartet werden. Dementsprechend ungenau werden die Ergebnisse. Eine weitere Ausgleichung mit den Winkeln $\Delta\alpha$, $\Delta\beta$ und $\Delta\gamma$ würde zu genaueren Ergebnissen führen. Für diese Arbeit genügt jedoch die Näherung der Werte mit dem kleinsten Fehler, wie nachfolgend in Gl. 4.45 beschrieben.

$$e^2 = \sum_{i=1}^3 \sum_{j=1}^3 (r_{ij}^* - r_{ij}(\alpha_k^*, \beta_l^*, \gamma_m^*))^2 \quad (4.45)$$

4.4 Raummodell zur Identifikation und Zuordnung von Passmarkenkoordinatenpaaren

Das Raummodell der kamerabasierten Präzisionsnavigation dient der Verarbeitung von identifizierten Passmarken, um eine Zuordnung der gemessenen 2D-Bildkoordinaten zu den korrespondierenden 3D-Weltkoordinaten der Passmarken vornehmen zu können. Das in diesem Zusammenhang genutzte *Tracker*-Modul ist ein eigenständiges, konfigurierbares System, das unter Verwendung der Bildverarbeitung kodierte Passmarken in Bildern erkennt und identifiziert. Es handelt sich dabei um ein bereits vorhandenes Modul, welches im Rahmen einer Bachelorarbeit an der HAW erstellt wurde (vgl. [Wilms (2008)]). Dieses Modul wurde lediglich um die Schnittstelle für die Ausgabe der erkannten Passmarkendaten erweitert. Das *Raummodell* an sich enthält die 3D-Weltkoordinaten der genutzten Marken und nimmt die Zuordnung der 2D- zu den 3D-Koordinaten vor.

Um den *Tracker* zur Identifikation der Passmarken verwenden zu können, müssen spezielle Passmarken angefertigt werden. Da zur Posenbestimmung der Kamera mindestens drei Marken im Sichtbereich der Kamera ständig erkannt werden müssen ist je nach Fahrbereich der

⁴LTI-Lib - ist eine Open Source Software, die seit 1998 am Lehrstuhl für Technischen Informatik an der Rheinisch Westfälischen Technischen Hochschule Aachen (RWTH) entwickelt wurde. Sie stellt unter anderem Bildverarbeitungsalgorithmen sowie Funktionen der linearen Algebra, Vektor- und Matrizenrechnung und Visualisierungswerkzeuge bereit.

Plattform und Ausrichtung der Kamera die Anzahl der Marken zu bestimmen und anzufertigen. Die Passmarken sowie eine Software zur Bereitstellung der Passmarkengeometriedaten wurden bereits in einer Bachelorarbeit an der HAW entwickelt (vgl. [Wippich (2008)]) und in einer weiteren Arbeit weiterentwickelt (vgl. [Wilms (2008)]). Die Passmarken bestehen aus einem Rechteck, in dem fünf Punkte vertikal in einer Linie angeordnet sind. Um einen hohen Kontrast zur Erkennung der Marken zu erzeugen, ist die Hintergrundpappe schwarz. Der Rahmen sowie die Punkte bestehen aus einer retroreflektierenden Folie. Durch die unterschiedliche Anordnung der Kreise im Rahmen werden Doppelverhältnisse gebildet, so dass die Marken voneinander unterscheidbar sind. Der Mittelpunkt des mittleren Kreises wird als Markenposition bezeichnet, die durch die 2D-Bildkoordinaten und die 3D-Weltkoordinaten beschrieben wird. Die Anbringung der Marken kann wahlweise vertikal oder horizontal erfolgen, wobei die Erkennung der Marken bis zu einer Kippung von 45° durchführbar ist. Ebenso ist eine Veränderung der Perspektive durch Bewegung der Kamera in einem bestimmten Rahmen erlaubt. Durch die spezielle Geometrie und Beschaffenheit sind die Marken auch unter wechselnden Lichtverhältnissen und Betrachtungswinkeln sehr präzise identifizierbar. In Abbildung 4.9 sind einige der im Rahmen dieser Arbeit angefertigten und genutzten Marken abgebildet. Durch die Anbringung an der Decke und Drehung der Marken um 90° zueinander ist in diesem Fall sowohl eine horizontale als auch vertikale Erkennung möglich. Die Plattform ist folglich in der Lage, komplette Drehungen um die eigene Achse zu vollführen, wenn die Kamera senkrecht zur Decke orientiert auf der Plattform installiert wird.



Abb. 4.9: Verwendete Passmarken vertikal und horizontal an der Decke angebracht

Die 3D-Weltkoordinaten sämtlicher Passmarken werden in dem *Raummodell* zusammen mit jeweils einer eindeutigen ID gespeichert, wobei die ID ebenfalls im *Tracker* der jeweiligen Marke zugeordnet wird. Nach jeder Suche nach Passmarken im aufgenommenen Bild übergibt der *Tracker* dem *Raummodell* die ID der erkannten Marken mit den dazugehörigen 2D-Bildkoordinaten. Das *Raummodell* nimmt aufgrund der ID die Zuordnung der 2D-Bildkoordinaten zu den 3D-Weltkoordinaten vor. Als Ergebnis werden die Koordinatenpaare zusammen zur weiteren Verwendung der *Posenbestimmung* bereitgestellt.

4.5 Vereinfachte Bahnplanung für die Navigation ohne Hindernisse

In diesem Abschnitt erfolgt die Beschreibung des Konzeptes für eine stark vereinfachte Bahnplanung und -verfolgung einer mobilen Plattform. Die Planung und Verfolgung von Bahnen befindet sich nicht im Fokus der Arbeit, sie wird aber benötigt, um eine erste Anwendung der gewonnenen Ergebnisse und somit das Abfahren einer virtuellen Spur zu demonstrieren. Aus diesem Grund wurde nur sehr geringer Aufwand in die Recherche investiert und ein stark vereinfachtes Modell zur direkten Ansteuerung von Positionen gewählt. In dem Modell werden keine Hindernisse berücksichtigt, so dass von einem leeren Raum ausgegangen werden kann. Zudem wird von der Bewegung auf einer Ebene ausgegangen, es erfolgt lediglich die Betrachtung der 2D-Koordinaten für die Planung. Das Konzept zur Bahnplanung entspricht einem an der Universität Bremen entwickelten Modell zur direkten, geraden Bahnplanung vom Start zum Zielpunkt in der ausschließlichen Vorwärtsbewegung (vgl. [Röfer (2001)]). Diese Modell genügt, um zum Beispiel einen Rundkurs abzufahren, der durch eine bestimmten Reihenfolge nacheinander abzufahrender Einzelziele definiert ist.

Die Abbildung 4.10 zeigt einen typischen Anwendungsfall für die Bahnplanung. Für die direkte Ansteuerung der Zielposition muss die Plattform zunächst aus der aktuellen Orientierung (γ_{Start}) in die Richtung des Ziels (γ_{Basis}) ausgerichtet werden. Es folgt die direkte Geradeausfahrt bis zur Zielposition und anschließend die erneute Drehung der Plattform in die Zielausrichtung (γ_{Ziel}). Für die Berechnung des Basiswinkels γ_{Basis} , des Startwinkels γ_{Start} und des Zielwinkels γ_{Ziel} werden die folgenden Gleichungen genutzt.

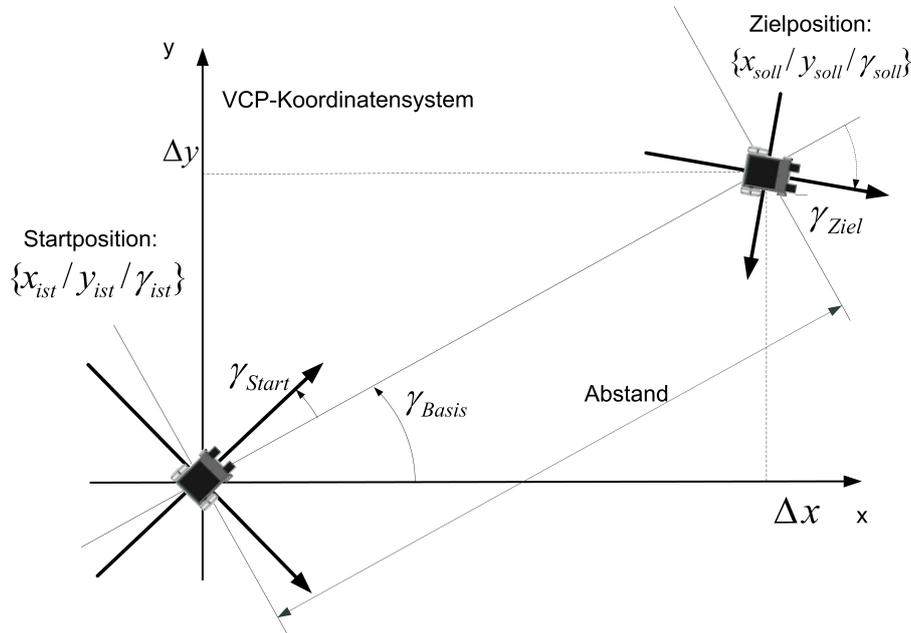


Abb. 4.10: Direkte Bahnplanung - Start und Ziel

$$\gamma_{Basis} = \text{atan2}(\Delta y, \Delta x) \quad \gamma_{Start} = \gamma_{ist} - \gamma_{Basis} \quad \gamma_{Ziel} = \gamma_{soll} - \gamma_{Basis} \quad (4.46)$$

Die Berechnung der Entfernung zum Ziel entfällt, weil die aktuelle Position ständig ermittelt wird und daraufhin die Bahnkorrektur erfolgt. Somit ist die Entfernung zum Ziel unwichtig für diese Anwendung.

Quellcode 4.1: Pseudocode für die Bahnverfolgung mit integrierter Bahnplanung

```
1 Procedure ErzeugeKommando(in VCP-Pose, in  $\Delta\vec{I}$ , in  $\Delta R$ , in VCP-  
   Startposition, out Kommando)  
2   Berechne Ziel-Pose in VCP-Koordinaten  
3   if ist Ziel-Pose gültig then  
4     if Ziel erreicht then  
5       if Plattform in Zielausrichtung then Kommando = STOP  
6         if weiteres Ziel in Liste then  
7           wähle nächstes Ziel  
8         end if  
9       else Berechne Rotation bis Zielausrichtung  
10        if Rotation > 0 then Kommando = DREHE RECHTS  
11        else if Rotation < 0 then Kommando = DREHE LINKS  
12        end if  
13      endif  
14    else Berechne Rotation bis Zielrichtung  
15      if Rotation > 0 then Kommando = DREHE RECHTS  
16      else if Rotation < 0 then Kommando = DREHE LINKS  
17      else Kommando = FAHRE VORWÄRTS  
18      end if  
19    end if  
20  else  
21    Fehler loggen  
22  end if  
23 end ErzeugeKommando
```

5 Implementation des gesamten Konzeptes

In diesem Kapitel werden die grundlegenden und hervorzuhebenden Anteile der Implementation, der in dieser Arbeit entwickelten Navigation für mobile Systeme, erläutert. Neben den Komponenten, deren Zusammenspiel sowie deren Verbindung werden die Schnittstellen der entwickelten Bibliotheken beschrieben. Auf eine detaillierte Beschreibung der einzelnen Klassen wird an dieser Stelle verzichtet. Die Schwerpunkte dieses Kapitels setzen sich wie folgt zusammen:

- Überblick der Komponenten,
- Beschreibung der beteiligten Komponenten und deren Schnittstellen sowie
- Verknüpfung der Module mit Hilfe von Impresario.

In der Implementation wird Drittanbieter-Software verwendet, um bereits bestehende Software für Standardlösungen einsetzen zu können. So wird zum Beispiel für das Logging *log4cxx* (vgl. [log4cxx (2008)]) und als XML-Parser zum Einlesen des Raummodells und der Zielliste der *Xerces XML-Parser* von Apache verwendet (vgl. [XercesXML (2008)]). Als mobile Plattform wird der einachsige Roboter *PIONEER P3-DX* eingesetzt, dessen Steuerung über das Application Programming Interface (API) *Advanced Robotics Interface for Applications (ARIA)* der Firma Mobile Robots integrierbar ist (vgl. [MobileRobots (2008)]). Des Weiteren wird in großen Teilen der entwickelten Software die *LTI-Lib* zur Berechnung von Vektor- und Matrixoperationen sowie für das Lösen von Gleichungssystemen verwendet (vgl. [LtiLib (2005)]). Der aus einer Bachelorarbeit entstandene *Tracker* (vgl. [Wilms (2008)]) wird ebenfalls wie eine Drittanbieter-Software eingebunden. Die weiteren Module wurden im Rahmen dieser Arbeit entwickelt beziehungsweise angepasst und erweitert.

Abbildung 5.1 zeigt eine Gesamt-Modulübersicht, die den Aufbau der entwickelten Software und die Abhängigkeiten zwischen den Modulen widerspiegelt. Drittanbieter-Software-Komponenten werden in dieser Darstellung abweichend von den hier entwickelten Komponenten gekennzeichnet, um den Eigenanteil von der bestehenden Software abzuheben. Das Logging-Modul wird von allen selbst entwickelten Komponenten verwendet, um die Fehlersuche und das Festhalten von Messergebnissen in der grafischen Oberfläche Impresario zu gewährleisten.

Alle in dieser Arbeit verwendeten Komponenten wurden in C++ unter der Verwendung des *Visual Studio .Net 2003* implementiert. Die Drittanbieter-Software-Komponenten liegen in Form von Quellcode vor und mussten für die Verwendung zunächst kompiliert werden. Die Bindung an die relativ alte Version des *Visual Studio .Net* entsteht durch die Komponenten *LTI-Lib* und *Impresario*, welche nur zusammen mit dieser Visual Studio Version zuverlässig arbeiten.

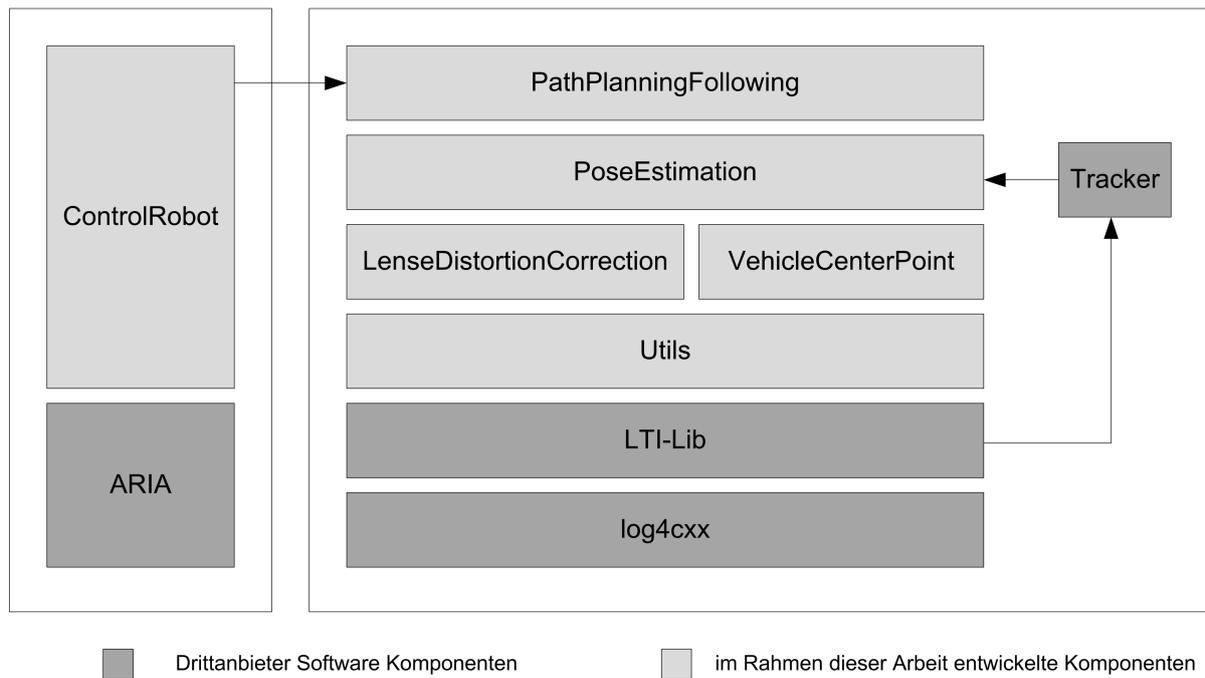


Abb. 5.1: Gesamtmodulübersicht der entwickelten Lösung

Im Anschluss in Quellcodeausschnitt 5.1 folgt Pseudocode zur Verdeutlichung des Kontrollflusses und der Funktion der Software. Aus dem Pseudocode wird der gesamte Ablauf von der initialen Posenbestimmung bis hin zur Ansteuerung der mobilen Plattform erkennbar.

Quellcode 5.1: Pseudocode des Kontrollflusses der gesamten Navigationssoftware

```
1 Procedure NavigiereZuZiel (in KameraKonfDatei, in
   RaummodellDatei, in ZiellisteDatei)
2   Einlesen Kameraparameter, Schätzpose und VCP-Position aus
   KameraKonfDatei
3   Einlesen der Passmarken 3D-Koordinaten aus RaummodellDatei
4   Systeminitialisierung für erste Posenbestimmung
5   Hole 2D-Koordinaten und IDs der identifizierten Marken von
   Tracker
6   Verzeichnungskorrektur
7   Initiale Posenbestimmung in VCP-Grundposition
8   Initialisierung VCP
9   Einlesen Zielliste aus ZiellisteDatei
10  while Ziel in Liste
11    if mindestens 3 Marken erkannt then
12      Verzeichnungskorrektur
13      Posenbestimmung der Kamera
14      // Kollinearität ausschließen
15      if Pose OK then
16        Posenbestimmung des VCP
17        Ausgabe der VCP-Pose
18        Pfadplanung
19        if Ziel in Liste then
20          if Ziel erreicht then
21            nächstes Ziel wählen
22          else
23            Ziel ansteuern
24          end if
25        end if
26      else
27        Pose = letzte gültige Pose
28      end if
29    else
30      Ausgabe = nicht genug Marken erkannt
31      Pose = letzte gültige Pose
32    end if
33  end while
34  Fahrzeug anhalten
35 end NavigiereZuZiel
```

5.1 Verwendung der Linsenverzeichnungskorrektur als Modul

Für die Erlangung der Sub-Pixelgenauigkeit in dieser Arbeit ist lediglich die Korrektur einzelner Bildpunkte erforderlich. Die Korrektur kompletter Bilder wurde im Projekt bereits prototypisch implementiert. Um die Verzeichnungskorrektur als Modul dieser Arbeit nutzen zu können, war die Erstellung einer Bibliothek notwendig, die Funktionen zur Umwandlung von verzeichneten Bildpunkten in verzeichnungsfreie Bildpunkte bereitstellt. Teile der bisherigen Implementation der Linsenverzeichnungskorrektur wurden wiederverwendet, jedoch ist in diesem Fall keine geometrische Transformation (Rückwärtstransformation) erforderlich, so dass die Gleichungen 4.17 und 4.18 aus Abschnitt 4.2 direkt angewendet werden konnten. Der unten aufgeführte Quellcodeausschnitt 5.2 setzt die beschriebene Vorgehensweise der Implementation um. Es wird ersichtlich, dass die Implementation der Rückwärtstransformation, abgebildet im Quellcodeausschnitt C.1 in Anhang C, wesentlich komplexer als die Vorwärtstransformation ist.

Quellcode 5.2: Korrektur der verzeichneten Bildpunkte zum Einsatz als Modul

```
1 void LenseDistortionCorrection::correctPix(Point * point){
2     double uc, vc;          // corrected Chip-Coordinate
3     double ud, vd;          // distorted Chip-Coordinate
4     // convert into metric system
5     vd= this->convertMetric(point->getV(), false);
6     ud= this->convertMetric(point->getU(), true);
7     // correction of distortion
8     uc = f1(ud,vd);
9     vc = f2(ud,vd);
10    // convert back to pixel system and store in point
11    point->setU(convertPix(uc, true));
12    point->setV(convertPix(vc, false));
13
14 }
```

5.2 Kamerabasierte Posenbestimmung der mobilen Plattform

Das Posenbestimmungs-Modul wurde als Software-Bibliothek zur Einbindung in andere Module Form einer *Dynamic Link Library (DLL)* entwickelt. Dieses Modul nutzt einige spezielle Datentypen der *LTI-Lib* sowie Algorithmen zur Lösung von Gleichungssystemen durch die Anwendung der Vektor- und Matrizenrechnung. Zudem findet eine Anwendung des *Tracker* Moduls zur Identifikation der Passmarken und die Bereitstellung der 2D-Koordinaten der jeweiligen Marken statt. Neben den genannten Drittanbieter-Software-Komponenten werden die Module *Utils* und *VehicleCenterPoint* für die Posenbestimmung genutzt. Das Modul *Utils* stellt Datentypen und Rechenoperationen bereit, die zusätzlich von anderen Modulen genutzt werden können, was eine Kapselung sinnvoll macht. Des Weiteren bietet das Modul *VehicleCenter-*

Point Funktionen zur Berechnung des VCP nach dem Konzept aus Abschnitt 4.3 an. Durch den modularen Aufbau der einzelnen Module wird die Weiterverwendung erheblich vereinfacht.

Die für die Posenbestimmung benötigten internen Kameraparameter werden beim Start der Posenbestimmung aus einer Text-Datei (`camera.ini`) gelesen, zudem enthält diese Datei die geschätzte Start-Pose der Kamera und die 3D-Koordinate der VCP-Grundposition. Das Erfassen der 3D-Koordinaten aller verwendeten Passmarken erfolgt dagegen durch das Parsen einer XML-Datei (`roomModel.xml`). Diese XML-Datei enthält die ID und die dazugehörigen 3D-Koordinaten (x,y,z) der jeweiligen Marke. Ein Auszug aus der Datei zur Veranschaulichung der Schemas ist in Quellcodeauschnitt 5.3 abgebildet. Die eingelesenen Daten werden für die weitere Verwendung in geeigneten Variablen gespeichert. Die Dateinamen der angegebenen Dateien sind durch die Angabe bei der Initialisierung der Posenbestimmung konfigurierbar.

Quellcode 5.3: Auszug aus der XML-Datei zur Angabe der Markenkoordinaten

```

1<RoomModel>
2<marks>
3  <!-- the coordinates will be declared in m -->
4  <mark   id="3"
5         x="1.7040"
6         y="0.0"
7         z="1.3350"/>
8  ...
9</marks>
10</RoomModel>

```

Nach der Initialisierung des Systems kann die Posenbestimmung durchgeführt werden. Es sind zwei Varianten zur Durchführung verfügbar. Im ersten Fall können die 3D-Koordinaten zusammen mit den korrespondierenden 2D-Koordinaten der erkannten Marken direkt unter Angabe der Anzahl der Marken angegeben werden. Die zweite Variante erlaubt die Angabe einer Liste der identifizierten Marken, inklusive der 2D-Koordinaten. In jenem Fall findet eine Zuordnung der 2D- zu den 3D-Koordinaten anhand der Marken ID durch das Raummodell statt. Die korrespondierenden Koordinatenpaare werden zur Anwendung der zuvor beschriebenen Funktion in einem Datentyp zusammengefasst und unter Angabe der Anzahl der Marken übergeben.

Bevor die Pose der Kamera berechnet werden kann, erfolgt implizit die Korrektur der Linsenverzeichnung. Zur Berechnung der Pose wird die nicht lineare Ausgleichung zur Lösung des aufgestellten Gleichungssystems eingesetzt. Das Gleichungssystem wird dynamisch nach der Anzahl der erkannten Passmarken aufgebaut, wobei als Vorbedingung mindestens drei Marken verfügbar sein müssen. Die iterative Lösung des Gleichungssystems erfolgt wahlweise durch die Anwendung der QR-Zerlegung mit der Householder-Transformation oder durch die singuläre Wertezerlegung (SVD). Beide Varianten werden von der *LTI-Lib* unterstützt. Als Ergebnis der Berechnung erfolgt die Rückgabe der ermittelten Kamera-Pose.

Funktionen zur Nachkalibrierung der Kamera-Pose zur Berechnung der Plattformkoordinaten im Welt-KS stellt das Modul *VehicleCenterPoint* bereit. In diesem Modul findet die Umsetzung der Berechnungen der Transformationsparameter $\Delta \mathbf{R}$ und $\Delta \vec{t}$ statt. Wiederum wird die LTI-Lib zur Lösung der Vektor- und Matrizenrechnung eingesetzt. Daraufhin wird die Berechnung der VCP Koordinaten im Welt-KS vorgenommen, gefolgt von der Berechnung der Orientierungswinkel (α, β, γ) aus der Rotationsmatrix. Die resultierende Plattform-Pose wird als Ergebnis der Nachkalibrierung zurückgegeben. Die Funktionen zur Abbildung von Weltkoordinaten im

Tabelle 5.1: Schnittstellenübersicht der Posenbestimmung

Funktion	Parameter
<code>void initPoseEstimation</code>	<code>const char * ini_file,</code> <code>const char * a_roomFile</code>
<code>Pose* estimateCameraPose</code>	<code>Point** a_points,</code> <code>int a_numberOfPoints</code>
<code>Pose* estimateCameraPose</code>	<code>const list<Mark>* a_markList</code>
<code>Pose* getCameraPose</code>	-
<code>void initPlatformPose</code>	<code>Pose* a_pose</code>
<code>Pose* estimatePlatformPose</code>	<code>Point** a_points,</code> <code>int a_numberOfPoints</code>
<code>Pose* estimatePlatformPose</code>	<code>const list<Mark>* a_markList</code>
<code>Pose* getPlatformPose</code>	-

VCP werden ebenso wie die Funktion zur Berechnung der Orientierungswinkel statisch zur Verfügung gestellt. Dadurch können diese beiden Funktionen später auch ohne Instantiierung aus *Impresario* heraus genutzt werden.

5.2.1 Schnittstellen der Posenbestimmung

Die öffentlichen Schnittstellen der Posenbestimmung werden in der Tabelle 5.1 dargestellt. Die Tabelle enthält den Funktionsnamen und die Parameterliste jeder Schnittstelle, die zur Verwendung der Posenbestimmung bereitgestellt werden. Die Beschreibung der Schnittstellen erfolgt im Anschluss. Mit Offenlegung der beschriebenen Schnittstellen steht das Posenbestimmungs-Modul zur weiteren Verwendung auch außerhalb dieser Arbeit zur Verfügung.

- `void initPoseEstimation(const char * ini_file, const char * a_roomFile)`
Diese Funktion wird unter Angabe von zwei Dateinamen zur Initialisierung der Posenbestimmung aufgerufen. Die erste Datei enthält die internen Kameraparameter, die Start-Pose der Kamera und die Grundposition des VCP. Die zweite Datei beinhaltet die 3D-Koordinaten der angebrachten Passmarken sowie deren Identifikationsnummern. Vor der Initialisierung ist keine Posenbestimmung möglich.
- `Pose* estimateCameraPose(Point** a_points, int a_numberOfPoints)`
Unter Angabe der 2D-3D-Koordinatenpaare und der Anzahl dieser Paare wird die aktuelle Pose der Kamera bestimmt. Die errechnete Pose wird als Ergebnis zurückgegeben.
- `Pose* estimateCameraPose(const list<Mark>* a_markList)`
Als Parameter wird die Liste mit den erkannten Markennummern und den dazugehörigen 2D-Koordinaten übergeben. Auf Grundlage der Liste erfolgt die Ermittlung der korrespondierenden 3D-Koordinaten und die Anzahl der Marken. Die gewonnenen Daten werden an die zuvor beschriebene Funktion übergeben. Das Ergebnis wird ebenfalls in Form der Kamera-Pose zurückgegeben.

- `Pose* getCameraPose()`
Gibt die Kamera-Pose zurück.
- `void initPlatformPose(Pose* a_pose)`
Soll zusätzlich zur Kamera-Pose die Pose einer Plattform ermittelt werden, muss zuvor die Initialisierung unter Angabe der Kamera-Pose zum Zeitpunkt der initialen Posenbestimmung vorgenommen werden.
- `Pose* estimatePlatformPose(Point** a_points, int a_numberOfPoints)`
Die Plattform-Pose wird ebenso wie die Kamera-Pose unter Angabe der Koordinatenpaare und deren Anzahl berechnet. Implizit wird die Kamera-Pose und daraus dann die VCP-Pose ermittelt. Der Rückgabewert ist die Plattform-Pose.
- `Pose* estimatePlatformPose(const list<Mark>* a_markList)`
Durch Übergabe der Markenliste kann auch die Plattform-Pose analog zur Kamera-Pose berechnet werden. Auch hier erfolgt intern der Aufruf der zuvor beschriebenen Funktion. Das Ergebnis ist erneut die Plattform-Pose.
- `Pose* getPlatformPose()`
Gibt die Plattform-Pose zurück.

Zur Nutzung des Moduls in `Impresario` wird das Modul `Navigator` bereitgestellt. Es benötigt den Eingabeparameter

- `std::list<utils::Mark> m_markList` - Die Liste aller identifizierten Marken im Kamerabild.

und liefert die folgenden Ausgabeparameter.

- `utils::Pose` - Die Pose der Plattform in Bezug zur Welt.
- `lti::matrix<double>` - Die Rotationsmatrix, der Rotation zwischen der Welt und dem mitbewegten KS.
- `lti::vector<double>` - Der Translationsvektor zwischen der Welt und dem mitbewegten KS.
- `lti::vector<double>` - Die Grundposition des VCP zum Zeitpunkt der initialen Posenbestimmung.

5.2.2 Passmarkenidentifizierer zur Erkennung der Passmarken während der Laufzeit

Das *Tracking* Modul wird in dieser Arbeit als Drittanbieter-Software-Komponente zur Identifikation der Passmarken und zur Bereitstellung der Passmarkenkoordinaten in der Bildebene eingebunden. Die Funktionsweise dieses Moduls wurde in Abschnitt 4.4 beschrieben. Diese Komponente ist bereits vorhanden und für die Nutzung in der grafischen Oberfläche `Impresario` konzipiert. Für die Verknüpfung dieses Moduls mit der Posenbestimmung zur Verwendung in

Impresario genügt es somit, eine Schnittstelle für die Bereitstellung der erkannten Markendaten zu implementieren. In Impresario werden die Schnittstellen zwischen den Modulen nicht wie üblich als Funktionen, sondern als Datentypen definiert. Über diese Datentypen erfolgt schließlich die Verknüpfung. Folgende Schnittstelle wurde zur Verknüpfung der Module vereinbart und sowohl in dieser Arbeit als auch in der parallel durchgeführten Bachelorarbeit zur Lokalisierung der Marken implementiert (vgl. [Wilms (2008)]).

- `std::list<utils::Mark> m_markList`
Dieser Datentyp ist eine Liste, die nur Elemente vom Typ `utils::Mark` enthält. Der Datentyp `utils::Mark` setzt die Nutzung des *Utils* Moduls aus dieser Arbeit voraus, welche für die allgemeine Nutzung auch außerhalb dieser Arbeit entwickelt wurde.

Durch die Verwendung dieses Moduls wird nicht nur die komplette Markenidentifikation geliefert, sondern zudem die Anbindung der auf der Plattform installierten Kamera übernommen. Dieses Modul leistet damit einen großen Beitrag zur Umsetzung der kamerabasierten Navigation in dieser Arbeit.

5.3 Planung der Plattformbewegung zur Ansteuerung von Zielen

Das Modul zur Bahnplanung und -verfolgung (*PathPlanningFollowing*) implementiert die in Abschnitt 4.5 beschriebene Funktionalität. Es wird lediglich eine stark vereinfachte Bahnplanung und -verfolgung bereitgestellt. Das Modul wurde konzipiert, um bei der Angabe der aktuellen VCP-Pose das jeweils nächste Kommando für die mobile Plattform zu generieren, welches zur Erreichung des aktuellen Ziels notwendig ist. Dieses Vorgehen wird für jede neue verfügbare VCP-Pose vorgenommen, so dass zu jeder neuen Pose ein Kontrollkommando generiert wird. Die Bahnverfolgung besteht somit aus der ständigen Korrektur der Bahn, indem das jeweils generierte Kommando auf der Plattform ausgeführt wird. Aus diesem Grund ist in dieser vereinfachten Form der Bahnverfolgung keine Regelung, wie sie in den meisten Bahnverfolgungsansätzen üblich ist, notwendig. Die Umsetzung erfolgt nach dem Pseudocode aus Quellcodeausschnitt 4.1. Eine Erläuterung der Kontrollkommandos sowie der verwendeten API *ARIA* erfolgt später in diesem Abschnitt.

Die anzusteuern Ziele werden aus einer XML-Datei eingelesen und in einer Liste gespeichert. Die Liste enthält ausschließlich Elemente vom Typ `utils::Pose`, so dass die Ziele samt ihrer Orientierung gespeichert werden. Das Einlesen erfolgt ähnlich wie beim Raummodell mit Hilfe des *Xerces XML-Parser* (vgl. [XercesXML (2008)]). Durch das sequentielle Abarbeiten der Liste wird eine virtuelle Route abgefahren, in der die Anzahl der Zielpunkte die Granularität der Route vorgibt.

An dieser Stelle wird noch einmal ausdrücklich auf die Navigation als Fokus dieser Arbeit hingewiesen. Das hier beschriebene Bahnplanungs- und -verfolgungs-Modul soll ausschließlich als eine erste Anwendung der entwickelten Posenbestimmung zur Vorführung der Ergebnisse genutzt werden.

Tabelle 5.2: Schnittstellenübersicht der Bahnplanung und -verfolgung

Funktion	Parameter
<code>void getControlCommand</code>	<code>const Pose* a_currentVcpPose,</code> <code>ControlCommand& a_command</code>
<code>void getControlCommand</code>	<code>const Pose* a_currentVcpPose,</code> <code>const lti::vector<double>* a_delta_t,</code> <code>const lti::matrix<double>* a_delta_R,</code> <code>const lti::vector<double>* a_vcpStartPosition,</code> <code>ControlCommand& a_command</code>

Die Schnittstelle ist allgemein spezifiziert, um den Austausch dieses Moduls gegen eine leistungsfähigere Lösung zu ermöglichen. Eine Übersicht der Schnittstellen und deren Parameter wird in Tabelle 5.2, gefolgt von einer Beschreibung vorgenommen.

- `getControlCommand(const Pose* a_currentVcpPose,`
`ControlCommand& a_command)`

Diese Schnittstelle wurde in dieser Arbeit nicht implementiert. Sie wurde geschaffen, um den Austausch dieses Moduls gegen ein Leistungsfähigeres zu erleichtern.

- `getControlCommand(const Pose* a_currentVcpPose,`
`const lti::vector<double>* a_delta_t,`
`const lti::matrix<double>* a_delta_R,`
`const lti::vector<double>* a_vcpStartPosition,`
`ControlCommand& a_command)`

Diese Funktion generiert ein Kontrollkommando für die mobile Plattform. Das Kommando wird berechnet, indem durch die angegebenen Parameter die Position und Orientierung des Ziels in Bezug zur Plattform (VCP) errechnet wird. Daraufhin wird das nächste Kommando generiert und in dem Parameter `ControlCommand a_command` gespeichert.

Zum Austausch des Moduls müsste das neue Modul lediglich die gleichen externen Schnittstellen implementieren und mit dem gleichen Namen versehen werden. Die erzeugte DLL könnte wie zuvor genutzt werden.

Zur Nutzung des Moduls in *Impresario* wird das Modul *PathPlanningFollowing* bereitgestellt. Es benötigt die Eingabeparameter

- `utils::Pose` - Die Pose der Plattform in Bezug zur Welt.
- `lti::matrix<double>` - Die Rotationsmatrix, der Rotation zwischen der Welt und dem mitbewegten KS.
- `lti::vector<double>` - Der Translationsvektor zwischen der Welt und dem mitbewegten KS.
- `lti::vector<double>` - Die Grundposition des VCP zum Zeitpunkt der initialen Posenbestimmung.

und liefert den folgenden Ausgabeparameter.

- `ControlCommand` - Kontrollkommando für die Plattform.

Steuerung der Plattform mit ARIA

Die Firma Mobile Robots stellt für ihre mobile Plattform *PIONEER P3-DX* ARIA als API zur Verfügung. Mit Hilfe dieser API können sämtliche Funktionen des Roboters aus der eigenen Software aufgerufen werden. Für die Anwendung in diesem Kontext werden lediglich die Funktionen zur direkten Bewegung der Plattform benötigt (*Direct Motion Commands*). Die Nutzung dieser API setzt eine Initialisierung der Roboterinstanz voraus. Jedoch bestehen die Direktkommandos nur aus Befehlen wie *fahren* oder *drehen* mit der Angabe der Geschwindigkeit der jeweiligen Bewegung. Zur Vereinfachung der Anwendung der API wurde ARIA in dieser Arbeit gekapselt und als Modul *ControlRobot* bereitgestellt. Mit diesem Modul wurde die Initialisierung auf nur einen Aufruf reduziert, zusätzlich wurden die Steuerungskommandos zum Zweck der einfachen Navigation angepasst. Die Übergabe des Kommandos erfolgt durch den für diesen Zweck entworfenen Kommando-Typ `ControlCommand`. Dieser Datentyp enthält sowohl die Richtungsangabe als auch die Geschwindigkeit für die Bewegung.

In der Tabelle 5.3 sind die externen Schnittstellenfunktionen der Plattformsteuerung aufgelistet. Die Plattform kann direkt über die Richtungsangaben und die Geschwindigkeit der jeweiligen Bewegung, oder durch die Angabe eines Kontrollkommandos mit der Funktion `executeCommand(ControlCommand a_command)` gesteuert werden.

Tabelle 5.3: Schnittstellenübersicht der Plattformsteuerung

Funktion	Parameter
<code>void executeCommand</code>	<code>const ControlCommand* m_command</code>
<code>void rotateLeft</code>	<code>double a_velocity</code>
<code>void rotateRight</code>	<code>double a_velocity</code>
<code>void rotateRightAtPoint</code>	<code>double a_velocity</code>
<code>void rotateLeftAtPoint</code>	<code>double a_velocity</code>
<code>void driveForward</code>	<code>double a_velocity</code>
<code>void driveBackward</code>	<code>double a_velocity</code>
<code>void stop</code>	

Die Bereitstellung des Moduls in Impresario erfolgt durch das Impresario Modul *AriaMoveRobot* mit dem Eingabeparameter vom Typ `ControlRobot::ControlCommand`.

5.4 Verwendung von Impresario zur Verbindung der Module

Im Rahmen dieser Arbeit wird Impresario genutzt, um die entwickelten und die bereits existierenden Module für kamerabasierte Navigation zu verknüpfen. Die benötigten Module werden über DLLs mit speziellem Aufbau in Impresario integriert. Durch die einfache Verbindung der Module ist diese Software gut geeignet, um die entwickelten Module und die gesamte Navigation zu testen. Des Weiteren können die Module bei Bedarf vor und während der Ausführung über die grafische Oberfläche konfiguriert werden.

Die Erstellung der DLLs erfolgt über ein *Visual Studio .Net Projekt*, welches ebenfalls von Impresario-Werkzeugen erstellt werden kann. Innerhalb der DLLs werden die Instanzen der

verwendeten Module erzeugt. Die Nutzung der implementierten Module erfolgt über die zuvor beschriebenen externen Schnittstellen.

Die Ein- und Ausgänge der resultierenden Module werden in der jeweiligen DLL als Ein- und Ausgabeparameter definiert. Hier werden die in den vorherigen Abschnitten vorgestellten Datentypen für die Definition dieser Parameter verwandt. Ein Beispiel für die Definition der Parameter ist in Quellcodeausschnitt 5.4 zu finden.

Quellcode 5.4: Definition der Ein- und Ausgabeparameter in einem Impresario-Modul

```

1 // define input prams
2 AddMacroInput ("Parametername", "Parameterbeschreibung",
   zugewiesene Variable);
3 // define output params
4 AddMacroOutput ("Parametername", "Parameterbeschreibung",
   zugewiesene Variable);

```

Zur Verwendung werden die benötigten Ausgänge mit den passenden Eingängen der Module verbunden. Die Ausführung erfolgt der Verbindung nach sequentiell und beginnt nach der Beendigung des letzten Moduls in der angelegten Kette erneut. Durch die sequentielle Ausführung können keine zeitlichen Probleme beim Zusammenspiel der Module entstehen. Jedoch ergibt sich die Ausführungszeit der gesamten Anwendung aus der Summe aller Einzelanwendungen. Bei Modulen mit hohen Ausführungszeiten wirkt sich das negativ auf die gesamte Ausführungszeit aus. In der hier durchgeführten Anwendung stellt das jedoch kein Problem dar, weil der *Tracker* als langsamstes Modul die essentiellen Markenpositionen liefert. Ohne diese Markenpositionen ist eine erneute Pose ohnehin nicht bestimmbar.

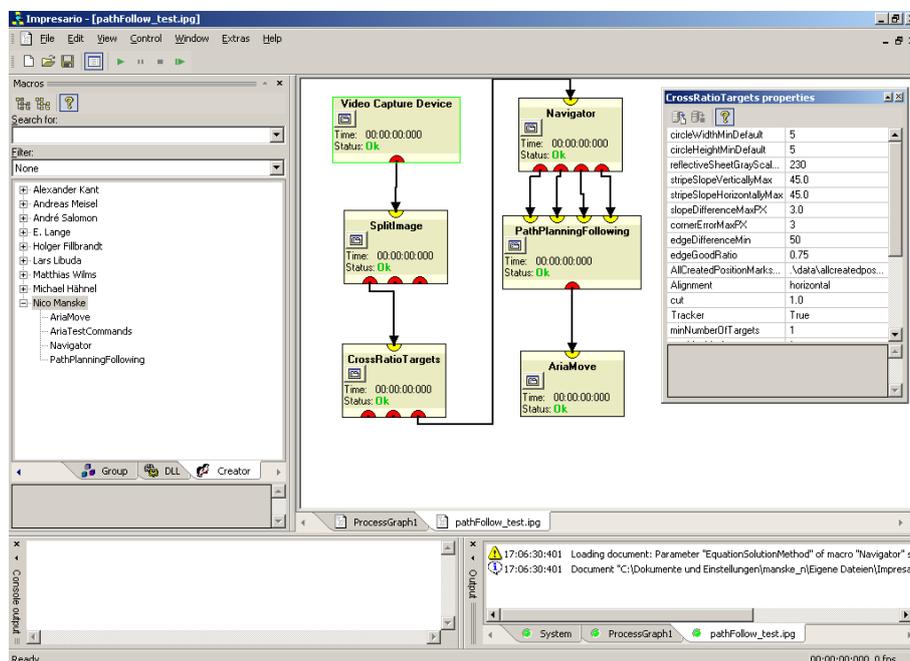


Abb. 5.2: Verknüpfung der Impresario Module

Die Umsetzung der Navigation wäre auch ohne Impresario umsetzbar. Die Verknüpfung der Module kann ebenso über die externen Schnittstellen vorgenommen werden, jedoch sind der *Tracker* und die Kameraanbindung für Impresario ausgelegt. Demzufolge wäre die Nutzung dieser Komponenten mit erheblichem Mehraufwand verbunden.

6 Messtechnische Auswertung der realisierten Lösung

In diesem Kapitel wird die entwickelte Lösung zur kamerabasierten Navigation in Form von Messungen ausgewertet. Des Weiteren wird auf die nicht direkt messbaren Anforderungen für den Einsatz unter realistischen Bedingungen eingegangen. Folgende Punkte werden hier aufgegriffen.

- Beschreibung des Test-Szenarios
- Vermessung der Testumgebung
- Präzision der Navigation
- Ausführungszeiten der Module

Die Genauigkeit der Plattform-Pose ist von der Präzision der Kamera-Pose abhängig, weil die Plattform-Pose aufbauend auf der Pose der Kamera berechnet wird. Dabei wird eine Koordinatentransformation vorgenommen, bei der kein Genauigkeitsverlust stattfindet. Aus diesem Grund entspricht die Präzision der Plattform-Pose der Genauigkeit der Kamera-Pose. Ein weiterer wichtiger Faktor ist das *Tracking*-Modul, welches die 2D-Koordinaten der Passmarken in dem Kamerabild liefert. Die Auswertung der Messdaten zur Präzision erfolgt in Abschnitt 6.2. Der *Tracker* benötigt aufgrund der aufwendigen Bildverarbeitung zur Identifikation der Passmarken zudem die höchste Ausführungszeit. Die Ausführungszeit der restlichen Module ist in Relation dazu zu vernachlässigen. Eine Gegenüberstellung der Messwerte zur Ausführungszeit folgt in Abschnitt 6.3.

Abbildung 6.1 zeigt die in dieser Arbeit genutzte mobile Plattform *PIONEER P3-DX* zur kamerabasierten Indoor-Navigation. Auf der Plattform befindet sich ein Notebook in einer *Docking Station* als zentraler Rechner. Dieser Rechner dient zur Ausführung der entwickelten und vorhandenen Software-Komponenten für die Posenbestimmung der Kamera und die darauf folgende Bestimmung der Plattform-Pose. Zudem steuert der Rechner den Plattformantrieb über eine serielle Anbindung. Des Weiteren ist die verwendete Kamera auf der Abbildung erkennbar, welche per *FireWire*-Schnittstelle am Rechner angeschlossen ist. Zur Vermeidung von Störungen durch ungünstige Lichtverhältnisse wurden zusätzlich Infrarotstrahler sowie ein Infrarotfilter am Kameragehäuse angebracht. Die Kamera ist hier senkrecht zur Decke orientiert, um auch bei einer kompletten Drehung der Plattform die Marken im Sichtbereich der Kamera zu behalten. Aus diesem Grund sind die Passmarken an der Decke des Raumes angebracht. Eine Übersicht des gesamten Szenarios ist der Abbildung 6.2 zu entnehmen. In Tabelle 6.1 sind die Merkmale der digitalen Kamera aufgelistet.

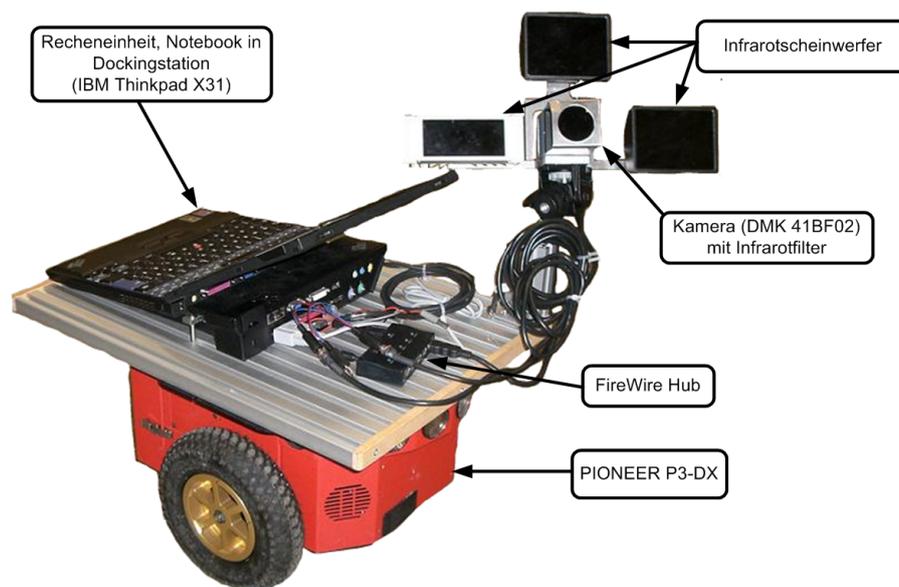


Abb. 6.1: Mobile Plattform zur Anwendung der kamerabasierten Navigation

Aus der Position der Marken resultiert ein Bereich, in dem die Plattform bewegt werden kann, ohne den Sichtkontakt zu den Passmarken zu verlieren. Ein Teil dieses Bereichs wurde in Form eines Rasters gekennzeichnet, um Referenzpunkte für die Messung der Genauigkeit und auch als VCP-Startposition zu erhalten. Die Referenzpunkte werden durch die Kreuzungen des Rasters gebildet und haben ID's erhalten. Das Raster besteht aus 7 horizontalen und 7 vertikalen Linien, woraus 49 Punkte entstehen. Bei der Erstellung des Rasters wurden keine definierten Abstände zwischen den Linien eingehalten, weil die umsetzbare Genauigkeit nicht für die gewünschte Genauigkeit im Millimeterbereich der Posenbestimmung ausreichen würde. Stattdessen wurde das Raster anschließend mit Hilfe der photogrammetrischen Software *Photodeler* vermessen. In diesem Zusammenhang fand gleichzeitig die Vermessung der 3D-Koordinaten der angebrachten Passmarken statt. Details zur Vermessung werden in Abschnitt 6.1 beschrieben. Die Abbildung D.1 aus Anhang D.1 visualisiert das Raster inklusive der 3D-Koordinaten und deren ID's.

Tabelle 6.1: Daten der digitalen FireWire Monochrom-Kamera DMK 41BF02 (The Imaging Source)

Merkmale	
Video-Formate @ Bildrate	1280 x 960 Y800 @ 15, 7.5, 3.75 fps
Empfindlichkeit	0.5 lx bei 1/7.5s, Verstärkung 20 dB
Belichtungszeit	1/10000 bis 30 s
CCD-Sensor	1/2" Sony CCD, Progressive Scan
Abmessungen	H: 50.6 mm, B: 50.6 mm, L: 50 mm
Gewicht	265 g
Datenschnittstelle	IEEE 1394 FireWire
Versorgungs-Spannung	8 bis 30 VDC

Als Recheneinheit zur Ausführung der Software wird ein IBM Thinkpad X31 Notebook verwendet, welches mit Hilfe einer fest installierten Dockingstation auf der Plattform fixiert wird. Die Leistungsdaten des Notebooks sind in der Tabelle 6.2 übersichtlich dargestellt. Zudem bietet die Dockingstation den notwendigen seriellen Anschluss zur Verbindung des Notebooks mit dem mobilen System *PIONEER P3-DX*. Die Plattform verfügt über eine feste Achse und ein

Tabelle 6.2: Leistungsdaten des genutzten Notebooks

Notebook IBM Thinkpad X31	
Komponente	
Prozessor	1.5 GHz Intel Pentium M
Arbeitsspeicher	512 MB
Festplatte	35 GB 5200 rpm
Betriebssystem	Microsoft Windows XP Professional SP 2

Stützrad. Die Lenkung wird über das Achsgetriebe durch unterschiedliche Geschwindigkeiten der Antriebsräder vorgenommen, dadurch ist dieser Roboter in der Lage, sich auf der Stelle zur drehen. Diese Eigenschaft eignet sich ausgezeichnet zum Manövrieren auf engem Raum.



Abb. 6.2: Gesamtübersicht der Testumgebung

6.1 Vermessung der Testumgebung mittels Photogrammetrie

Die Vermessung der Testumgebung erfolgt mit der Photogrammetrie Software *Photomodeler*. Diese Methode ist wesentlich präziser als die Vermessung mit einem Maßband oder einem Zollstock, insbesondere die Vermessung der Passmarken Koordinaten würde sich mit herkömmlichen Mitteln aufgrund der in drei Meter Höhe befindlichen Passmarken als sehr schwierig erweisen. Im Rahmen der zur Verfügung stehenden Mittel ist die photogrammetrische Vermessung somit die präziseste Variante.

Zur Durchführung der Vermessung werden Fotos der Testumgebung benötigt, die zur Festlegung der Referenzpunkte in die Software importiert werden. Bei der Aufnahme der Fotos sind folgende Punkte zu beachten.

- Die Zoom- und Fokuseinstellungen der genutzten Kamera dürfen während der Aufnahmen nicht verändert werden. Die inneren Kameraparameter müssen für alle Fotos identisch sein, um präzise Ergebnisse erzielen zu können.
- Die Fotos sollten aus verschiedenen Positionen und Winkeln (Perspektiven) erstellt werden.
- Die Fotos sollten die gesamte Testumgebung abbilden, damit der Zusammenhang der gewählten Referenzpunkte mit geringstem Fehler hergestellt werden kann.
- Zudem sind Fotos aus kürzeren Entfernungen notwendig, um alle Referenzpunkte genau kennzeichnen zu können. Dies gilt insbesondere für die Rasterpunkte sowie die Passmarken, denn diese sind auf Fotos der gesamten Testumgebung auch bei einer hohen Bildauflösung schwer referenzierbar.
- Gelingt es nicht die gesamte Testumgebung in der nötigen Qualität auf einem Foto festzuhalten, so dass die markanten Referenzpunkte auf diesem Foto referenziert werden können muss die Verbindung über weitere Referenzpunkte erfolgen. Bei diesem Vorgehen ist jedoch eine zusätzliche Ungenauigkeit der Messergebnisse zu berücksichtigen.

Zur Vermessung wird im *Photomodeler* ein neues Projekt erstellt, bei dem zunächst die verwendete Kamera angegeben werden muss. Die Kamera kann aus einem zuvor erstellten Kalibrierungsprojekt zur Verfügung gestellt werden. Diese Option empfiehlt sich aus Gründen der Zuverlässigkeit der Kalibrierung mit dem *Photomodeler*. Zudem wird diese Variante ebenso für die Bestimmung der internen Parameter der auf der Plattform installierten Kamera genutzt. Neben den Referenzpunkten aus dem Testraster und den Passmarken ist die Verwendung von weiteren markanten Punkten als Referenzen angebracht, um die Genauigkeit des Messergebnisses zu fördern. Durch die Referenzierung dieser Punkte auf Fotos von verschiedenen Perspektiven wird von der Photogrammetrie Software ein Modell erstellt, welches alle Punkte enthält. Zur Bereitstellung der 3D-Koordinaten der Referenzpunkte muss der jeweilige Punkt auf mindestens zwei Fotos zugeordnet werden. Zur Erhöhung der Genauigkeit sollte jeder Punkt jedoch auf mehreren Fotos zugeordnet werden.

Zu Beginn der Projekterstellung sollten nur wenige Fotos dem Projekt hinzugefügt werden. Die Software kann die Vermessung nur vornehmen, wenn jedes der im Projekt enthaltenen Fotos über mindestens sechs Referenzpunkte mit einem anderen Fotos korrespondiert. Es ist sinnvoll, nach dem Hinzufügen jedes weiteren Fotos und dem Zuordnen der Punkte eine Vermessung durchzuführen, um die Einhaltung der oben genannten Bedingungen sicherzustellen. Nach dem Hinzufügen aller Fotos und der Vermessung des Projektes können die Residuen der markierten Punkte unter Angabe eines Faktors angezeigt werden. Die Residuen geben die Ungenauigkeit des jeweiligen Referenzpunktes an.

Zur Nutzung der Messergebnisse in der Testumgebung wird zusätzlich der Ursprung des Welt-KS im Punkt mit der ID 19 festgelegt. Des Weiteren erfolgt die Festlegung der x- und y-Achse, wodurch gleichzeitig die z-Achse und somit die Ausrichtung des Welt-KS festgelegt wird. Eine Skalierung des Projektes auf den realen Maßstab ist nicht zwingend notwendig, jedoch hilfreich

bei der späteren Evaluierung der auf der Plattform gemessenen Werte. Die Skalierung wird mit Hilfe einer präzise bekannten Strecke zwischen zwei Referenzpunkten umgesetzt. In Abbildung 6.3 wird der sogenannte *Scale/Rotate* Vorgang im *Photomodeler* dargestellt.



Abb. 6.3: Festlegung des Ursprungs und der Ausrichtung des Weltkoordinatensystems sowie die Skalierung des Projektes im Photomodeler

Ist die Vermessung in der gewünschten Genauigkeit abgeschlossen, sind die beteiligten Punkte übersichtlich zusammen mit ihren ID's und 3D-Koordinaten in einer Tabelle darstellbar. Zusätzlich kann unter anderem die Präzisionsangabe und das Residuum der einzelnen Punkte in die Tabelle eingebracht werden. Zur weiteren Verwendung der Daten wird der Export der Tabelle in eine Textdatei empfohlen. Das Ergebnis der vorgenommenen Vermessung der Testumgebung wird in Anhang D in tabellarischer Form (Tabelle D.1) abgebildet.

6.2 Präzision der VCP-Bestimmung

Zur Messung der Präzision werden die im vorherigen Abschnitt beschriebenen Messpunkte als Referenzpunkte verwendet. Um eine gute Verteilung der Messwerte zu erreichen, ohne alle Referenzpunkte zu messen, werden nur ausgewählte Punkte des Rasters mit dem mobilen System nachgemessen. Die Messung der vier Eckpunkte, weiterer Eckpunkte im inneren des Rasters und des Mittelpunktes ergibt eine sinnvolle Verteilung. Diese Punkte entsprechen den ID's 19, 25, 61, 67, 35, 37, 49, 51 und 43. Die Visualisierung der Messpunktverteilung kann der Abbildung D.2 aus Anhang D.2 entnommen werden.

Eine Millimeter genaue Messung ist mit den zur Verfügung stehenden Mitteln nicht umsetzbar. Eine Abweichung im Bereich von wenigen Millimetern ist bei der Messung realistisch. Die Montage eines Profils an der mobilen Plattform, das kurz über dem Boden abschließt, führt zu einer Genauigkeit der Plattform-Positionierung von 2-3 Millimetern. Das mobile System wird bei jeder Messung mit der äußeren Profilkante direkt über dem zu messenden Referenzpunkt

positioniert. Um diesen Punkt nun mit der mobilen Plattform zu messen muss zuvor jedoch der VCP ebenfalls auf die äußere Profilkante festgelegt werden. Die Festlegung des VCP erfolgt zum Zeitpunkt der initialen Posenbestimmung der Plattform in der Grundposition (vgl. Abschnitt 4.3.1).

Befindet sich der VCP bei der jeweiligen Messung über dem zu messenden Referenzpunkt wird die ausgegebene Plattform-Pose, die dem VCP entspricht, abgelesen und notiert. Zudem wird bei jeder Messung die Anzahl der an der Messung beteiligten Passmarken festgehalten. Die vorgenommenen Messungen wurden in der Tabelle D.2 in Anhang D.3 dokumentiert. Die Abweichungs-Vektoren der Messwerte werden in der Abbildung 6.4 visualisiert. Die Darstellung erfolgt zweidimensional, demzufolge werden die geringen Abweichungen in der z-Achse nicht abgebildet. Die gestrichelten Linien stellen die Verbindung zwischen den Referenzpunkten dar, während die durchgezogenen Linien die Verbindungen zwischen den Messwerten wieder spiegeln. Das visualisierte Raster und die Abweichungen sind maßstabsgetreu abgebildet. Die Werte in den Klammern an den jeweiligen Punkten geben die resultierenden Abweichungen der Ist-Werte zu den Soll-Werten in (x/y/z)-Richtung in Zentimetern an.

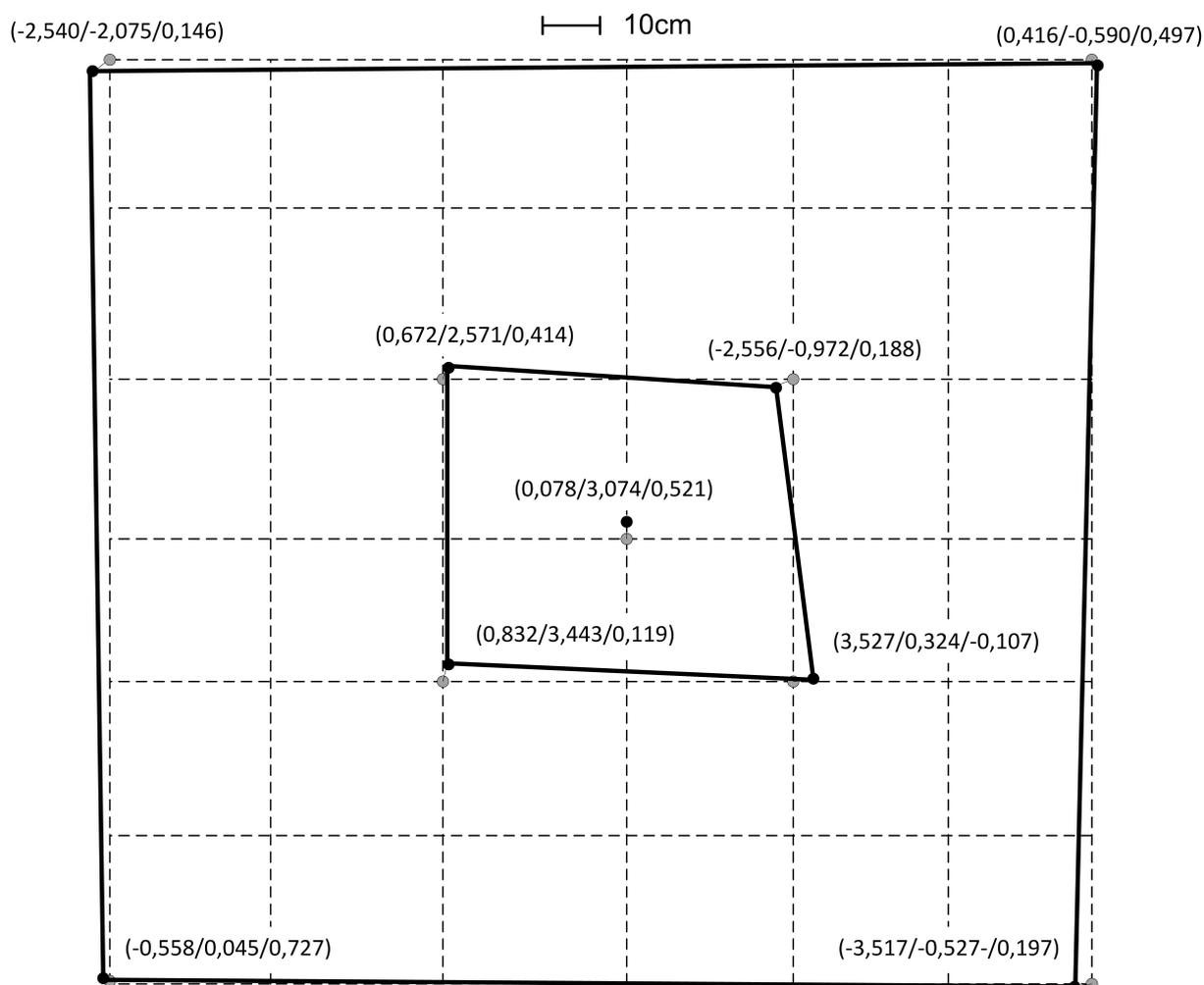


Abb. 6.4: Visualisierung der Abweichungen, der auf der Plattform gemessenen Punkte zu den Referenzpunkten

Zur Berechnung der Abweichung pro Koordinaten wird die Länge des Abweichungs-Vektors angegeben. Die Länge entspricht der 2-Norm und wird wie folgt berechnet (vgl. [Herrmann (2004)] S.20).

$$\|\vec{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i^2|} \quad (6.1)$$

In diesen Fall werden die einzelnen Abweichungen in den Koordinatenachsen für die Berechnung der 2-Norm verwendet.

$$\|\vec{x}\|_2 = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (6.2)$$

Aus der Auswertung der gemessenen Werte in Bezug zu den Referenzwerten ergeben sich Abweichungen in den einzelnen Achsen von bis zu 3,5 cm und bei den 2-Normen von ebenfalls bis zu 3,5 cm. Die mittlere 2-Norm der gemessenen Werte liegt bei 2,69 cm. Das Ergebnis der Auswertung der Präzision entspricht somit nicht dem angestrebten Bereich von einigen Millimetern, sondern von einigen Zentimetern.

Die Genauigkeit der Posenbestimmung wird durch folgende Faktoren und Komponenten maßgeblich beeinflusst.

1. Ungenauigkeiten bei der Vermessung der Testumgebung, insbesondere bei der Ermittlung der 3D-Passmarkenkoordinaten. Wenn die gesamte Testumgebung nicht auf einem Foto abgebildet werden kann, können nicht alle Referenzpunkte zusammen zugeordnet werden. Das hat zur Folge, dass die Verbindung des Testrasters und der Passmarkenkoordinaten über andere Referenzpunkte und andere Fotos erfolgen muss. Dadurch kann das gesamte Weltmodell an Genauigkeit verlieren.
2. Die verwendeten Kameras für die Aufnahme der Fotos zur Vermessung der Testumgebung und für die Navigation müssen festen Fokuseinstellungen und Belichtungszeiten unterliegen. Wird das nicht beachtet, werden die internen Parameter, insbesondere die Verzeichnungsparameter und die Bildweite, bereits bei der Kalibrierung der internen Parameter verfälscht. Dies führt zu einem zu ungenauen Ergebnissen bei der Vermessung der Testumgebung und zum anderen zur Ungenauigkeit bei der Bestimmung der Pose.
3. Die Ermittlung der 2D-Koordinaten der Passmarken durch das Tracking-Modul sollte Sub-Pixel genau erfolgen.

Die ersten beiden Punkte konnten aufgrund der zur Verfügung stehenden Mittel nicht optimal berücksichtigt werden. Es wurde eine handelsübliche digitale Kompaktkamera für die Vermessung der Testumgebung genutzt. Bei dieser Kamera ist die manuelle Fokussierung einstellbar, jedoch genügt die resultierende Qualität der Fotos für die Vermessung der Testumgebung nicht, um eine präzise Referenzierung der Messpunkte und der Passmarken auf einem Foto vorzunehmen.

In der bereits abgeschlossenen Bachelorarbeit zur Identifikation der Marken wurde der Messfehler der Kreismitte bei der angewandten Positionsbestimmung der Passmarken bereits untersucht und bei unter fünf Metern mit maximal 2,47 Prozent angegeben (vgl. [Wippich (2008)]). Bei der Entfernung der Kamera zur Decke von unter drei Metern ist von einem noch geringeren Fehler auszugehen, womit die Bestimmung der Pixelkoordinaten der Passmarken im Sub-Pixelbereich liegt. Eine genauere Messung der Pixelwerte ist mit den zur Verfügung stehenden

Tabelle 6.3: Durchschnittliche Laufzeiten der einzelnen Impresario Module und die Gesamtlaufzeit als Summe der einzelnen Modul-Ausführungszeiten

Impresario-Modul	ØLaufzeit in ms
CrossRationTargets (Tracker)	181,71
Navigator (PoseEstimation)	10,01
PathPlanningFollowing (Bahnplanung)	0,15
	Σ 191,87

Mitteln nicht realisierbar. Im Bereich des *Tracker*, also im letzten Punkt, ist die Abweichung der Messwerte der Posenbestimmung also nicht zu suchen.

6.3 Zeitverhalten unter realen Bedingungen

Die Messung der Ausführungszeiten des Systems erfolgt per Software direkt in den Modulen, die gemessenen Zeiten werden in der Log-Datei gespeichert sofern die Log-Stufe *DEBUG* aktiviert ist. Aufgrund der Ausführung der einzelnen Module in Impresario muss das Zeitverhalten der Module einzeln jeweils im Modul gemessen werden. Die Werte können aufgrund der sequentiellen Ausführung der Module nach der Messung addiert werden. In der Tabelle 6.3 werden die durchschnittlichen Laufzeiten der einzelnen Module aufgelistet und zusammengefasst. Die Messung der Laufzeiten erfolgte beim Abfahren einer vorgegebenen Strecke, bei der vier Punkte im Testraster als Teilziele angegeben wurden. Nach dem Erreichen des letzten Teilziels wurde die Strecke erneut abgefahren. Für die Bildung des Mittelwertes der einzelnen Modullaufzeiten wurden 20 nicht zusammenhängende Messungen berücksichtigt. Die Messwerte der einzelnen Module sind in Anhang D.3 in den Tabellen (D.3, D.4 und D.5) erfasst.

Die Analyse der Geschwindigkeit des *Trackers* aus [Wippich (2008)] kann nicht genutzt werden, weil die dort verwendeten Bilder lokal gespeichert waren und von einem leistungsfähigeren Rechner durchgeführt wurden. Aus [Wilms (2008)] sind zu diesem Zeitpunkt noch keine Ergebnisse bekannt.

Ersichtlich aus den gemessenen Laufzeiten benötigt das Modul zum Identifizieren der Passmarken deutlich die längste Laufzeit mit einem Durchschnitt von 181,71 Millisekunden. Dagegen fallen die Laufzeiten der Posenbestimmung und des Bahnplanung-Moduls mit 10,01 und 0,15 Millisekunden sehr gering aus.

Der erwartete Mehraufwand des *Tracker*-Moduls entsteht durch die aufwendige Bildverarbeitung. Diese hängt in erster Linie von der Auflösung der Bildquelle und der Rechenleistung des eingesetzten Notebooks ab. Zudem spielen die Anzahl der erkannten Marken im Bild eine wesentliche Rolle. Mit steigender Anzahl der Marken wird die Laufzeit erhöht.

Die Ausführungszeit der Posenbestimmung hängt hauptsächlich von der Anzahl der notwendigen Iterationen bei der Ausgleichsrechnung ab, dieser Faktor ist nicht beeinflussbar. Die Methode zur Lösung des Gleichungssystems hat ebenfalls Einfluss auf die Laufzeit. Während die QR-Zerlegung weniger Rechenaufwand benötigt als die SVD Variante, ist die Letztere numerisch stabiler. In dieser Arbeit ist die Stabilität vorrangig, aus diesem Grund wird an der SVD festgehalten. Die Anzahl der erkannten Passmarken spielt auch beim Aufstellen des Gleichungssystems eine Rolle.

Die Planung und Verfolgung der Bahn wird mit nahezu konstantem Zeitaufwand durchgeführt. Es findet lediglich die Berechnung der Zielkoordinaten und die Entscheidung des nächsten Kommandos für die mobile Plattform statt.

Obwohl beim Messvorgang die bereits aktivierte Verfolgung der erkannten Passmarken zu einer Verkürzung der Laufzeit führt, kann insgesamt lediglich eine Abtastrate von 5,214 Hertz erreicht werden. Die angestrebten 10 Hertz als Abtastrate werden nicht erreicht, jedoch kann dieses Defizit ausgeglichen werden. Für die Anwendung in dieser Arbeit bedeutet das eine Einschränkung der Geschwindigkeit des mobilen Systems.

6.4 Berücksichtigung der nicht messbaren Anforderungen

Die in der Zielsetzung dieser Arbeit in Abschnitt 1.2 aufgestellten Rahmenbedingungen sind bis auf die Abtastrate und die Genauigkeit nicht direkt messbar. Die Auswertung dieser Punkte, bezüglich der realisierten Lösung kann demzufolge nicht mit Zahlen belegt werden. Im Folgenden werden diese Punkte einzeln aufgegriffen und deren Erfüllung in der vorgestellten Lösung diskutiert.

- *skalierbar von kleinen Räumen bis auf Lagergröße* - Die Anwendung des vorgestellten Konzeptes ist je nach Anzahl und Verteilung der Passmarken von kleinen Räumen bis auf Lagergröße skalierbar. Die Größe des Raumes oder des Lagers hat keinen Einfluss auf die Genauigkeit, die Ausführungszeit oder anderen Rahmenbedingungen.
- *robust gegen Umgebungseinflüsse* - Umgebungseinflüsse wie Akustik, Hitze, Magnetfelder und Radiowellen haben keine Auswirkung auf die kamerabasierte Navigation, doch ungünstige Lichtverhältnisse können Störungen verursachen. Zur Vorbeugung der Störung durch Lichteinstrahlungen wurde ein Infrarotfilter vor dem Objektiv angebracht. Zudem wurde eine spezielle Halterung zur Installation von Infrarotbeleuchtern am Kameragehäuse angefertigt. Aufgrund des hohen Infrarotanteils im Sonnenlicht gelangt der Einsatz des Filters jedoch bei direkter Sonnenlichteinstrahlung an die Grenzen.
- *geringes Gewicht und Größe* - Die digitale Kamera als Bildquelle und das Sub-Notebook als Recheneinheit führen zu einer geringen Größe und niedrigem Gewicht der notwendigen Hardware, so dass der Einsatz auf einem kleinen mobilen System ohne Probleme gewährleistet werden kann. Die zusätzlich angebrachten Infrarotbeleuchter ändern daran nichts, weil die Größe und das Gewicht in Relation zur Plattform zu vernachlässigen ist.
- *günstig (Hardware-Kosten)* - Die genannten Hardware-Komponenten sind in Bezug auf die Kosten sehr günstig für den industriellen Einsatz. In der Industrie stehen für diese Zwecke häufig weitaus höhere Budgets zur Verfügung. Aus diesem Grund können dort sogar hochwertigere Kameras und Recheneinheiten zum Einsatz kommen.
- *sparsam (Energie)* - Der Energieverbrauch der Komponenten ist ebenfalls gering. Die Kamera sowie die Infrarotbeleuchter werden vom mobilen System gespeist, welches hier nur mit einem 12 Volt Akkumulator mit 7,2 Ah betrieben wird. Das Notebook verfügt wie üblich über eigene Ressourcen zur Energieversorgung.

- *flexibel, günstig änderbar* - Finden Umbauten oder auch Vergrößerungen des Einsatzortes statt, ist der Fahrbereich lediglich um weitere Marken zu ergänzen. Im Gegensatz zu Induktionsschleifen wären keine zusätzlichen Arbeiten notwendig. Die Anbringung der Marken an der Decke ist im Allgemeinen sinnvoll, gerade in Bezug auf mögliche Umbauarbeiten.
- *frei beweglich* - Die realisierte Lösung stellt als Ergebnis eine Pose mit sechs Freiheitsgraden zur Verfügung. Die eingesetzte Plattform ist somit grundsätzlich in allen Richtungen frei beweglich. Bei gewünschten Rotationen um die x- oder y-Achse müssen jedoch weitere Passmarken an Wänden und am Boden angebracht werden, da bei Nick- oder Roll-Bewegungen im Sichtbereich der Kamera keine Marken zu finden wären.

7 Ausblick auf Verbesserungen und Erweiterungen

Ein Ausblick auf weitere Entwicklungsschritte und zusätzliche Verbesserungen findet in diesem Kapitel statt. In diesem Zusammenhang werden folgende Themen betrachtet.

- Auswertung von Odometriedaten zur Ergänzung der Pose
- Verbesserung der Präzision bis in den Millimeterbereich
- Verringerung der Ausführungszeiten zur Erhöhung der Abtastrate
- Verbesserung der Anwendbarkeit der Navigation
- Portierung der Lösung in den FAUST-Kontext

7.1 Auswertung von Odometriedaten zur Ergänzung der Pose

Die in dieser Arbeit entwickelte kamerabasierte Navigation setzt voraus, dass mindestens drei Passmarken im Kamerabild sichtbar sind und erkannt werden. Durch den Einsatz des Infrarotfilters und der dazugehörigen Beleuchtung der Marken mit infrarotem Licht werden Störungen durch schlechte Lichtverhältnisse minimiert. Durch das Auftreten anderer Störungen oder das Fahren außerhalb des Sichtbereiches der Marken kann es trotzdem zu einer Situation kommen, in der zu wenige oder gar keine Marken erkannt werden. Ebenso kann die Kollinearität der erkannten 2D-Passmarkenkoordinaten zu stark abweichenden Werten führen. Zur Sicherstellung der fortlaufenden Navigation des mobilen Systems ist die zusätzliche Auswertung von Odometriedaten eine zuverlässige Variante.

Die auf Odometriedaten aufbauende Positionsbestimmung ist durch Reibung, Schlupf und die relative Bestimmung der Position nicht so präzise wie die kamerabasierte Navigation. Für die Unterstützung der Navigation bei Ausfall der kamerabasierten Posenbestimmung ist diese Option dennoch einsetzbar.

Aufgrund der relativen Messung wäre eine Positionsbestimmung ausgehend von der letzten gültigen Pose nötig. Die Daten müssen ständig aufgezeichnet werden, die Auswertung hingegen ist nur nötig, wenn keine neue Pose bestimmt werden kann, um weiter zu navigieren. Sobald das mobile System wieder kamerabasierte Posen ermittelt, könnten die aufgezeichneten Odometriedaten verworfen werden.

7.2 Verbesserung der Präzision bis in den Millimeterbereich

Zur Erlangung der angestrebten Genauigkeit im Millimeterbereich können mehrere Maßnahmen beitragen. Eine Maßnahme ist eine Millimeter genaue Vermessung der 3D-Koordinaten der Passmarken, obgleich die Vermessung der Marken in der Testumgebung für die zur Verfügung stehenden Mittel bereits sehr genau ist.

Durch die Verwendung einer professionellen hochauflösenden Kamera mit manuellen Fokus- und Belichtungseinstellungen kann mit relativ geringem Aufwand eine genauere Vermessung der Passmarkenkoordinaten erfolgen. Mit dem Einsatz einer professionellen Kamera kann zudem die gesamte Testumgebung auf einem Foto dargestellt werden. Aufgrund der hohen Auflösung und Qualität ist die Referenzierung aller Referenzpunkte und Passmarken umsetzbar. Die resultierende höhere Präzision der Passmarkenkoordinaten und somit des gesamten Modells führt zu kleineren Abweichungen bei der Posenbestimmung.

Weitere Passmarken, die in einem Bild erkannt werden, fördern die Genauigkeit ebenfalls. Durch die Verwendung von codierten konzentrischen Kreisen als Passmarken statt der rechteckigen Marken wäre die Anbringung von jeweils zwei Passmarken im Winkel von 90° nicht notwendig. Somit kann mit der gleichen Markenanzahl eine dichtere Verteilung der Marken erzielt werden. Zudem verringert eine andere Verteilung der Marken die Wahrscheinlichkeit von kollinearen Punkten in der Bildebene. Des Weiteren können die erkannten Bildpunkte bereits vor der Eingabe in die Posenbestimmung auf Kollinearität, durch das Erstellen einer Regressionsgerade, untersucht werden.

7.3 Verringerung der Ausführungszeiten zur Erhöhung der Abtastrate

Die Ausführungszeit der gesamten Navigation wird in erster Linie durch das *Tracker* Modul zur Erkennung der Passmarken beeinflusst. Der Grund dafür ist die aufwendige Bildverarbeitung zur Identifizierung der Marken. Eine deutliche Beschleunigung dieses Moduls wird durch eine Recheneinheit mit einem leistungsfähigerem Prozessor erreicht. Der momentan eingesetzte Pentium M Prozessor mit 1,5 GHz entspricht nicht dem Stand der Technik, somit ist durch den Einsatz einer Recheneinheit mit einem aktuellen Prozessor eine spürbare Beschleunigung der Markenerkennung zu erwarten.

Zur Verringerung der Laufzeit der gesamten Navigation kann zusätzlich die bereits erwähnte Auswertung der Odometriedaten der Plattform genutzt werden. Wenn die Pose durch die Odometrie ergänzt wird, ist die Pose ständig bekannt. Die kamerabasierte Posenbestimmung wird dann genutzt, um die Pose in puncto Genauigkeit zu korrigieren. Dieses Vorgehen ist in der Praxis eine gängige Methode bei Navigationssystemen. In diesem Zusammenhang kann in Situationen, in denen die Präzision vernachlässigt werden kann, die Geschwindigkeit der Plattform erhöht und bei wenig Raum wieder verringert werden.

Für den Einsatz im Szenario der Testumgebung wird ein ähnliches Vorgehen bereits ohne die Auswertung von Odometriedaten umgesetzt. Bei der direkten Ansteuerung von Zielen und der

Drehung in die Zielrichtung, wird die Bewegungsgeschwindigkeit der Plattform abhängig von Entfernung und Winkel zum Ziel angepasst. Diese Methode hat schnelle Bewegungen im Fernbereich und langsamere Bewegungen im Nahbereich von Zielen zur Folge, denn im Nahbereich wird eine sehr genaue Posenangabe benötigt. Dadurch wird die Ausführungszeit der Anwendung nicht erhöht, jedoch wird das Abfahren virtueller Spuren deutlich beschleunigt.

7.4 Verbesserung der Anwendbarkeit der Navigation

Zur Vereinfachung der Anwendbarkeit des Navigations-Moduls sind mehrere Schritte denkbar. Eine große Erleichterung ist die Nutzung von Scale-invariant Feature Transform (SIFT) Bildmerkmalen als Orientierungspunkte für die Posenbestimmung, anstelle von aufwendig anzufertigenden Passmarken. Zudem entfällt dann die fehlerbehaftete Vermessung der 3D-Koordinaten als Faktor der Ungenauigkeit und es stünden jederzeit mindestens drei Punkte für die Posenbestimmung zur Verfügung. Ein großer Nachteil dieser Variante ist jedoch die noch aufwendigere Bildverarbeitung und die damit verbundene Ausführungszeit. Dieser Nachteil lässt sich aber durch die in Zukunft ständig wachsende Prozessorleistung kompensieren.

Eine einfachere Anwendung der vorhandenen Lösung kann durch die einmalige Bestimmung der Kamera-Pose in Bezug zur mobilen Plattform nach der Installation der Kamera erfolgen. Solange die Pose der Kamera auf der Plattform nicht verändert wird, muss die initiale Posenbestimmung in der Grundposition nicht bei jedem Start der Navigation erneut durchgeführt werden. Durch diese Vereinfachung kann die Navigation an jedem Punkt des Fahrbereiches gestartet werden. Die Pose der Kamera setzt sich in Bezug zum mobilen System aus der Kamera-Pose und der VCP-Grundposition zusammen und kann ebenfalls in einer Konfigurationsdatei hinterlegt werden.

Die Anwendung des Moduls in Impresario sollte zusätzlich das Auswählen der Konfigurationsdateien der Navigation unterstützen. Die vorhandenen Konfigurationsdateien enthalten die Kameraparameter, die Schätz-Pose für die erste Posenbestimmung sowie die Festlegung des VCP der Plattform. Eine weitere Datei enthält die ID's und 3D-Koordinaten der genutzten Passmarken. Gleiches gilt für das Bahnplanungs-Modul, in das die Zielpunkte ebenfalls durch eine Konfigurationsdatei geladen werden. Die Dateinamen und der Pfad zu den Dateien werden momentan fest kodiert, was eine Anpassung der jeweiligen Datei notwendig macht.

7.5 Portierung der Lösung in den FAUST-Kontext

Für die Portierung der kamerabasierten Navigation in den Kontext des FAUST-Projektes, zum Beispiel auf das SCV, sind mehrere Schritte notwendig. Dabei stellt sich die Frage, ob auf dem SCV ebenfalls ein eigener Rechner für die Navigation eingesetzt wird, oder ob die Navigation auf einem der Embedded GEME-2000 Rechnern mit Pentium III 650 MHz Prozessoren ausgeführt werden soll. Bei der ersten Variante könnten die bestehenden Module der Navigation unverändert genutzt werden, der Navigationsrechner hingegen muss an den Controller Area Network (CAN)-Bus zur Übermittlung der Navigationsdaten an das SCV angeschlossen werden. Zum Senden der Daten über den CAN-Bus ist zudem ein weiteres Modul zu entwickeln.

Wird die Navigation auf einem der GEME-2000 Rechnern vorgenommen, müssen zunächst alle Module zur Ausführung unter Linux mit dem GNU C Compiler (GCC) kompiliert werden. Da alle Module und die verwendeten Drittanbieter-Software Komponenten auch mit dem GCC kompilierbar sind, sind für diesen Schritt nur wenige Compiler abhängige Änderungen vorzunehmen. Aufgrund der relativ niedrigen Taktrate der GEME-2000 Rechner von 650 MHz wird die notwendige Bildverarbeitung dabei zum kritischen Teil. Die resultierende Abtastrate würde weit unter der Bisherigen liegen und nicht mehr akzeptable Ausführungszeiten liefern. Als Konsequenz muss die Markenerkennung ausgelagert oder leistungsfähigere Rechner eingesetzt werden.

8 Zusammenfassung

Die in dieser Arbeit entwickelte kamerabasierte Navigation ermittelt die Pose von mobilen Systemen mit Hilfe einer auf der Plattform installierten Kamera. Die Orientierung der Kamera erfolgt an definierten Passmarken. Hierzu wird zunächst die Kamera-Pose unter Verwendung eines 13 Parameter 3D-Kameramodells bestimmt, welche anschließend in die Pose des Koordinatenursprungs der mobilen Plattform, den sogenannten VCP umgerechnet wird. Für die Umrechnung sind Kenntnisse der Kamera-Pose und des VCP im Weltkoordinatensystem zum Startzeitpunkt notwendig. Die Lösung stellt jeweils drei Freiheitsgrade in der Translation und der Rotation bereit. Auf diese Weise ist das Konzept nahezu auf jede Art von Plattform portierbar.

Das Konzept wird durch die Anwendung des realisierten Navigations-Moduls und des Moduls zur einfachen Planung und Verfolgung einer Bahn auf einer mobilen Plattform als Beleg für die Lösung realisiert. Bei dieser ersten Anwendung wird einer virtuellen Spur autonom gefolgt, wobei die Spur durch die Angabe von 3D-Koordinaten definiert wird.

Die Messung der Präzision der Posenbestimmung ergibt Abweichungen von 2-3 Zentimetern und entspricht nicht der angestrebten Genauigkeit im Bereich von Millimetern. Dennoch liefern die gemessenen Posen, für die zur Verfügung stehenden Mittel zur Vermessung der Testumgebung, akzeptable Werte. Die Abweichungen werden nicht von der Posenbestimmung, sondern durch die Ungenauigkeit der Vermessung der Testumgebung hervorgerufen.

Während bei der Abweichung der Genauigkeit die Testumgebung ausschlaggebend ist, wird die Ausführungszeit der gesamten Ermittlung der Plattform-Pose durch das *Tracking*-Modul bestimmt. Dieses Modul unterliegt aufgrund der aufwendigen Bildverarbeitung zur Erkennung und Identifikation der Passmarken einer hohen Laufzeit von durchschnittlich 180 Millisekunden. Insgesamt wird somit eine Abtastrate um 5 Hertz realisiert, die angestrebten 10 Hertz können nicht erreicht werden. Die Verwendung leistungsfähigerer Hardware oder das Einbeziehen von Fahrzeugdaten würde aber zur Kompensation dieser Einschränkung führen.

Bis auf die Anforderungen an die Präzision, die Abtastrate und die Robustheit gegenüber Umgebungseinflüssen werden alle weiteren Anforderungen für den Einsatz unter realen Bedingungen erfüllt. Die Defizite in der Präzision und der Abtastrate können durch beschriebene Maßnahmen ausgeglichen werden. Um den Umgebungseinflüssen in Form von ungünstigen Lichtverhältnissen entgegen zu wirken werden in dieser Arbeit bereits Infrarotscheinwerfer und ein Infrarotfilter eingesetzt. Die weiteren genannten Umgebungseinflüsse beeinflussen die kamerabasierte Navigation nicht.

Des Weiteren wird ein Überblick zu Prinzipien der Positionsbestimmung gegeben, deren Bewertung anhand von aufgestellten Anforderungen vorgenommen wird. Mit der Auswertung der gewonnenen Erkenntnisse wird ersichtlich, dass keine der vorgestellten Techniken zur Anwendung für dieses Szenario priorisiert werden kann. Es existieren jedoch auch keine Anforderungen oder Eigenschaften, die gegen die Verwendung der vorgestellten Lösung sprechen.

Abbildungsverzeichnis

1.1	Szenario der Navigation eines mobilen Systems im Indoor-Bereich	2
2.1	Koordinatensysteme des 3D-Kameramodells	6
2.2	Transformation zwischen Welt- und Kamerakoordinatensystem	7
2.3	Zusammenhang zwischen Bild- und Bildhauptpunktkoordinatensystem	7
2.4	Kameraparameter des Modells im Weltkoordinatensystem	8
2.5	Kalibrieranordnung für die Kalibrierung einer Kamera (vgl. [Eos Systems (2008)])	10
2.6	Bildaufnahme aus Kamera- und Photomodeler-Ansicht	10
2.7	Darstellung der externen Parameter einer Kamera-Pose im Photomodeler	11
2.8	Radiale und tangentielle Auswirkungen der Linsenverzeichnung (vgl. [Eschenburg (2006)])	12
2.9	Zusammenhang der Plattform- und Kamera-Pose	13
3.1	Kardanisch aufgehängte Trägheits- Bewegungseinheit (vgl. [King (2005)])	15
3.2	Blockschaltbild der Posenbestimmung mit Trägheitssensoren (vgl. [Welch und Foxlin (2002)])	15
3.3	Basis Setup für die Hand-Eye-Kalibrierung	23
3.4	Beziehung der Transformationen zu den Koordinatensystemen	24
4.1	Gesamtübersicht zum Konzept der kamerabasierten Präzisionsnavigation	26
4.2	Zusammenhang von Welt- und Kamerakoordinatensystem	27
4.3	Kameralage im Weltkoordinatensystem	28
4.4	Zentralprojektion eines Raumpunktes auf die Bildebene	29
4.5	Blockdiagramm zur Vorgehensweise bei der Posenbestimmung der Kamera	31
4.6	Translations- und Rotationsparameter zwischen Plattform und Kamera	35
4.7	Visualisierung der Posenbestimmung des Fahrzeuges	37
4.8	Transformation der Koordinaten eines Weltpunktes in VCP-Koordinaten	42
4.9	Verwendete Passmarken vertikal und horizontal an der Decke angebracht	44
4.10	Direkte Bahnplanung - Start und Ziel	45
4.11	Bahnplanung im VCP-Koordinatensystem	46
5.1	Gesamtmodulübersicht der entwickelten Lösung	49
5.2	Verknüpfung der Impresario Module	58
6.1	Mobile Plattform zur Anwendung der kamerabasierten Navigation	60
6.2	Gesamtübersicht der Testumgebung	61
6.3	Festlegung des Ursprungs und der Ausrichtung des Weltkoordinatensystems sowie die Skalierung des Projektes im Photomodeler	63
6.4	Visualisierung der Abweichungen, der auf der Plattform gemessenen Punkte zu den Referenzpunkten	64
A.1	Aufnahmen der Kalibrieranordnung aus verschiedenen Perspektiven und Winkeln	84
C.1	Vorwärtstransformation bei der Linsenverzeichnungskorrektur	89

C.2	Indirekte Methode der geometrischen Transformation (vgl. [Meisel (2005)]) . . .	90
D.1	Visualisierung des Testrasters	93
D.2	Verteilung der Messpunkte zur Auswertung der Präzision	95

Tabellenverzeichnis

3.1	Bewertung der Prinzipien gegenüber den Anforderungen	20
5.1	Schnittstellenübersicht der Posenbestimmung	53
5.2	Schnittstellenübersicht der Bahnplanung und -verfolgung	56
5.3	Schnittstellenübersicht der Plattformsteuerung	57
6.1	Daten der digitalen FireWire Monochrom-Kamera DMK 41BF02 (The Imaging Source)	60
6.2	Leistungsdaten des genutzten Notebooks	61
6.3	Durchschnittliche Laufzeiten der einzelnen Impresario Module und die Gesamtlaufzeit als Summe der einzelnen Modul-Ausführungszeiten	66
A.1	Daten der Plattform-Kamera inkl. interner Parameter	83
D.1	3D-Koordinaten des Mess-Rasters und der Passmarken in der Testumgebung . .	94
D.2	Messergebnisse der gewählten Referenzpunkte	96
D.3	Laufzeitmessung des Tracker in Zusammenhang mit der Anzahl der identifizierten Marken	96
D.4	Laufzeitmessung der Posenbestimmung und der resultierenden Laufzeit des Navigators in Zusammenhang mit der Anzahl der Iterationen	97
D.5	Laufzeitmessung zur Planung und Verfolgung der Bahn inklusive des Absetzen des Kommandos an die mobile Plattform	97

Quellcodeverzeichnis

4.1	Pseudocode für die Bahnverfolgung mit integrierter Bahnplanung	47
5.1	Pseudocode des Kontrollflusses der gesamten Navigationssoftware	50
5.2	Korrektur der verzeichneten Bildpunkte zum Einsatz als Modul	51
5.3	Auszug aus der XML-Datei zur Angabe der Markenkoordinaten	52
5.4	Definition der Ein- und Ausgabeparameter in einem Impresario-Modul	58
C.1	Nicht lineare Ausgleichung zur Korrektur der Linsenverzeichnung	91

Glossar

Drittanbieter-Software ist wiederverwendbare Software, die entwickelt wurde, um in verschiedenen Umfeldern unabhängig von dem Hersteller der Software genutzt zu werden.

FAUST besteht aus mehreren Projekten zu den Themen Fahrerassistenzsysteme, Sensorik und Telemetrie in autonomen Fahrzeugen.

Freiheitsgrade eines physikalischen Systems sind Parameter die die Rotation (Drehwinkel) und Translation (Raumkoordinaten) eines Körpers im Raum beschreiben. Die Zahl der Freiheitsgrade ist eine Systemeigenschaft, die die Dynamik des Systems eingrenzt.

Impresario ist eine grafische Oberfläche für die Verknüpfung von Bildverarbeitungsalgorithmen aus der LTI-Lib.

Kalibrierung bezeichnet die initiale Bestimmung der internen Kameraparameter anhand einer aus verschiedenen Perspektiven aufgenommenen Kalibrieranordnung .

LTI-Lib ist eine Open Source Software, die seit 1998 am Lehrstuhl für Technischen Informatik an der Rheinisch Westfälischen Technischen Hochschule Aachen (RWTH) entwickelt wurde. Sie stellt unter anderem Bildverarbeitungsalgorithmen sowie Funktionen der linearen Algebra, Vektor- und Matrizenrechnung und Visualisierungswerkzeuge bereit.

Messkammer oder Messbildkamera ist eine Kamera mit geringen Abbildungsfehlern oder mit Fehlern, die sich durch Kalibrieren genau bestimmen lassen.

Photomodeler ist eine photogrammetrische Vermessungssoftware, die sich in der Forensik, Unfall-Rekonstruktion und Erstellung von 3D-Modellen bereits bewährt hat.

Pose ist die kombinierte Angabe von Position und Orientierung im dreidimensionalen Raum.

Posenbestimmung bezeichnet die Ermittlung der Pose unter Verwendung der durch die Kalibrierung ermittelten internen Kameraparameter.

Tool Center Point ist ein gedachter Referenzpunkt, der sich an geeigneter Stelle am Werkzeug befindet und die Position und Orientierung des Werkzeuges repräsentiert.

Vehicle Center Point entspricht dem Tool Center Point bezogen auf ein Fahrzeug.

Abkürzungsverzeichnis

API	Application Programming Interface
ARIA	Advanced Robotics Interface for Applications
CAN	Controller Area Network
CCD	Charge Coupled Device
DLL	Dynamic Link Library
DOF	Degrees of Freedom, Freiheitsgrade
FAUST	Fahrerassistenz- und Autonome Systeme
GCC	GNU C Compiler
GPS	Global Positioning System
HEC	Hand-Eye-Calibration
ID	Identifikationsnummer
IMAPS	Indoor Distance Measurement and Positioning System
INS	Inertiales Navigationssystem
KS	Koordinatensystem
MEMS	Mikro-Elektromechanisches System
SCV	Sensor Controlled Vehicle
SIFT	Scale-invariant Feature Transform
SVD	Singular Value Decomposition
TCP	Tool Center Point
VCP	Vehicle Center Point

Literaturverzeichnis

- [Barbour und Elwell 2005] BARBOUR, Neil ; ELWELL, John: *Inertial Instruments: Where To Now?* 2005. – URL <http://www.media.mit.edu/resenv/classes/MAS836/Inertialnotes/DraperOverview.pdf> - Zugriff: 3.12.2008
- [Choi u. a. 2006] CHOI, Young M. ; SUH, Young S. ; PARK, Sang K.: Pose Estimation from Landmark-based vision and inertial sensors. In: *SICE-ICASE International Joint Conference* (2006), Oktober, S. 1668–1671. – URL <http://ieeexplore.ieee.org/iel5/4108013/4108014/04109240.pdf?tp=&isnumber=4108014&arnumber=4109240&punumber=4108013>
- [Eos Systems 2008] EOS SYSTEMS INC.: *Photomodeler*, 2008. – URL <http://www.photomodeler.com/index.htm>. – Zugriff: 3.12.2008
- [Eschenburg 2006] ESCHENBURG, Jonas: *Optisches Kameratracking anhand natürlicher Merkmale*. Diplomarbeit Universität Augsburg. 2006. – URL http://www.informatik.uni-augsburg.de/lehrstuehle/mm1/publications/reports/2006_13_esc/tr2006-13.pdf. – Zugriff: 3.12.2008
- [Föhr 1990] FÖHR, Dipl.Ing R.: *Photogrammetrische Erfassung räumlicher Informationen aus Videobildern*, Rheinisch Westfälische Technische Hochschule (RWTH), Dissertation, 1990
- [Göbel 2001] GÖBEL, Jürgen: *Radartechnik Grundlagen und Anwendungen*. VDE Verlag, 2001. – ISBN 3-8007-2582-7
- [Gregor 2006] GREGOR, Sebastian: *Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden*. Bachelorarbeit HAW Hamburg. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf> - Zugriff: 3.12.2008
- [Hallenberg 2007] HALLENBERG, Johan: *Robot Tool Center Point Calibration using Computer Vision*, Linköping University, Department of Electrical Engineering, Diplomarbeit, 2007. – URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-9520> - Zugriff: 3.12.2008
- [Haralick u. a. 1989] HARALICK, Robert M. ; JOO, Hyonam ; LEE, Chung-Nan ; VINAY G. VAIDYA, Xinhua Z. ans ; KIM, Man B.: Pose Estimation from Corresponding Point Data. In: *IEEE Transactions on Systems, Man and Cybernetics* Bd. 19, 1989, S. 1426–1444. – URL <http://ieeexplore.ieee.org/iel1/21/1677/00044063.pdf?tp=&isnumber=&arnumber=44063>

- [Hartley und Zisserman 2003] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple View Geometry in Computer Vision*. Second Edition. University of Cambridge, 2003. – ISBN 0-512-54051-8
- [Herrmann 2004] HERRMANN, Norbert: *Höhere Mathematik für Ingenieure, Physiker und Mathematiker*. Oldenbourg Verlag München Wien, 2004. – ISBN 3-486-27498-8
- [INDOOR 2008] INDOOR: *Fördervorhaben INDOOR*, 2008. – URL <http://www.indoor-navigation.de> - Zugriff: 3.12.2008
- [Jähne 2005] JÄHNE, Professor Dr. B.: *Digitale Bildverarbeitung*. Springer, 2005 (6). – ISBN 3-540-24999-0
- [King 2005] KING, A.D.: *Inertial Navigation - Forty Years of Evolution*. Marconi ElectronicsSystems Ltd. GEC REVIEW (Veranst.), 2005. – URL http://www.imar-navigation.de/download/inertial_navigation_introduction.pdf - Zugriff: 3.12.2008
- [Lemelson 2005] LEMELSON, Hendrik: *Eine Übersicht über In- und Outdoor Positionierungssysteme*. Studienarbeit Universität Mannheim. 2005. – URL http://www.informatik.uni-mannheim.de/pi4.data/content/courses/2005-ws/seminar/Ausarbeitung_Positionierung-Uebersicht_Indoor_und_Outdoor_Positionierungssysteme.pdf. – Zugriff: 3.12.2008
- [log4cxx 2008] APACHE LOGGING SERVICES: *log4cxx*, 2008. – URL <http://logging.apache.org/log4cxx/index.html>. – Zugriff: 3.12.2008
- [LtiLib 2005] RHEINISCH WESTFÄLISCHE TECHNISCHE HOCHSCHULE (RWTH): *LTI Lib*, 2005. – URL <http://ltilib.sourceforge.net/doc/homepage/index.shtml>. – Zugriff: 3.12.2008
- [Ma Feb 1996] MA, Sang D.: A self-calibration technique for active vision systems. In: *Robotics and Automation, IEEE Transactions on* 12 (Feb 1996), Nr. 1, S. 114–120. – URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=481755&isnumber=10242>. – DOI 10.1109/70.481755. – ISSN 1042-296X
- [Manske 2008] MANSKE, Nico: *Verfahren zur Positionsbestimmung im Indoor-Bereich*. Seminararbeit HAW Hamburg. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubi-comp/projekte/master07-08/manske/bericht.pdf>. – Zugriff: 3.12.2008
- [Manske und Jost 2008] MANSKE, Nico ; JOST, Thorsten: *Posenbestimmung in Räumen mit einem 3D-Kameramodell*. Projektbericht HAW Hamburg. 2008. – URL http://svn.kbsvn.net/filedetails.php?repname=Team3&path=%2Ftrunk%2Fmaster%2FPosenbestimmung%2Fdoc%2Freport%2FPJ_Report.pdf&rev=0&sc=0. – Zugriff: 3.12.2008
- [Meisel 1994] MEISEL, Andreas: *3D-Bildverarbeitung für feste und bewegte Kameras*, Rheinisch Westfälische Technische Hochschule (RWTH), Dissertation, 1994

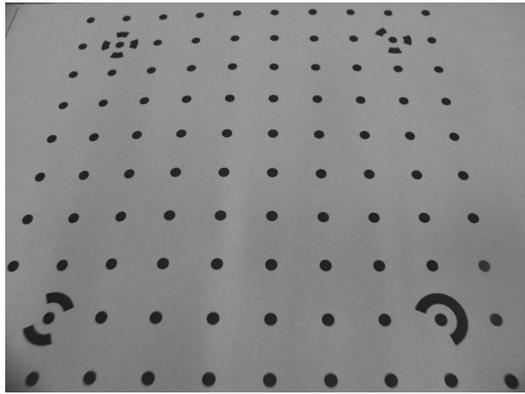
- [Meisel 2005] MEISEL, Prof. Dr.-Ing. A.: *Vorlesungsunterlagen Robot Vision*. HAW Hamburg. 2005. – URL http://www.informatik.haw-hamburg.de/wp_robot_vision.html. – Zugriff: 3.12.2008
- [MobileRobots 2008] MOBILEROBOTS INC.: *Advanced Robotics Interface for Applications*, 2008. – URL <http://www.activrobots.com/SOFTWARE/aria.html>. – Zugriff: 3.12.2008
- [Press u. a. 2007] PRESS, William H. ; TEUKOLSKY, Saul A. ; VETTERLING, William T. ; FLANNERY, Brian P.: *Numerical Recipes - The Art of Scientific Computing*. Third Edition. Cambridge University Press, 2007. – ISBN 978-0-521-88407-5
- [Röfer 2001] RÖFER, Dr.-Ing. Dipl.-Inform. T.: *Bahnplanung*. Universität Bremen, Fachbereich 3, Informatik, Vorlesungsskript "Kognitive Robotik II". 2001. – URL <http://www.informatik.uni-bremen.de/~roefer/kr01/07.pdf>. – Zugriff: 3.12.2008
- [Shiu und Ahmad Feb 1989] SHIU, Y.C. ; AHMAD, S.: Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX=XB$. In: *Robotics and Automation, IEEE Transactions on* 5 (Feb 1989), Nr. 1, S. 16–29. – DOI 10.1109/70.88014. – ISSN 1042-296X
- [Sturm 2000] STURM, Peter: Algorithms for Plane-Based Pose Estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, URL <http://perception.inrialpes.fr/Publications/2000/Stu00b/Sturm-cvpr00.pdf>, June 2000, S. 1010–1017
- [Tsai und Lenz Jun 1989] TSAI, R.Y. ; LENZ, R.K.: A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. In: *Robotics and Automation, IEEE Transactions on* 5 (Jun 1989), Nr. 3, S. 345–358. – DOI 10.1109/70.34770. – ISSN 1042-296X
- [Wei u. a. Feb 1998] WEI, Guo-Qing ; ARBTER, K. ; HIRZINGER, G.: Active self-calibration of robotic eyes and hand-eye relationships with model identification. In: *Robotics and Automation, IEEE Transactions on* 14 (Feb 1998), Nr. 1, S. 158–166. – DOI 10.1109/70.660864. – ISSN 1042-296X
- [Welch und Foxlin 2002] WELCH, G. ; FOXLIN, E.: Motion tracking: no silver bullet, but a respectable arsenal. In: *Computer Graphics and Applications, IEEE* 22 (2002), Nov.-Dec., Nr. 6, S. 24–38. – ISSN 0272-1716
- [Wilms 2008] WILMS, Matthias: *Lokalisierung und Verfolgung von codierten Messmarken mit Hilfe von Bildverarbeitung*. in Bearbeitung Bachelorarbeit HAW Hamburg. 2008
- [Wippich 2008] WIPPICH, Matthias: *Bildverarbeitungsbasierte Identifikation von Positionsmarken für die Präzisionsnavigation*. Bachelorarbeit HAW Hamburg. 2008
- [XercesXML 2008] THE APACHE XML PROJECT: *Xerces-C++ XML Parser*, 2008. – URL <http://xerces.apache.org/xerces-c/>. – Zugriff: 3.12.2008
- [Zhuang Aug 1998] ZHUANG, Hanqi: Hand/eye calibration for electronic assembly robots. In: *Robotics and Automation, IEEE Transactions on* 14 (Aug 1998), Nr. 4, S. 612–616. – DOI 10.1109/70.704231. – ISSN 1042-296X

A Kalibrierung der Kamera des mobilen Systems

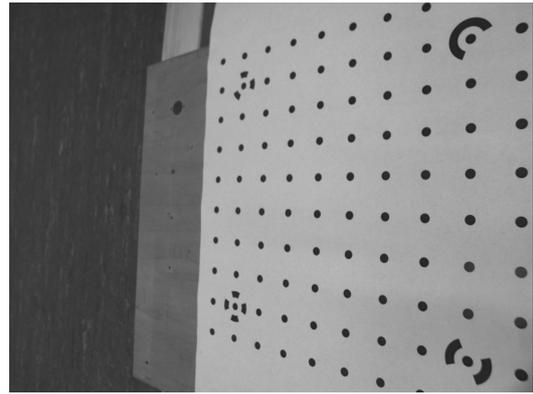
Die internen Parameter sowie die weiteren notwendigen Parameter der auf der Plattform genutzten Kamera wurden unter Verwendung des *Photomodeler* ermittelt und in die Konfigurationsdatei (camera.ini) eingetragen. Eine Übersicht der Parameter ist in Tabelle A.1 zu finden. Zur Bestimmung dieser Parameter wurde ein Kalibrierprojekt im *Photomodeler* erstellt. Die zur Kalibrierung notwendigen Aufnahmen der Kalibrieranordnung sind in Abbildung A.1 dargestellt. Es wurden acht Aufnahmen aus vier Perspektiven und jeweils zwei um 90° gedrehte Positionen vorgenommen. Dabei ist auf die Belegung der Ecken mit Punkten zu achten, denn aufgrund der radialen Verzeichnung sind die Fehler in den Rändern und Ecken am größten.

Tabelle A.1: Daten der Plattform-Kamera inkl. interner Parameter

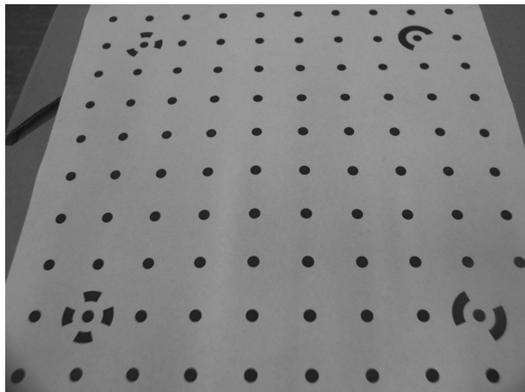
Parameter	Wert
Δu_H [mm]	3.0656
Δv_H [mm]	2.1213
c Bildweite [mm]	5.0929
K1	0.005269
K2	-0.0002452
P1	-0.00005856
P2	0.0003403
Chipbreite [mm]	5.9903
Chiphöhe [mm]	4.5000
Auflösung	1280 x 960



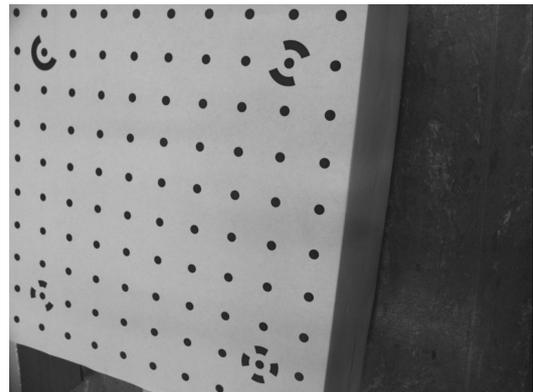
(a) Perspektive 1 Winkel 0°



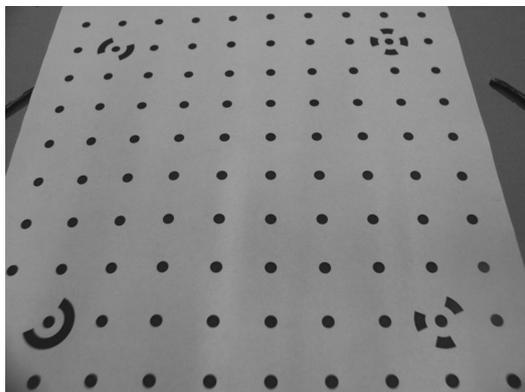
(b) Perspektive 1 Winkel 90°



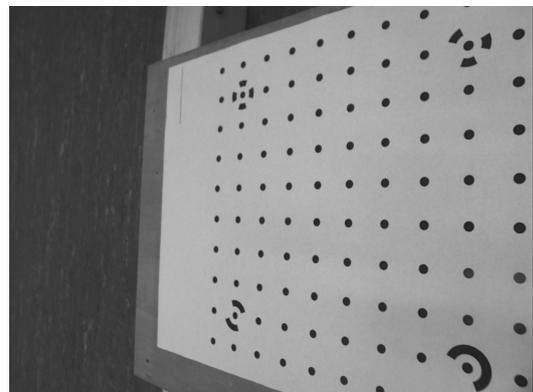
(c) Perspektive 2 Winkel 0°



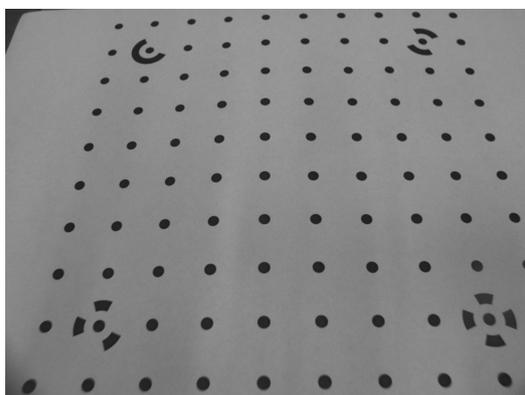
(d) Perspektive 2 Winkel 90°



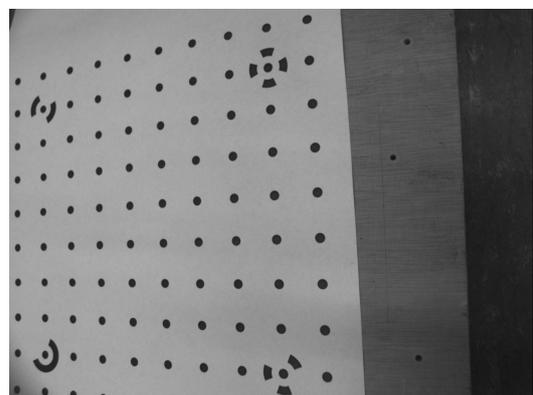
(e) Perspektive 3 Winkel 0°



(f) Perspektive 3 Winkel 90°



(g) Perspektive 4 Winkel 0°



(h) Perspektive 4 Winkel 90°

Abb. A.1: Aufnahmen der Kalibrieranordnung aus verschiedenen Perspektiven und Winkeln

B Berechnungsansätze

B.1 Nichtlineare Ausgleichung mit der Newtonschen Nullstellensuche

Das Newtonsche Näherungsverfahren gilt in der Mathematik als Standardverfahren zur numerischen Lösung von nichtlinearen Gleichungen und Gleichungssystemen. Zu einer gegebenen stetig differenzierbaren Funktion lassen sich Näherungswerte zur Nullstelle finden. In einem Ausgangspunkt wird die Tangente der Funktion bestimmt ($f'(x_n)$), dessen Nullstelle als Näherung der Nullstelle der Funktion $f(x)$ genutzt wird. Von dieser Näherung ausgehend werden weitere Verbesserungsschritte iterativ vorgenommen bis die Änderung einen angegebenen Wert erreicht hat ($|x_{n+1} - x_n| < \varepsilon$). Aus der Funktion (B.1) erfolgt die Iterationsvorschrift (B.2) durch die Umstellung nach x_{n+1} (vgl. [Herrmann (2004)] S. 238-241).

$$f'(x_n) = \frac{f(x_n)}{x_n - x_{n+1}} \quad (\text{B.1})$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (\text{B.2})$$

In den meisten Anwendungen des Verfahrens (in Gleichungssystemen) werden jedoch nicht nur eine sondern mehrere Variablen gesucht (nicht $f(x)$) sondern $f(x_1, x_2, x_3, \dots)$. Dazu ist eine andere Darstellung weise von Newton (B.2) erforderlich. Es erfolgt die Umstellung nach $-f(x_n)$ (vgl. [Herrmann (2004)] S. 250-251).

$$-f(x_n) = f'(x_n) \cdot (x_{n+1} - x_n) \quad (\text{B.3})$$

Die umgestellte Gl. B.3 ist erweiterbar auf Funktionen mit mehreren Variablen x_1, x_2, x_3, \dots auf die Gleichung (B.4). m entspricht der Anzahl der Unbekannten und n der Anzahl der Iterationen.

$$-f(x_{1n}, \dots, x_{mn}) = \frac{\partial f(x_{1n}, \dots, x_{mn})}{\partial x_{1n}} \cdot (x_{1n+1} - x_{1n}) + \dots + \frac{\partial f(x_{1n}, \dots, x_{mn})}{\partial x_{mn}} \cdot (x_{mn+1} - x_{mn}) \quad (\text{B.4})$$

Für die Lösung durch die Aufstellung eines Gleichungssystems werden bei m Unbekannten m Gleichungen benötigt. Das resultierende Gleichungssystem wird durch die Anwendung der Matrizenrechnung auf den Standardansatz $\vec{\varepsilon} = A * \vec{\xi} - \vec{l}$ gelöst (vgl. [Meisel (1994) Kapitel C.1]).

$$\begin{pmatrix} -f_1(x_1, \dots, x_{mn}) \\ \vdots \\ -f_m(x_1, \dots, x_{mn}) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_{1n}} & \dots & \frac{\partial f_1}{\partial x_{mn}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_{1n}} & \dots & \frac{\partial f_m}{\partial x_{mn}} \end{pmatrix} \cdot \begin{pmatrix} x_{1n+1} - x_{1n} \\ \vdots \\ x_{mn+1} - x_{mn} \end{pmatrix} \quad (\text{B.5})$$

B.2 Gleichungen und Ableitungen für die Berechnung der Pose

Hier werden alle Gleichungen aufgeführt die in dieser Form bei der Posenbestimmung angewendet wurden. Die Gleichung $f u_i$ und $f v_i$ werden nach $t_x, t_y, t_z, \alpha, \beta$ und γ abgeleitet. Für die sehr komplexen Ableitungen wurde die Software *Maple* verwendet. Die Ableitungen werden in der Jacobi-Matrix A angewendet.

Ableitungen von $f u_i$ nach $t_x, t_y, t_z, \alpha, \beta, \gamma$.

$$f u_i = \Delta u_H - \frac{c(\cos(\alpha) \cos(\gamma)(x_i - t_x) + (\cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma))(y_i - t_y) + (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \sin(\beta) \cos(\gamma))(z_i - t_z))}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)} - u_i$$

$$\begin{aligned} \frac{\partial f u_i}{\partial \alpha} = & - \frac{C(-\sin(\alpha) \cos(\gamma)(x_i - t_x) + (-\sin(\alpha) \sin(\gamma) + \cos(\alpha) \sin(\beta) \cos(\gamma))(y_i - t_y) + (\cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma))(z_i - t_z))}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)} \\ & + \frac{C(\cos(\alpha) \cos(\gamma)(x_i - t_x) + (\cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma))(y_i - t_y) + (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \sin(\beta) \cos(\gamma))(z_i - t_z))(-\cos(\alpha) \cos(\beta)(y_i - t_y) - \sin(\alpha) \cos(\beta)(z_i - t_z))}{(\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z))^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial f u_i}{\partial \beta} = & - \frac{C(\sin(\alpha) \cos(\beta) \cos(\gamma)(y_i - t_y) - \cos(\alpha) \cos(\beta) \cos(\gamma)(z_i - t_z))}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)} \\ & + \frac{C(\cos(\alpha) \cos(\gamma)(x_i - t_x) + (\cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma))(y_i - t_y) + (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \sin(\beta) \cos(\gamma))(z_i - t_z))(\cos(\beta)(x_i - t_x) + \sin(\alpha) \sin(\beta)(y_i - t_y) - \cos(\alpha) \sin(\beta)(z_i - t_z))}{(\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z))^2} \end{aligned}$$

$$\frac{\partial f u_i}{\partial \gamma} = - \frac{C(-\cos(\alpha) \sin(\gamma)(x_i - t_x) + (\cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma))(y_i - t_y) + (\sin(\alpha) \cos(\gamma) + \cos(\alpha) \sin(\beta) \sin(\gamma))(z_i - t_z))}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)}$$

$$\begin{aligned} \frac{\partial f u_i}{\partial t_x} = & \frac{C \cos(\alpha) \cos(\delta)}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)} \\ & - \frac{C(\cos(\alpha) \cos(\delta)(x_i - t_x) + (\cos(\alpha) \sin(\delta) + \sin(\alpha) \sin(\beta) \cos(\delta))(y_i - t_y) + (\sin(\alpha) \sin(\delta) - \cos(\alpha) \sin(\beta) \cos(\delta))(z_i - t_z)) \sin(\beta)}{(\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z))^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial f u_i}{\partial t_y} = & - \frac{C(-\cos(\alpha) \sin(\delta) - \sin(\alpha) \sin(\beta) \cos(\delta))}{\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z)} \\ & + \frac{C(\cos(\alpha) \cos(\delta)(x_i - t_x) + (\cos(\alpha) \sin(\delta) + \sin(\alpha) \sin(\beta) \cos(\delta))(y_i - t_y) + (\sin(\alpha) \sin(\delta) - \cos(\alpha) \sin(\beta) \cos(\delta))(z_i - t_z)) \sin(\alpha) \cos(\beta)}{(\sin(\beta)(x_i - t_x) - \sin(\alpha) \cos(\beta)(y_i - t_y) + \cos(\alpha) \cos(\beta)(z_i - t_z))^2} \end{aligned}$$

$$\frac{\partial f u_i}{\partial t_z} = - \frac{C(-\sin(\alpha)\sin(\delta) + \cos(\alpha)\sin(\beta)\cos(\delta))}{\sin(\beta)(x_i - t_x) - \sin(\alpha)\cos(\beta)(y_i - t_y) + \cos(\alpha)\cos(\beta)(z_i - t_z)}$$

$$- \frac{C(\cos(\alpha)\cos(\delta)(x_i - t_x) + (\cos(\alpha)\sin(\delta) + \sin(\alpha)\sin(\beta)\cos(\delta))(y_i - t_y) + (\sin(\alpha)\sin(\delta) - \cos(\alpha)\sin(\beta)\cos(\delta))(z_i - t_z))\cos(\alpha)\cos(\beta)}{(\sin(\beta)(x_i - t_x) - \sin(\alpha)\cos(\beta)(y_i - t_y) + \cos(\alpha)\cos(\beta)(z_i - t_z))^2}$$

Ableitungen von fv_i nach $t_x, t_y, t_z, \alpha, \beta, \gamma$.

$$fv_i = \Delta v_H + \frac{c(-\cos(\beta)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} - v_i$$

$$\frac{\partial fv_i}{\partial \alpha} = \frac{c((-\sin(\alpha)\cos(\gamma) - \cos(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} - \frac{c(-\cos(\beta)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))(-\cos(\alpha)\cos(\beta)(y_i-t_y) - \sin(\alpha)\cos(\beta)(z_i-t_z))}{(\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z))^2}$$

$$\frac{\partial fv_i}{\partial \beta} = -\frac{C(-\sin(\alpha)\cos(\beta)\sin(\gamma)(y_i-t_y) + \cos(\alpha)\cos(\beta)\sin(\gamma)(z_i-t_z))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} + \frac{C(-\cos(\alpha)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))(\cos(\beta)(x_i-t_x) + \sin(\alpha)\sin(\beta)(y_i-t_y) - \cos(\alpha)\sin(\beta)(z_i-t_z))}{(\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z))^2}$$

$$\frac{\partial fv_i}{\partial \gamma} = \frac{c(-\cos(\beta)\cos(\delta)(x_i-t_x) + (-\cos(\alpha)\sin(\delta) - \sin(\alpha)\sin(\beta)\cos(\delta))(y_i-t_y) + (-\sin(\alpha)\sin(\delta) + \cos(\alpha)\sin(\beta)\cos(\delta))(z_i-t_z))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)}$$

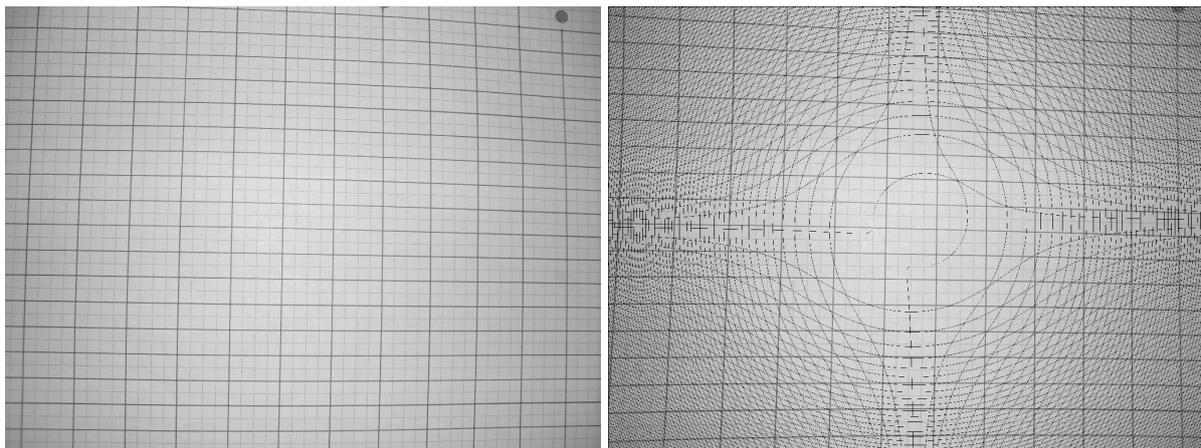
$$\frac{\partial fv_i}{\partial t_x} = -\frac{C\cos(\alpha)\sin(\gamma)}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} - \frac{C(-\cos(\alpha)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))\sin(\beta)}{(\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z))^2}$$

$$\frac{\partial fv_i}{\partial t_y} = -\frac{C(-\cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} + \frac{C(-\cos(\alpha)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))\sin(\alpha)\cos(\beta)}{(\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z))^2}$$

$$\frac{\partial fv_i}{\partial t_z} = -\frac{C(-\sin(\alpha)\cos(\gamma) - \cos(\alpha)\sin(\beta)\sin(\gamma))}{\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z)} - \frac{C(-\cos(\alpha)\sin(\gamma)(x_i-t_x) + (\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma))(y_i-t_y) + (\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma))(z_i-t_z))\cos(\alpha)\cos(\beta)}{(\sin(\beta)(x_i-t_x) - \sin(\alpha)\cos(\beta)(y_i-t_y) + \cos(\alpha)\cos(\beta)(z_i-t_z))^2}$$

C Linsenverzeichnungskorrektur eines kompletten Bildes

Die Verzeichnungskorrektur von kompletten Bildern erfordert eine geometrische Bildtransformation vom Quellbild in das Zielbild, weil die Pixelwerte der verschiedenen Koordinaten während der Korrektur verändert werden. Es ist üblich die indirekte Methode für die Transformation anzuwenden, bei der zu jeder Koordinate des Zielbildes im Quellbild die korrespondierenden Grauwerte bestimmt werden (vgl. Abbildung C.2). Bei der direkten Methode, der Vorwärtstransformation werden nicht alle Pixel im Bild mit einem Wert belegt, es entstehen Lücken. Abbildung C.1 zeigt ein verzeichnetes Bild (Abbildung C.1a) und das korrespondierende, verzeichnungsfreie Bild (Abbildung C.1b). Das verzeichnungsfreie Bild wurde mit dem in dieser Arbeit verwendeten Linsenverzeichnungskorrektur-Modul erstellt.



(a) verzeichnetes Bild einer Digitalkamera

(b) korrespondierendes verzeichnungsfreies Bild

Abb. C.1: Vorwärtstransformation bei der Linsenverzeichnungskorrektur

Zu einem gegebenen verzeichnungsfreien Bildpunkt $\vec{u}_c^\#$ (korrigierte Koordinate) muss die verzeichnete Koordinate $\vec{p}_d^\#$ bestimmt werden von der der Grauwert entnommen wird. Die Gleichungen 4.17 und 4.18 für die Verzeichnungskorrektur aus Abschnitt 2.4 können nicht direkt angewendet werden, weil sie für die Rückwärtstransformation nach u_d und d_d umgestellt werden müssen. Der Grund dafür ist, dass die Gleichung nicht linear und nicht direkt auflösbar ist. Durch die in Abschnitt B.1 beschriebene nichtlineare Ausgleichung mit der Nullstellensuche nach Newton ist die Rückwärtstransformation dennoch realisierbar. Die Anwendung der nichtlinearen Ausgleichung für dieses spezielle Beispiel wird im Anhang C.1 beschrieben.

Die Implementierung der Linsenverzeichnungskorrektur erfolgt unter der Verwendung der [Lti-Lib (2005)], denn diese bietet sowohl Funktionen für die Bildverarbeitung als auch Funktionen und Datentypen zum Lösen von Matrizen- und Gleichungssystemen. Ein Ausschnitt des Quellcodes der iterativen Nullstellensuche ist in Listing C.1 in Anhang C zu finden.

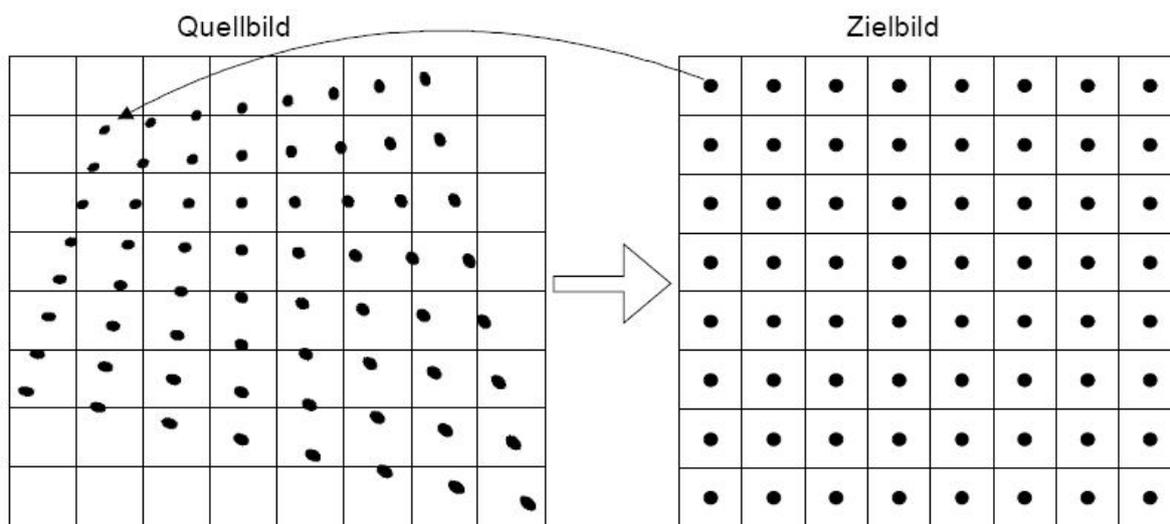


Abb. C.2: Indirekte Methode der geometrischen Transformation (vgl. [Meisel (2005)])

C.1 Anwendung der nichtlinearen Ausgleichung zur Korrektur kompletter Bilder

Für $m = 2$ Unbekannte werden $m = 2$ Gleichungen benötigt. Es werden die Gleichungen (4.17) und (4.17) aus Abschnitt 2.4 für diese beiden Gleichungen genutzt und nach Null umgestellt. Es resultieren die Gleichungen (C.1) und (C.2)

$$f_1(u_d, v_d) = u_d \cdot (1 + K_1 \cdot r^2 + K_2 \cdot r^4) + P_1 \cdot (r^2 + 2u_d^2) + 2P_2 \cdot u_d \cdot v_d - u_c \quad (\text{C.1})$$

$$f_2(u_d, v_d) = v_d \cdot (1 + K_1 \cdot r^2 + K_2 \cdot r^4) + P_2 \cdot (r^2 + 2v_d^2) + 2P_1 \cdot u_d \cdot v_d - v_c \quad (\text{C.2})$$

⇒ Es kann ein lineares Gleichungssystem aufgestellt werden, welches durch Anwendung des Standardansatzes $\vec{\epsilon} = A * \vec{\xi} - \vec{l}$ lösbar ist.

$$\begin{pmatrix} -f_1(u_d, v_d) \\ -f_2(u_d, v_d) \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial f_1}{\partial u_d} & \frac{\partial f_1}{\partial v_d} \\ \frac{\partial f_2}{\partial u_d} & \frac{\partial f_2}{\partial v_d} \end{pmatrix}}_{\text{Jacobi-Matrix}} \cdot \begin{pmatrix} u_{dn+1} - u_{dn} \\ v_{dn+1} - v_{dn} \end{pmatrix} \quad (\text{C.3})$$

Im Projekt wurde das Gleichungssystem durch die in der LTI-Lib implementierten Householder-Transformation gelöst.

Nachfolgend werden die partiellen Ableitungen zu den Funktionen gebildet. Nach der Umformung ergibt sich für die partiellen Ableitungen von f_1 :

$$\frac{\delta f_1}{\delta u_d} = 1 + K_1 \cdot 3u_d^2 + K_2 \cdot (5u_d^4 + 6u_d^2 v_d^2) + P_1 \cdot 6u_d + P_2 \cdot 2v_d \quad (\text{C.4})$$

$$\frac{\delta f_1}{\delta v_d} = K_1 \cdot 2u_d v_d + K_2 \cdot (4u_d^3 v_d + 4u_d v_d^3) + P_1 \cdot 2v_d + P_2 \cdot 2u_d \quad (\text{C.5})$$

Die partiellen Ableitungen von f_2 berechnen sich zu:

$$\frac{\delta f_2}{\delta u_d} = K_1 \cdot 2u_d v_d + K_2 \cdot (4u_d^3 v_d + 4u_d v_d^3) + P_2 \cdot 2u_d + P_1 \cdot 2v_d \quad (\text{C.6})$$

$$\frac{\delta f_2}{\delta v_d} = 1 + K_1 \cdot 3v_d^2 + K_2 \cdot (6u_d^2 v_d^2 + 5v_d^4) + P_2 \cdot 6v_d + P_1 \cdot 2u_d \quad (\text{C.7})$$

C.2 Quellcodeausschnitt der Linsenverzeichnungskorrektur gesamter Bilder

In Quellcodeausschnitt C.1 ist die aufwendige Rückwärtstransformation eines kompletten Bildes zur Herstellung der Verzeichnungsfreiheit aufgeführt.

Quellcode C.1: Nicht lineare Ausgleichung zur Korrektur der Linsenverzeichnung

```

1 // search the pixel in the disturbed source
2 for(int y=0; y<rowSize; y++)
3 {
4     vc= convertMetric(y, false); // position in the correct
        image
5     for(int x=0; x<columnSize; x++)
6     {
7         uc= convertMetric(x, true); // position in the correct
            image
8
9         ud= uc; // guessed start value
10        vd= vc; // guessed start value
11        for (int iteration=0; iteration< 10; iteration++) // max
            10 iterations
12        {
13            // calculate functions...
14            dl[0] = -f1(ud,vd,uc);
15            dl[1] = -f2(ud,vd,vc);
16
17            // calculate derivations
18            Jacobi[0][0]= f1u(ud,vd);
19            Jacobi[0][1]= f1v(ud,vd);
20            Jacobi[1][0]= f2u(ud,vd);
21            Jacobi[1][1]= f2v(ud,vd);
22
23            // update Jacobi-Matrix
24            HouseholderParams.systemMatrix= Jacobi;
25            Householder.setParameters(HouseholderParams);
26
27            // calculate Jacobi-Matrix
28            Householder.apply(dl, dx);
29
30            // check the iteration progress
31            if ( sqrt(pow(dx[0],2) + pow(dx[1],2)) < 0.0000001) {

```

```
32         break;
33     }
34
35     // calculate new approximated coordinates
36     ud += dx[0];
37     vd += dx[1];
38 }
39
40 xSrc= convertPix(ud, true);
41 ySrc= convertPix(vd, false);
42 xDiff = x- xSrc;
43 yDiff = y- ySrc;
44 diff= sqrt(pow(xDiff,2) + pow(yDiff,2));
45 maxDiff= max(maxDiff,diff);
46
47 if(checkBorder((int)xSrc, columnSize) && checkBorder((int)
48     )ySrc, rowSize))
49 {
50     // mit bilinearer Interpolation
51     dst[y][x] = bilinInterpol(xSrc, ySrc, src);
52     dist[y][x] = diff * 21.79;
53 }
54 else
55 {
56     dst[y][x] = 0;
57 }
58 }
```

D Daten zur Testumgebung

In Abbildung D.1 wird das Raster der Testumgebung visualisiert. Die Tabelle D.1 enthält die 3D-Koordinaten der Referenzpunkte für die Genauigkeitsmessung sowie der Koordinaten der Passmarken an der Decke. Die Passmarken werden durch die ID's 1 bis 18 und die Rasterpunkte durch die ID's 19 bis 67 bezeichnet. Die Punkte 19 und 61 bilden die y-Achse während die Punkte 19 und 25 die x-Achse darstellen. Der Koordinatenursprung befindet sich demzufolge in Punkt 19.

D.1 Visualisierung des Mess-Rasters der Testumgebung

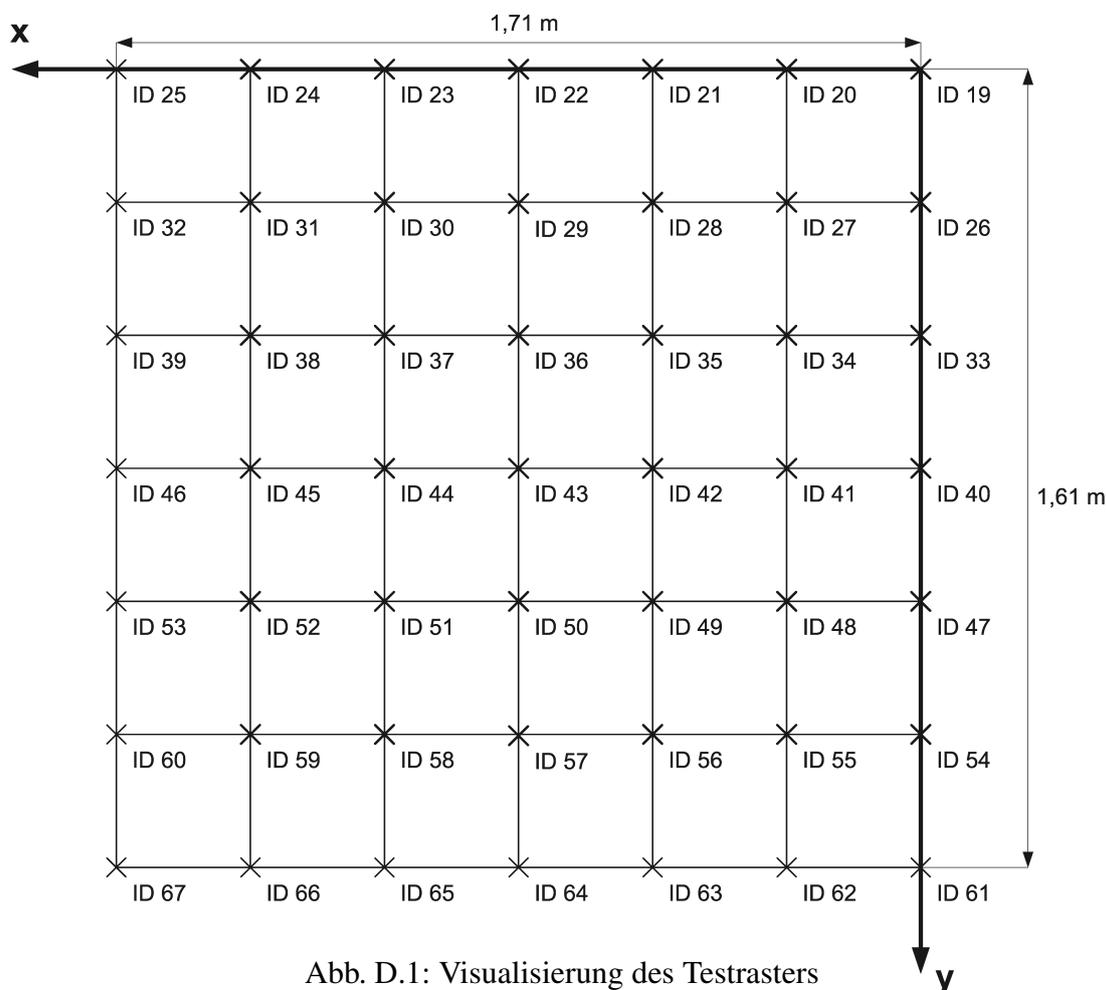


Abb. D.1: Visualisierung des Testrasters

Tabelle D.1: 3D-Koordinaten des Mess-Rasters und der Passmarken in der Testumgebung

ID	X (m)	Y (m)	Z (m)	Id	X (m)	Y (m)	Z (m)
1	-0,480041	-0,372469	3,001389	36	0,81798	0,558859	0,001131
2	-0,471483	0,673212	2,991224	37	1,130116	0,557486	0,004438
3	1,788851	2,204698	2,998926	38	1,422716	0,55474	0,001594
4	1,984882	1,790974	2,999797	39	1,705797	0,555296	0,003542
5	1,796953	1,123313	3,00261	40	-0,00147	0,839867	-0,002222
6	1,956192	0,730657	3,001704	41	0,248839	0,835689	-0,001089
7	1,931464	-0,325517	3,000524	42	0,533167	0,830925	-0,000557
8	1,791792	0,049897	2,998625	43	0,819931	0,827678	0,00054
9	0,980765	-0,36389	3,004344	44	1,127896	0,825226	0,000599
10	0,785649	0,059841	2,999656	45	1,416747	0,824012	0,001346
11	0,944576	0,69539	3,000645	46	1,699918	0,821122	0,00348
12	0,769843	1,091908	2,998456	47	-0,00095	1,086777	-0,001311
13	0,926009	1,785295	2,997377	48	0,249874	1,087927	-0,000872
14	0,753493	2,188051	2,996601	49	0,535785	1,089003	0,000696
15	-0,638261	2,238872	2,990698	50	0,822654	1,088766	0,00106
16	-0,640001	1,065164	2,989742	51	1,127675	1,088625	0,000687
17	-0,626352	0,031305	2,994793	52	1,41222	1,091898	0,001804
18	-0,488194	1,826233	2,992259	53	1,697433	1,094309	0,002409
19	0	0	0	54	-0,000442	1,362218	0,000069
20	0,247977	0,000625	-0,000181	55	0,25139	1,359259	0,000896
21	0,521951	0,000544	-0,001984	56	0,54096	1,35975	-0,000339
22	0,811831	-0,001795	-0,001024	57	0,825382	1,358026	0,000637
23	1,13	-0,001296	0,000074	58	1,125524	1,357028	-0,000429
24	1,432538	-0,00065	-0,000115	59	1,406123	1,35721	0,002507
25	1,710445	0	0	60	1,691414	1,355392	0,001015
26	0,000478	0,265125	-0,001315	61	0,002533	1,618579	0
27	0,247546	0,265062	-0,001046	62	0,25311	1,614591	0,000749
28	0,525289	0,264163	-0,000699	63	0,543232	1,61528	0,001568
29	0,815064	0,264539	0,000629	64	0,827147	1,612343	0,002858
30	1,128834	0,261721	0,00184	65	1,125653	1,612195	0,003576
31	1,428243	0,260971	0,000831	66	1,403671	1,61245	0,003009
32	1,707541	0,261314	0,003555	67	1,687394	1,611701	0,000904
33	-0,002601	0,564339	-0,002575				
34	0,247949	0,562376	0,000057				

D.2 Visualisierung der Messpunktverteilung zur Messung der Präzision

Abbildung D.2 zeigt die Verteilung der Messpunkte im Testraster.

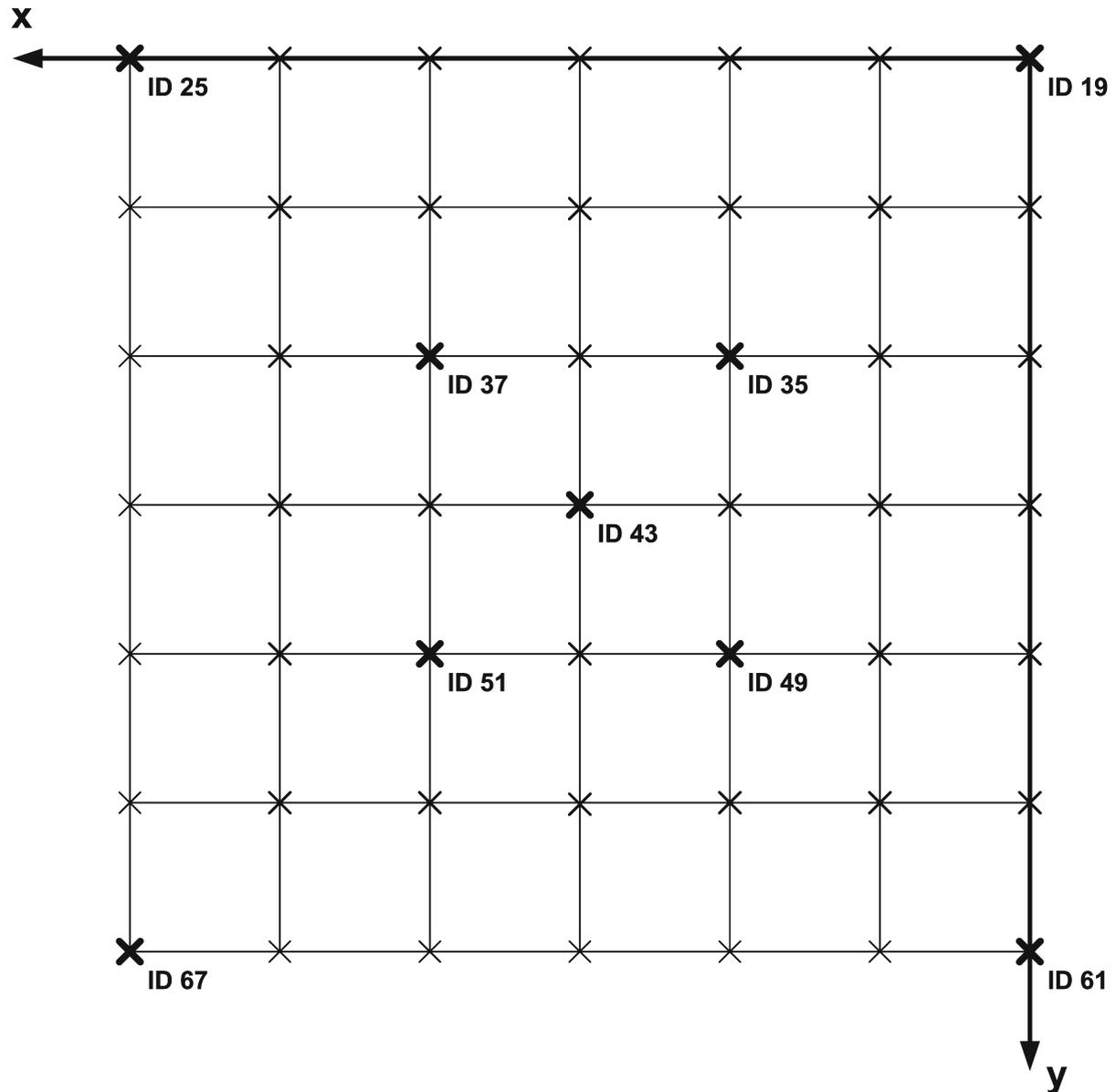


Abb. D.2: Verteilung der Messpunkte zur Auswertung der Präzision

D.3 Messergebnisse der Präzision und der Ausführungszeiten

Tabelle D.2: Messergebnisse der gewählten Referenzpunkte

ID	Marken	γ in $^\circ$	Ist-Werte in cm			Soll-Werte in cm			$\ \vec{x}\ _2$ in cm
			x	y	z	x	y	z	
19	4	0,5	-0,416	0,590	-0,497	0,000	0,000	0,000	0,876
25	4	87,9	173,584	2,075	-0,146	171,045	0,000	0,000	3,283
35	4	87,7	55,334	57,053	-0,180	52,778	56,081	0,008	2,741
37	6	-1,1	112,340	53,178	0,030	113,012	55,749	0,444	2,689
49	4	-90,3	50,052	108,576	0,177	53,579	108,900	0,070	3,543
51	4	1,8	111,935	105,420	-0,050	112,768	108,863	0,069	3,544
61	4	90,6	3,770	162,385	0,197	0,253	161,858	0,000	3,561
67	4	90,7	169,297	161,125	-0,637	168,739	161,170	0,090	0,918
43	4	-0,6	81,915	79,694	-0,467	81,993	82,768	0,054	3,119
\emptyset									2,697

Tabelle D.3: Laufzeitmessung des Tracker in Zusammenhang mit der Anzahl der identifizierten Marken

Tracker		
Messung Nr.:	Anzahl Marken	Zeit in ms
1	4	171,87
2	4	139,56
3	3	131,98
4	3	195,73
5	4	196,87
6	4	177,57
7	4	180,78
8	6	228,88
9	5	231,50
10	5	185,58
11	5	189,21
12	4	135,54
13	4	168,53
14	4	166,02
15	4	156,50
16	3	143,09
17	4	161,20
18	5	244,58
19	5	235,01
20	5	194,28
\emptyset		181,71

Tabelle D.4: Laufzeitmessung der Posenbestimmung und der resultierenden Laufzeit des Navigators in Zusammenhang mit der Anzahl der Iterationen

Messung Nr.:	Iterationen	PoseEstimation	Navigator
		Zeit in ms	Zeit in ms
1	6	4,09	9,23
2	6	4,11	9,27
3	4	2,85	7,73
4	3	2,42	7,59
5	10	6,19	11,40
6	9	5,66	11,47
7	9	5,74	8,44
8	8	5,99	11,15
9	7	5,18	10,49
10	7	5,12	10,52
11	7	5,30	10,70
12	8	5,05	15,46
13	8	5,17	8,18
14	8	5,15	9,09
15	8	5,20	9,47
16	5	4,16	8,91
17	9	5,72	8,67
18	8	5,81	11,27
19	8	5,62	10,75
20	7	5,13	10,43
		Ø 4,98	Ø 10,01

Tabelle D.5: Laufzeitmessung zur Planung und Verfolgung der Bahn inklusive des Absetzens des Kommandos an die mobile Plattform

Messung Nr.:	Zeit in ms	
1	0,14	
2	0,14	
3	0,13	
4	0,13	
5	0,13	
6	0,15	
7	0,14	
8	0,17	
9	0,17	
10	0,14	
11	0,13	
12	0,13	
13	0,13	
14	0,13	
15	0,13	
16	0,13	
17	0,15	
18	0,19	
19	0,18	
20	0,13	
		Ø 0,15

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 5.Dezember 2008

Ort, Datum

Unterschrift