

# Bachelorarbeit

Dirk Jacobi

Identifikation und räumliche Lokalisierung  
skalierungsinvarianter Merkmale für die visuelle  
Navigation

Dirk Jacobi

Identifikation und räumliche Lokalisierung  
skalierungsinvarianter Merkmale für die visuelle  
Navigation

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Technische 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. Wolfgang Fohl

Abgegeben am 25. August 2009

**Dirk Jacobi**

**Thema der Bachelorarbeit**

Identifikation und räumliche Lokalisierung skalierungsinvarianter Merkmale für die visuelle Navigation

**Stichworte**

visuelle Navigation, mobile Roboter, skalierungsinvariante Merkmale, SIFT, Raum-punktvermessung, Kamera

**Kurzzusammenfassung**

Gegenstand dieser Bachelorarbeit ist es, für einen mobilen Roboter eine Navigation zu ermöglichen, die auf Erkennung von markanten Punkten im Raum basiert. Der Roboter soll während seiner Fahrt auf eine Liste mit Raumkoordinaten zurückgreifen können, um anhand von gefundenen markanten Punkten sofort die dazugehörigen Raumkoordinaten ermitteln zu können.

**Dirk Jacobi**

**Title of the paper**

Identification and special localization of scale invariant features for the visual navigation

**Keywords**

mobile navigation, mobile robots, scale invariant feature transform, SIFT, 3D coordinates, cameras

**Abstract**

This Bachelor Thesis is concerned with the development and implementation of a navigation system for mobile robots, based on the detection of prominent points in the room. The Robot has access to a list of 3D coordinates while he is driving and is able to ascertain the 3D coordinates to the spotted prominent point.

## **Danksagung**

An dieser Stelle möchte ich die Gelegenheit nutzen um einigen Leuten zu danken, die mich während meines gesamten Studiums und speziell bei dieser Arbeit unterstützt haben.

An erster Stelle gilt mein Dank meinen Eltern und meinen Großeltern, die mir mein Studium erst ermöglicht und mich zu jeder Zeit unterstützt haben. Ein weiterer Dank geht an die Korrekturleserinnen Brigitte Lösche und Gaby Ramcke.

Natürlich danke ich auch meinem Betreuer Herrn Meisel, der mir mit vielen Anregungen und Erläuterungen zur Seite stand, sowie meinem Zweitkorrektor Herrn Fohl für die Übernahme dieser Aufgabe.

Weiter sind noch Harald und Inge zu erwähnen denen ich auch für Ihre Unterstützung danke. Christoph möchte ich für die tolle Zusammenarbeit während des Studiums danken. Die Zeit mit Dir hat Spaß gemacht.

Zuletzt danke ich ganz besonders Doro, die mir während des Studiums immer zur Seite stand und mich jederzeit motiviert hat. Es ist toll jemanden an seiner Seite zu haben auf den man sich immer verlassen kann.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>
<b>1. Einführung</b>	<b>10</b>
1.1. Motivation . . . . .	10
1.2. Ziel der Arbeit . . . . .	12
1.3. Struktur der Arbeit . . . . .	13
<b>2. Grundlagen</b>	<b>15</b>
2.1. SIFT-Algorithmus . . . . .	15
2.2. Koordinatensysteme . . . . .	16
2.3. Kameraparameter . . . . .	18
2.3.1. Bildhauptpunktverschiebung . . . . .	18
2.3.2. Linsenverzeichnung . . . . .	19
2.4. Homogene Koordinaten . . . . .	20
2.5. Abbildung eines Raumpunktes . . . . .	21
2.6. Verwendete Hard- und Software . . . . .	23
2.6.1. Kamera . . . . .	23
2.6.2. Laborrechner . . . . .	24
2.6.3. Photomodeler . . . . .	24
2.6.4. LTI-Lib . . . . .	24
2.6.5. Impresario . . . . .	25
2.6.6. Visual Studio . . . . .	25
<b>3. Einrichtungsmodul</b>	<b>26</b>
3.1. Einführung . . . . .	26
3.2. Testumgebung . . . . .	26
3.3. Bestimmung der Bildorientierung . . . . .	27
3.3.1. Raummodell als Weltkoordinatensystem . . . . .	27
3.3.2. Berechnung der intrinsischen Kameraparameter . . . . .	29
3.3.3. Berechnung der Kameraposition . . . . .	29
3.4. Raumpunktbestimmung der SIFT-Merkmale . . . . .	30

---

3.4.1. Identifikation von SIFT-Punkten . . . . .	31
3.4.2. SIFT-Punkt Vermessung . . . . .	34
3.5. Genauigkeitsanalyse . . . . .	36
3.5.1. Testumgebung . . . . .	36
3.5.2. Zuordnungssicherheit . . . . .	37
3.5.3. Ortsgenauigkeit der korrekt zugeordneten SIFT-Punkte . . . . .	39
<b>4. Navigationsmodul</b>	<b>42</b>
4.1. Einführung . . . . .	42
4.2. Navigation anhand von vermessenen SIFT-Merkmalen . . . . .	43
4.2.1. Auffindung von SIFT-Merkmalen . . . . .	45
4.2.2. Identifikation von SIFT-Merkmalen . . . . .	45
4.3. Genauigkeitsanalyse . . . . .	47
4.3.1. Analyse der Navigationsgenauigkeit . . . . .	47
4.3.2. Auffindbarkeit von SIFT-Merkmalen . . . . .	50
<b>5. Schlussbetrachtung</b>	<b>52</b>
5.1. Fazit . . . . .	52
5.2. Ausblick . . . . .	53
<b>Literaturverzeichnis</b>	<b>54</b>
<b>A. UML Klassendiagramme</b>	<b>57</b>
<b>B. Messergebnisse des Einrichtungsmoduls</b>	<b>60</b>
<b>C. Inhalt der CD</b>	<b>61</b>

# Tabellenverzeichnis

2.1. Kameradaten der DMK 41BF02 . . . . .	24
4.1. Navigationsgenauigkeit der Kameraposition . . . . .	48
4.2. Navigationsgenauigkeit der Kameralage . . . . .	49
B.1. Messwerte der SIFT-Merkmale des Einrichtungsmoduls . . . . .	60

# Abbildungsverzeichnis

1.1. fahrerloses Transportsystem „carobot“ der Firma psb . . . . .	11
1.2. Testroboter an der HAW Hamburg für die visuelle Navigation . . . . .	12
1.3. künstliche Positionsmarken . . . . .	13
2.1. Panoramabild durch SIFT-Merkmale . . . . .	16
2.2. Zusammenhang der unterschiedlichen Koordinatensysteme . . . . .	17
2.3. Bildhauptpunktverschiebung . . . . .	19
2.4. Linsenverzeichnungseffekte . . . . .	20
2.5. Abbildung eines Raumpunktes . . . . .	22
2.6. Die Kamera DMK 41BF02 mit Objektiv . . . . .	24
3.1. Die Testumgebung . . . . .	27
3.2. Ablauf der manuellen Bildorientierung . . . . .	28
3.3. Kalibrierplatte vom Photomodeler . . . . .	29
3.4. Setzen der anmessbaren Punkte . . . . .	30
3.5. Kameras im 3D-Modell . . . . .	31
3.6. Ablauf der automatisierten Raumpunktbestimmung der SIFT-Merkmale . . . . .	32
3.7. Impresario-Module zur Auffindung von SIFT-Merkmalen . . . . .	33
3.8. Erkannte SIFT-Merkmale im Bild . . . . .	34
3.9. Falsche Zuordnung von SIFT-Punkten . . . . .	36
3.10. Anzahl der richtigen/falschen Zuordnungen von SIFT-Merkmalen . . . . .	37
3.11. SIFT-Punkte mit hohem Verwechslungspotential . . . . .	38
3.12. Anzahl der richtigen/falschen Zuordnungen von SIFT-Merkmalen (mit Verwechslungsfilter) . . . . .	39
3.13. Wirkung der Filterfunktionen auf die Anzahl der SIFT-Merkmale . . . . .	40
3.14. Abweichungen richtig zugeordneter SIFT-Punkte . . . . .	40
3.15. Messfehler durch ungünstige Aufnahmesituation . . . . .	41
3.16. Gute Aufnahmesituation durch einen großen Winkel zwischen den Kameras . . . . .	41
4.1. Übersicht der bisherigen Navigationsmodule für Impresario . . . . .	42
4.2. Ablauf der Erweiterung für das Navigationsmodul . . . . .	44
4.3. Vermessene Merkmalsvektoren im XML-Format . . . . .	46
4.4. Raumpositionen im XML-Format . . . . .	46



---

4.5. Übersicht der neuen Navigationsmodule für Impresario . . . . .	47
4.6. Übersicht zur Auffindbarkeit von SIFT-Navigationspunkten . . . . .	48
4.7. Abweichungen der SIFT-Navigationspunkte . . . . .	49
4.8. Bilder der simulierten Kamerafahrt . . . . .	50
4.9. Verwechslungsfehler bei der Navigation . . . . .	51
A.1. SiftPoint Klassenbeschreibung . . . . .	57
A.2. SameSiftPoint Klassenbeschreibung . . . . .	57
A.3. Camera Klassenbeschreibung . . . . .	58
A.4. Mark Klassenbeschreibung . . . . .	59

# 1. Einführung

In diesem Kapitel wird das Thema vorgestellt. Es wird eine Übersicht auf bisherige Entwicklungen und Anwendungsgebiete gegeben. Des Weiteren werden das Ziel der Arbeit sowie deren Gliederung näher gebracht.

## 1.1. Motivation

In der heutigen Zeit sind Roboter bereits vielfach in der Industrie im Einsatz. Sie übernehmen Aufgaben in der Herstellung, Montage oder Kontrolle diverser Produkte. In kaum einer großen Industrieanlage wird heute auf den Einsatz von Robotern verzichtet. Viele Roboter sind stationär und können ihre Aufgaben nur an einem festen Ort ausführen.

„Die Motivation, warum wir uns mit der Entwicklung [...] mobiler Roboter befassen, liegt zum großen Teil in [...] dem Wunsch, Roboter einsetzen zu können, die mit und für Menschen bei ihrer gewöhnlichen Arbeit und in ihrer alltäglichen Umgebung arbeiten“ (Nehmzow 2002, S. 1). So könnten sie etwa als Helfer Arbeiten in öffentlichen Einrichtungen, wie z.B. Krankenhäusern, Ämtern usw., in technischen Betrieben oder im privaten Alltag übernehmen.

Einige mobile Roboter haben ihre Arbeit schon aufgenommen. Fahrerlose Transportsysteme (FTS) befördern bereits unterschiedlichste Materialien in vielen Produktionsbetrieben (siehe Abb. 1.1). Sie bieten dem Betreiber dabei eine Reihe von Vorteilen gegenüber dem Einsatz des manuellen Transportes. Die hohe Verfügbarkeit und Zuverlässigkeit tragen maßgeblich dazu bei, die Produktion zu steigern und dabei Kosten zu senken. Besonders in Umgebungen, die für den Menschen sehr gefährlich oder gar nicht zugänglich sind, könnten autonome mobile Roboter in Zukunft vermehrt eingesetzt werden.

„Eine Grundvoraussetzung für das autonome Agieren mobiler Roboter ist die Bewältigung der sicheren Navigation“ (Stegmann und Liu 2007, S. 82). Die Genauigkeit der Orts- und Lagebestimmung ist dabei wesentlich für den Grad seiner Autonomie (vgl. Blume 2008, S. 66). Im Indoor-Bereich gibt es bereits eine Vielzahl an Navigationsverfahren. Dazu zählt z.B. die sog. Lasernavigation. Hierbei wird, wie in der Seefahrt früher üblich, über das Prinzip der Kreuzpeilung die eigene Position bestimmt. Andere gängige Verfahren sind die Positionsbestimmung mit Ultraschall, Radio- und Microwellen (vgl. Manske 2008b, S.2-4).



Abbildung 1.1.: fahrerloses Transportsystem „carobot“ der Firma psb ([psb FTS carobot 2007](#))

Eine sehr vielversprechende Technik zur Indoor-Positionsbestimmung ist die Verwendung von Ultra Wide Band (UWB) Frequenzen. Der mobile Roboter kann dabei sein Signal an mehrere Sender im Raum aussenden und über die Laufzeit und Stärke des Antwortsignals seine aktuelle Position genau bestimmen (vgl. [Laaraiedh u. a. 2009](#), S. 53). Es gibt jedoch viele Umgebungen, die aufgrund ihrer Beschaffenheit, den Einsatz dieser Navigationsverfahren schwierig gestalten oder gar nicht erst möglich machen.

Aus diesen Gründen könnten Roboter im Indoor Bereich mit Kameras ausgestattet werden, um anhand der gelieferten Kamerabilder im Raum zu navigieren. Kameras haben den Vorteil, dass sie dem Roboter ein komplettes „Sichtfeld“ ermöglichen.

In einem der Laborräume der Hochschule für Angewandte Wissenschaften (HAW) Hamburg, in dem ein Testroboter für visuelle Navigation betrieben wird (siehe [Abb. 1.2](#)), werden zur Positionsbestimmung künstliche Marken eingesetzt, die fest an der Wand angebracht und deren Positionen genau bekannt sind (siehe [Abb. 1.3](#)). Die Marken dienen dem mobilen Roboter als Bezugspunkte und ermöglichen es ihm seine genaue Position zu berechnen (siehe ([Manske 2008a](#))). Um mit diesen Marken nicht nur in eine Richtung, sondern in einem kompletten Raum navigieren zu können, bedarf es einer großen Anzahl von Marken, die über den gesamten Raum verteilt werden müssen. Für den alltäglichen Einsatz in Gebäuden ist dieser Ansatz somit eher nicht geeignet. Es muss also eine Möglichkeit gefunden werden, sich an anderen Objekten in einem Raum zu orientieren.

In dieser Arbeit soll gezeigt werden, dass eine Möglichkeit besteht anstatt mit künstlichen

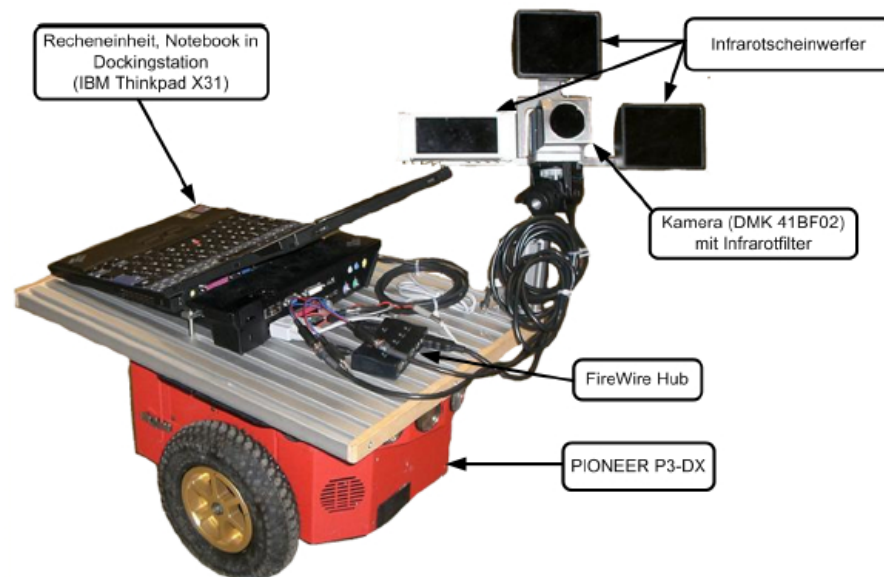


Abbildung 1.2.: Testroboter an der HAW Hamburg für die visuelle Navigation ([Manske 2008a](#), S. 60)

Marken mittels natürlicher Merkmale zu navigieren. Bekannte Verfahren zum Auffinden natürlicher Merkmale sind die Eckendetektoren von Harris und Stephens (vgl. [Harris und Stephens 1988](#)) oder der SUSAN Eckendetektor von Smith (vgl. [Smith 1995](#)). Mit ihnen ist es möglich z.B. Türecke in Kamerabildern zu identifizieren.

Das Scale Invariant Feature Transform (SIFT) Verfahren, welches von David Lowe entwickelt wurde (vgl. [Lowe 2004](#), S. 2), erkennt eindeutige Merkmale in Kamerabildern, ohne sich dabei auf die Eckendetektion zu beschränken. Es scheint daher, für das Vorhaben der visuellen Navigation anhand natürlicher Merkmale, besonders geeignet.

## 1.2. Ziel der Arbeit

Anknüpfend an vorangegangene Entwicklungen und Implementationen des SIFT Verfahrens, soll in dieser Arbeit die Voraussetzung geschaffen werden, einen mobilen Roboter anhand von SIFT-Merkmalen im Raum navigieren zu können. Für dieses Vorhaben werden zwei Module benötigt:

### 1. Einrichtungsmodul

In diesem Modul müssen die Voraussetzungen dafür geschaffen werden, in einem Raum mittels SIFT-Merkmalen zu navigieren.

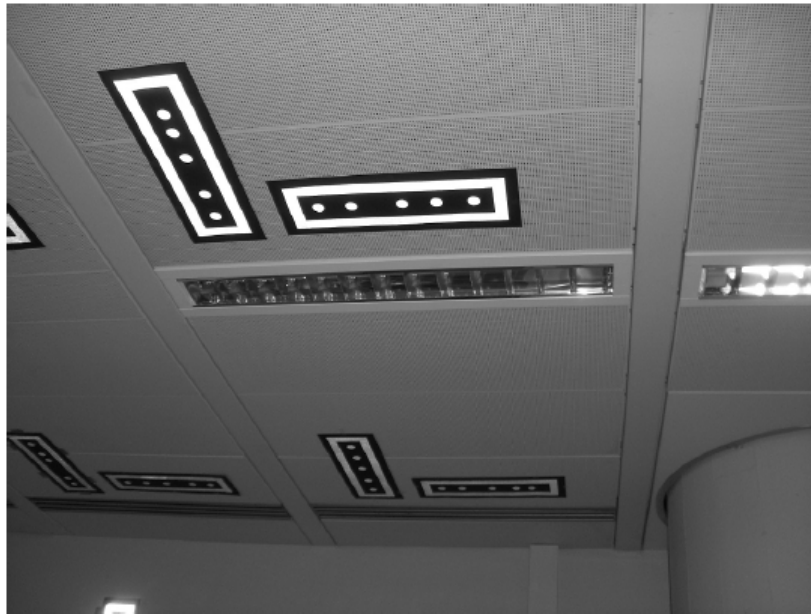


Abbildung 1.3.: künstliche Positionsmarken ([Manske 2008a](#), S. 44)

## 2. Navigationsmodul

Dieses Modul erweitert ein bereits bestehendes System zur visuellen Navigation. Es soll die, als Bezugspunkt dienenden, künstlichen Marken durch SIFT-Merkmale ersetzen.

Diese beiden Module sollen im Rahmen dieser Bachelorarbeit entwickelt und auf ihre Genauigkeit analysiert werden.

## 1.3. Struktur der Arbeit

Die Arbeit gliedert sich in 3 Hauptteile, welche sich mit den allgemeinen Grundlagen, dem Einrichtungsmodul und dem Navigationsmodul befassen.

In Kapitel 2 werden die mathematischen und allgemeinen Grundlagen der Arbeit, wie

- SIFT
- Koordinatensysteme
- Kameraparameter
- homogene Koordinaten

- Abbildung eines Raumpunktes

kurz erläutert.

Das Kapitel 3 beschreibt das Einrichtungsmodul und geht dabei auch auf

- Rahmenbedingungen
- Genauigkeit
- Verbesserungen

ein.

Im Kapitel 4 wird das Navigationsmodul näher gebracht, welches sich mit

- Realisierung
- Genauigkeitsanalyse

befasst.

Am Ende wird die Arbeit noch einmal zusammengefasst sowie ein Ausblick auf mögliche weitere Entwicklungen gegeben.

## 2. Grundlagen

In diesem Kapitel werden fundamentale Verfahren beschrieben, auf die in dieser Arbeit zurückgegriffen wird.

### 2.1. SIFT-Algorithmus

Um dem Vorsatz einer sicheren Navigation im Raum ohne künstliche Landmarken gerecht zu werden, muss zunächst sichergestellt sein, dass eine Möglichkeit existiert, markante Punkte in Bildern wiederzufinden. Das Scale Invariant Feature Transform (SIFT) Verfahren, welches von David G. Lowe 1999 erstmals erwähnt und 2004 erweitert wurde, verspricht die Anforderungen zu erfüllen (vgl. [Lowe 2004](#)).

Die, mit dem SIFT Algorithmus, gefundenen markanten Punkte, sind invariant gegenüber Skalierung, Rotation und teilweise auch gegenüber unterschiedlicher Beleuchtungen und Perspektiven. Ein SIFT-Punkt wird durch einen 128-dimensionalen Merkmalsvektor beschrieben.

Zur Ermittlung dieser SIFT-Punkte werden potentielle SIFT-Kandidaten im Bild mit Hilfe von Gauß- und Laplace-Pyramiden erkannt und mit einem Nachbarschaftsvergleich genau bestimmt. Für einen erkannten SIFT-Punkt wird im Anschluss die Position und Orientierung berechnet und der Merkmalsvektor erzeugt (vgl. [Lange 2008](#), S. 14-25).

Durch einen Vergleich der Merkmalsvektoren, mit Hilfe der euklidischen Distanz, ist es möglich, gleiche SIFT-Punkte in unterschiedlichen Bildern wiederzufinden.

Der SIFT-Algorithmus findet beispielsweise Anwendung bei der Erstellung von Panoramabildern aus mehreren Fotos. Hierbei werden die Bilder auf gleiche SIFT-Merkmale am Bildrand untersucht und an diesen Stellen zusammengefügt (siehe [Abb. 2.1](#)).



Abbildung 2.1.: Panoramabild durch SIFT-Merkmale (Brown und Lowe 2003)

## 2.2. Koordinatensysteme

In dieser Arbeit wird zwischen verschiedenen Koordinatensystemen unterschieden (siehe Abb. 2.2):

- Weltkoordinatensystem
- Kamerakoordinatensystem (Projektionszentrumkoordinatensystem)
- Bildkoordinatensystem

Es gelten hierfür und für alle weiteren mathematischen Berechnungen folgende Formatierungen:

- Raumpunkt in Weltkoordinaten:  $\mathbf{X}_w = (X_w, Y_w, Z_w)^T$
- Raumpunkt in Kamerakoordinaten:  $\mathbf{X}_c = (X_c, Y_c, Z_c)^T$
- Bildpunkt:  $\mathbf{x}_b = (x_b, y_b)^T$
- Verzeichneter Bildpunkt:  $\mathbf{p}_d = (u_d, v_d)$
- Verzeichnungsfreier Bildpunkt:  $\mathbf{p}_c = (u_c, v_c)$
- Rotationsmatrix:  $\mathbf{R} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{pmatrix}$
- Transformationsvektor:  $\mathbf{t} = (t_x, t_y, t_z)^T$

Für die Beschreibung der Lage eines Objektes im Raum gibt es zwei Möglichkeiten, zum einen das Weltkoordinatensystem  $\mathbf{X}_w = (X_w, Y_w, Z_w)^T$  und zum anderen das Kamerakoordinatensystem  $\mathbf{X}_c = (X_c, Y_c, Z_c)^T$  (vgl. Jähne 1997, S. 190).



Die Weltkoordinaten haben einen Ursprung, der an einem beliebigen Ort im Raum gesetzt wird. Ein in Weltkoordinaten gegebener Punkt  $\mathbf{X}_w$  wird mit  $\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{t})$  in die Koordinaten  $\mathbf{X}_c$  des Kamerakoordinatensystems umgerechnet. Der Translationsvektor  $\mathbf{t} = (t_x, t_y, t_z)^T$  beschreibt die Verschiebung des Weltkoordinatenursprungs in das Zentrum der Kamera. Mit Hilfe der Rotationsmatrix  $\mathbf{R}$  wird das Weltkoordinatensystem im Zentrum der Kamera richtig orientiert.

Bei der Aufnahme eines Bildes werden nun die Koordinaten des dreidimensionalen Raumes auf einer zweidimensionalen Bildebene abgebildet. Diese Bildebene wird durch das Bildkoordinatensystem  $\mathbf{x}_b = (x_b, y_b)^T$  beschrieben. Damit man bei der digitalen Bildverarbeitung das Problem der negativen Koordinaten umgeht, wird der Ursprung in die linke obere Bildecke gelegt und nur positive Koordinaten zugelassen.

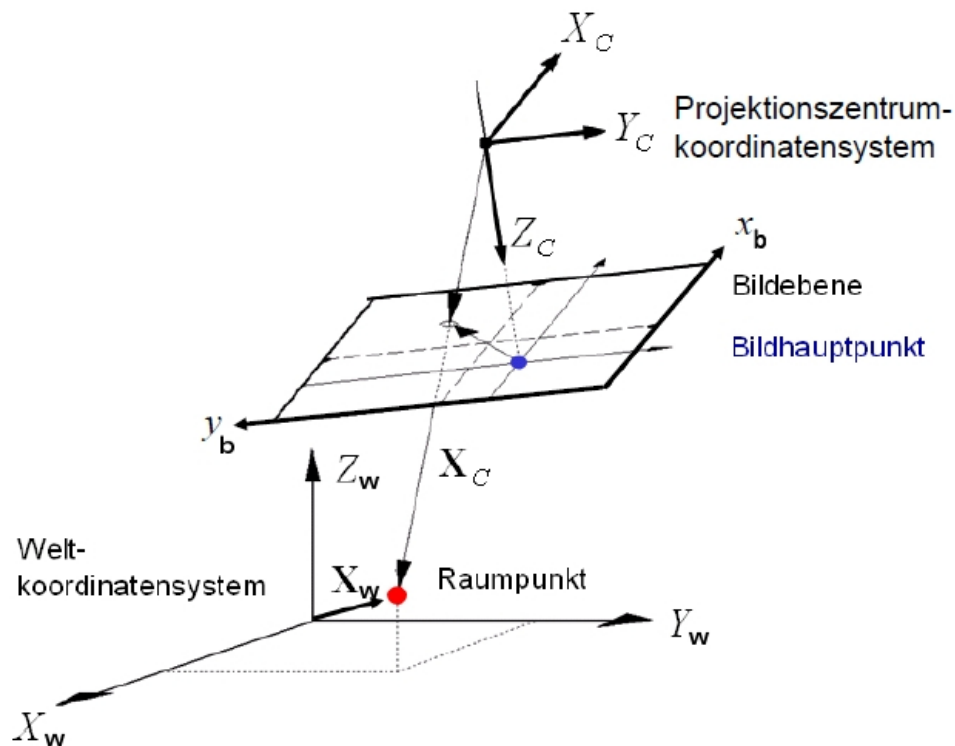


Abbildung 2.2.: Zusammenhang der unterschiedlichen Koordinatensysteme (vgl. [Meisel 2008](#), S. 23)

## 2.3. Kameraparameter

Die Kameraparameter lassen sich in zwei Gruppen aufteilen:

- extrinsische Kameraparameter
- intrinsische Kameraparameter

Die extrinsischen Kameraparameter geben die Lage  $(X_w, Y_w, Z_w, \omega, \phi, \kappa)$  der beweglichen Kamera in Weltkoordinaten an. Die Bestimmung dieser Werte ist zum einen notwendig für die Berechnung der 3D-Koordinaten von Punkten in Bildern, später zum anderen für die Position der Kamera (auf dem mobilen Roboter) bei der Navigation.

Intrinsische Kameraparameter geben die inneren Werte der Kamera an. Hierzu gehören die optischen, geometrischen und digitalen Eigenschaften der Kamera.

Würde man das einfachste Kameramodell, die Lochkamera, verwenden, würde man eine ideale Abbildung des aufgenommenen Bildes erhalten. Jeder Punkt des aufgenommenen Objektes würde linear auf der Bildebene abgebildet werden. In der Realität gibt es jedoch keine einsetzbare Lochkamera und somit auch keine ideale Abbildung. Man ist an eine Optik und die damit verbundenen Eigenschaften gebunden. Die internen Gegebenheiten der Kamera (mit Optik) werden durch folgende Parameter beschrieben:

- Bildweite ( $f$ )
- Größe des Sensors
- Anzahl der Bildpunkte des Sensors
- Bildhauptpunktverschiebung  $(p_x, p_y)$
- Linsenverzeichnungskoeffizienten  $(K_1, K_2, K_3, P_1, P_2)$

Unter der Bildweite versteht man den Abstand der Optik zur Sensorfläche.

### 2.3.1. Bildhauptpunktverschiebung

Der Bildhauptpunkt ist die Position, an der die optische Achse auf den Bildsensor trifft. Idealerweise ist der Mittelpunkt des Sensors der Kamera genau mittig zum Bildhauptpunkt. Da dies, aus bautechnischen Gründen, aber so gut wie nie der Fall ist, findet eine Verschiebung des Bildhauptpunktes um  $(p_x, p_y)$  statt (siehe Abb. 2.3).

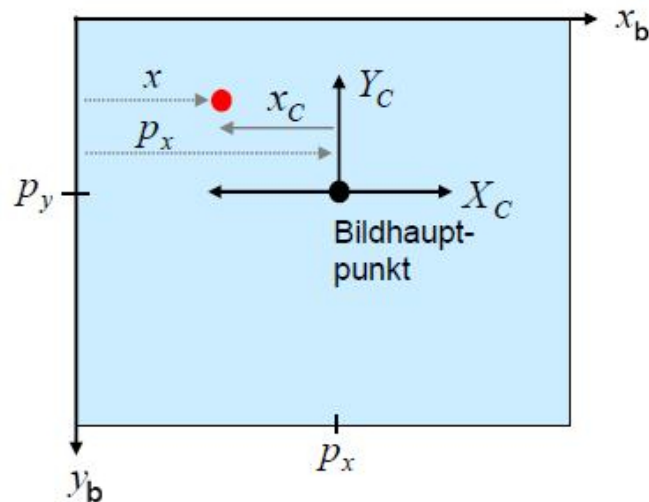


Abbildung 2.3.: Bildhauptpunktverschiebung (vgl. Meisel 2008, S. 21)

### 2.3.2. Linsenverzeichnung

Die Linsenverzeichnung beschreibt die Verzerrung des aufgenommenen Bildes (siehe Abb. 2.4). Man unterscheidet hierbei zwischen

- radialer Linsenverzeichnung
- tangentialer Linsenverzeichnung

Die radiale Linsenverzeichnung entsteht durch die Ablenkung und Brechung der Lichtstrahlen durch die der Kamera vorgesetzten Optik. Das Ergebnis sind kissen- oder tonnenförmige Verzerrungen des Bildes. Bei weitwinkligen Objektiven ist die radiale Linsenverzeichnung oftmals schon mit bloßem Auge zu erkennen.

Die tangentiale Linsenverzeichnung entsteht dadurch, dass der Sensor nicht genau orthogonal zur optischen Achse liegt. Das entstandene Bild ist geschert. Für eine genaue Rückvermessung der Bildpunkte müssen in den Bildern mit Hilfe der Linsenverzeichnungskoeffizienten diese Verzerrungen rausgerechnet werden. Folgende Gleichungen beschreiben die Korrektur eines verzeichneten Bildpunktes  $\mathbf{p}_d(u_d, v_d)$  in einen verzeichnungsfreien Punkt  $\mathbf{p}_c(u_c, v_c)$  (vgl. Hartley und Zisserman 2003, S. 189-193).

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

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

mit

$$r = \sqrt{u_d^2 + v_d^2}$$

Durch die Korrektur der Linsenverzeichnung lässt sich die Genauigkeit der Raumpunktvermessung deutlich erhöhen.

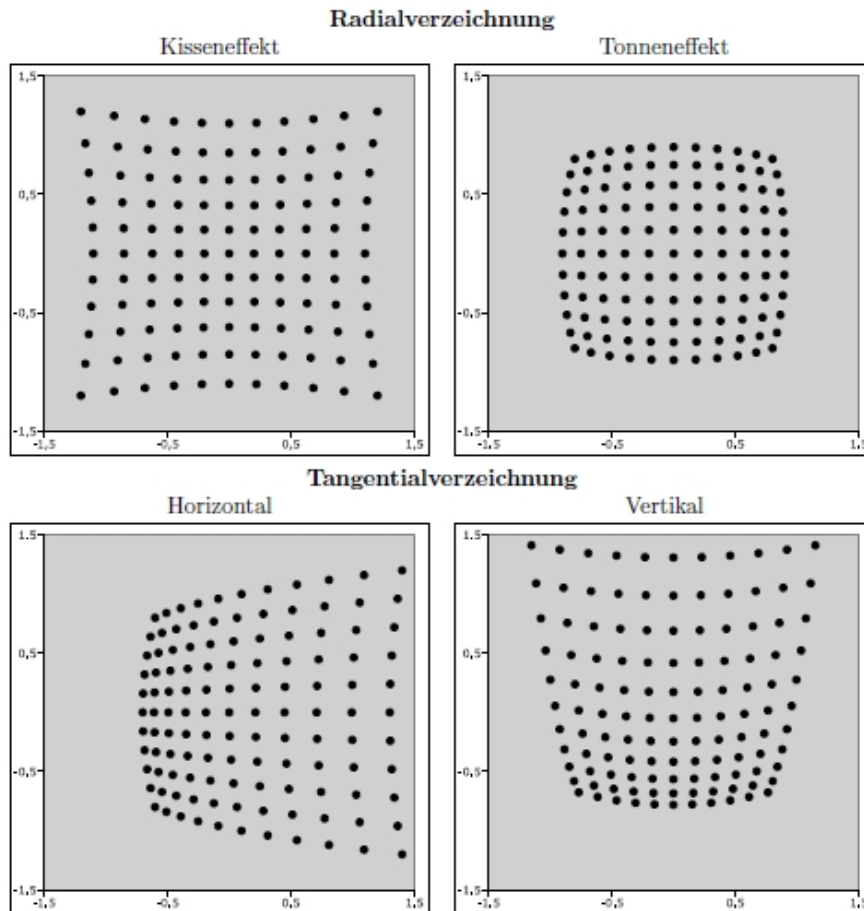


Abbildung 2.4.: Linsenverzeichnungseffekte (vgl. [Eschenburg 2006](#), S. 25)

## 2.4. Homogene Koordinaten

In der digitalen Bildverarbeitung werden zur Berechnung von Transformationen (Translation, Rotation, Skalierung, Scherung) sehr häufig homogene Koordinaten eingesetzt, da hierdurch die projektive Abbildung (ohne Linsenverzeichnung) durch eine Folge von Matrizenmultiplikationen beschrieben werden kann.

Im Allgemeinen gilt, dass ein  $n$ -dimensionaler Punkt in homogenen Koordinaten durch einen  $(n+1)$ -dimensionalen Vektor beschrieben wird. Man erweitert also den Koordinatenvektor um eine Dimension. Ein dreidimensionaler Raumpunkt würde durch folgende homogene Koordinaten beschrieben werden:

$$\mathbf{X}_w = \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Mit diesem eleganten Formalismus ist es möglich, jede Art von Transformation auf eine Multiplikation der homogenen Koordinaten mit einer Transformationsmatrix abzubilden. Die Berechnung der Rotation eines dreidimensionalen Raumpunktes könnte somit durch folgende Gleichung beschrieben werden:

$$\mathbf{X}_w = \mathbf{R} \cdot \mathbf{X}_w$$

Aufgrund der Assoziativität der Matrizenmultiplikation kann die Transformationsmatrix aus vielen unterschiedlichen Transformationsmatrizen zusammengesetzt werden.

Um wieder die euklidischen Koordinaten zu erhalten, muss jede Koordinate durch die hinzugefügte Dimensionskoordinate dividiert werden.

## 2.5. Abbildung eines Raumpunktes

Wird mit einer Kamera eine Aufnahme erstellt, so wird ein dreidimensionaler Raumpunkt in Kamerakoordinaten  $\mathbf{X}_c = (X_c, Y_c, Z_c)$  auf einem zweidimensionalen Bildpunkt  $\mathbf{x}_b = (x_b, y_b)$  abgebildet. Um die mathematischen Zusammenhänge der Abbildung zu beschreiben, geht man von dem einfachsten Kameramodel, der Lochkamera, aus. Alle Lichtstrahlen gelangen durch die kleine Lochblende linear auf die Bildebene (siehe Abb. 2.5). Die Position des abgebildeten Punktes  $\mathbf{x}_b$  lässt sich anhand des Strahlensatzes wie folgt bestimmen:

$$\mathbf{x}_b = (x_b, y_b) \left( f \cdot \frac{X_c}{Z_c}, f \cdot \frac{Y_c}{Z_c} \right)$$

Berücksichtigt man nun auch die Bildhauptpunktverschiebung um  $p_x, p_y$  so erhält man:

$$\mathbf{x}_b = (x_b, y_b) \left( f \cdot \frac{X_c}{Z_c} + p_x, -f \cdot \frac{Y_c}{Z_c} + p_y \right)$$

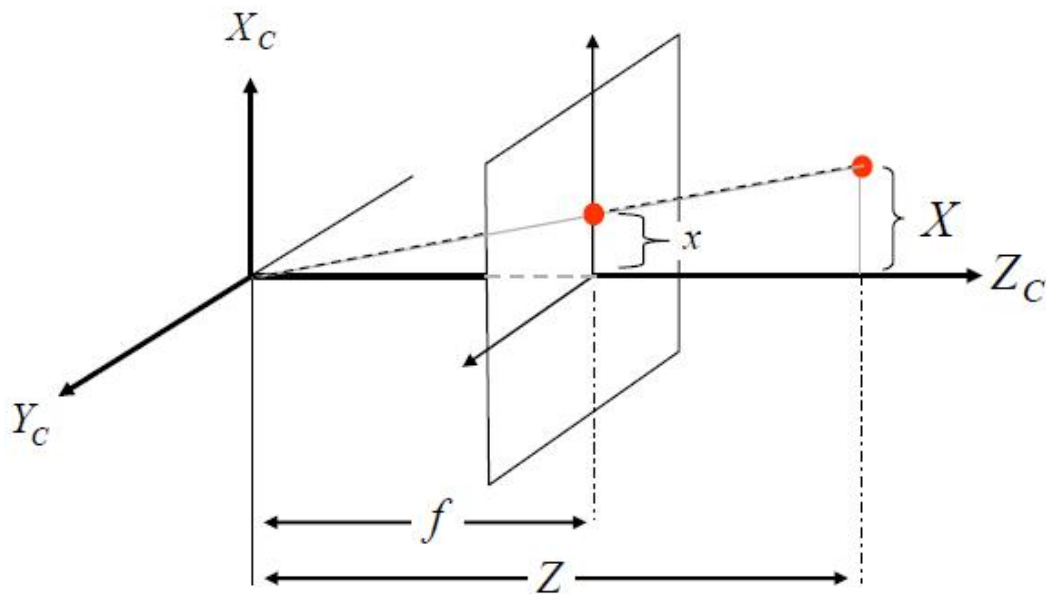


Abbildung 2.5.: Abbildung eines Raumpunktes (Meisel 2008, S. 19)

In homogenen Koordinaten kann man die Berechnung auf eine Transformationsmatrix zusammenfassen:

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} \sim \begin{pmatrix} +f & 0 & px & 0 \\ 0 & -f & py & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

In den meisten Fällen sind die Koordinaten des Raumpunktes jedoch nicht in Kamerakoordinaten sondern in Weltkoordinaten gegeben. Aus diesem Grund ist eine Transformation des Weltkoordinatensystems in das Zentrum der Kamera nötig. Dabei werden, wie oben beschrieben, mit  $\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{t})$  die Koordinaten umgerechnet.  $\mathbf{R}$  ist dabei wie folgt definiert:

$$\begin{pmatrix} \cos \phi \cos \kappa & \cos \omega \sin \kappa + \sin \omega \sin \phi \cos \kappa & \sin \omega \sin \kappa - \cos \omega \sin \phi \cos \kappa \\ -\cos \phi \sin \kappa & \cos \omega \cos \kappa - \sin \omega \sin \phi \sin \kappa & \sin \omega \cos \kappa + \cos \omega \sin \phi \sin \kappa \\ \sin \phi & -\sin \omega \cos \phi & \cos \omega \cos \phi \end{pmatrix}$$

Zur Berechnung der Bildkoordinaten des abgebildeten Raumpunktes unter Berücksichtigung der Transformation gilt:

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} \sim \begin{pmatrix} -f & 0 & px & 0 \\ 0 & +f & py & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{R} & -\mathbf{R} \cdot \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Fasst man beide Matrizen zusammen, so erhält man die Projektionsmatrix

$$\mathbf{P} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{pmatrix}$$

und es gilt:

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Um die homogenen Koordinaten wieder in euklidische Koordinaten umzuwandeln, muss, wie oben beschrieben, durch die hinzugefügte Koordinate dividiert werden. Die Bildkoordinaten des abgebildeten Raumpunktes lassen sich also wie folgt berechnen:

$$x_b = \frac{h_{11} \cdot X_w + h_{12} \cdot Y_w + h_{13} \cdot Z_w + h_{14}}{h_{31} \cdot X_w + h_{32} \cdot Y_w + h_{33} \cdot Z_w + h_{34}}$$

$$y_b = \frac{h_{21} \cdot X_w + h_{22} \cdot Y_w + h_{23} \cdot Z_w + h_{24}}{h_{31} \cdot X_w + h_{32} \cdot Y_w + h_{33} \cdot Z_w + h_{34}}$$

## 2.6. Verwendete Hard- und Software

### 2.6.1. Kamera

Die Bilder wurden mit der Monochrom-Kamera DMK 41BF02 von der Firma „The Imaging Source“ erstellt. Als Objektiv diente das H0514-MP der Firma „computar“. In der Tabelle 2.1 sind die technischen Daten der Kamera und des Objektivs aufgeführt.



Abbildung 2.6.: Die Kamera DMK 41BF02 mit Objektiv

Auflösung	1280 x 960 Pixel
Sensortyp	1/2 " Sony CCD
Sensorgroße	7,6 x 6,2 mm
Brennweite Objektiv	5 mm
Anschluss	Firewire

Tabelle 2.1.: Kameradaten der DMK 41BF02

## 2.6.2. Laborrechner

Im Labor stand für diese Arbeit ein Intel Core 2 Quad Q6600 mit 2,4 GHz zur Verfügung. Als Betriebssystem wurde Windows XP mit dem Service Pack 2 eingesetzt. Die gesamte Software wurde auf diesem Rechner entwickelt und getestet.

## 2.6.3. Photomodeler

Die Software „Photomodeler“ der Firma Eos Systems Inc. ist ein Tool zur Erstellung professioneller 3D-Modelle aus Bilderserien ([Systems 2009](#)). Zudem ist es mit dem Photomodeler möglich, die extrinsische und intrinsische Kameraparameter berechnen zu lassen und diese zu extrahieren. In dieser Arbeit wurde der Photomodeler in der Version 6 genutzt.

## 2.6.4. LTI-Lib

Die LTI-Lib ist eine freie Softwarebibliothek, die Algorithmen und Funktionen für die Bereiche Bildverarbeitung und Computer Vision beinhaltet. Die vom Lehrstuhl für Technische Informatik (LTI) an der RWTH Aachen Universität entwickelte LTI-Lib macht sich zum Ziel, eine



gemeinsam nutzbare, objektorientierte C++ Bibliothek zur Verfügung zu stellen, deren Algorithmen auch Echtzeitanforderungen genügt (vgl. [LTI-Lib 2005](#)). Getestet wurde die LTI-Lib unter Linux (GCC) und unter Microsoft Windows (Visual C++).

### **2.6.5. Impresario**

Impresario ist ein Programm, mit dem es möglich ist, Prozessabläufe für Bild und Videomaterial, schnell und dynamisch zu erstellen (vgl. [Duif 2006](#), S. 2). Hierzu werden einzelne Module über eine grafische Oberfläche miteinander verknüpft. Die Module beinhalten Bildbearbeitungsalgorithmen, welche meist aus der LTI-Lib stammen. Es werden aber auch eigene Module mit eigenen Algorithmen unterstützt.

### **2.6.6. Visual Studio**

Für die Programmierung der eigenen Software wurde die Sprache C++ und die Entwicklungsumgebung Visual Studio 2003 von Microsoft verwendet. Die Hauptentscheidung für diese Wahl liegt in der Kompatibilität. Arbeiten, auf die diese aufbaut, wurden ebenfalls auf der oben genannten Umgebung bzw. der oben genannten Sprache entwickelt. Zudem konnte somit auf die Funktionen der LTI-Lib zurückgegriffen werden.

## 3. Einrichtungsmodul

In diesem Kapitel wird beschrieben wie ein Raum erfasst wird, um später in diesem zu navigieren.

### 3.1. Einführung

Damit in einem Raum visuell navigiert werden kann, bedarf es einer ausreichenden Zahl von wiederauffindbaren, stabilen und eindeutigen Punkten anhand derer man sich orientieren kann. Bis jetzt übernehmen künstliche Landmarken diese Aufgabe. Sie werden an der Wand montiert und genau vermessen. Die Positionen werden, zusammen mit der Kodierung der Landmarke, dem Fahrzeug bereit gestellt. Das Fahrzeug erkennt während der Fahrt die Marken an der Wand, kennt ihre genauen Positionen im Raum und kann somit die genaue Fahrzeugposition berechnen.

Die SIFT-Merkmale sollen nun die Rolle der künstlichen Landmarken einnehmen und dem Fahrzeug als Orientierungspunkte dienen. Um dies zu ermöglichen, sind mehrere Schritte notwendig, welche sich in manuelle und automatisierte Abläufe aufteilen.

### 3.2. Testumgebung

Als Testumgebung wurde ein Teil des Flures im 7ten Stock der HAW Hamburg ausgewählt (siehe Abb. [3.1](#)). Die Umgebung stellte aufgrund ihrer geringen Breite und großen Tiefe eine Art „worst case Scenario“ dar. Zudem liefern die weißen Wände keine ideale Struktur, um gute SIFT-Punkte zu ermitteln. Dennoch handelt es sich bei dem gewählten Raum um eine realistische Umgebung, da solche Räume in vielen potentiellen Anwendungsgebieten für visuelle Navigation zu finden sind. Genauere Ausführungen zu den Eigenschaften und Problemen der Testumgebung finden sich im Abschnitt [3.5](#)



Abbildung 3.1.: Die Testumgebung

### 3.3. Bestimmung der Bildorientierung

Zur Bestimmung der Bildorientierung sind unterschiedliche Schritte nötig, die alle manuell durchgeführt werden müssen:

1. Festlegung des Weltkoordinatensystems mit Hilfe eines Raummodells
2. Ermittlung der intrinsischen Kameraparameter
3. Berechnung der Kameraposition

Im Diagramm [3.2](#) wird der Ablauf schematisch dargestellt.

#### 3.3.1. Raummodell als Weltkoordinatensystem

Um die Position von SIFT-Merkmalen berechnen zu können, muss zunächst ein Weltkoordinatensystem in dem zu navigierenden Raum festgelegt werden. Für diesen Zweck wurde die Testumgebung von Hand vermessen und der Ursprung der Weltkoordinaten in die linke untere Flurecke gesetzt. Mit dem Photomodeler ist es möglich, in einem Bild der gesamten Testumgebung, alle gemessenen Strecken zu markieren und diesen eine feste Länge zuzuweisen. Somit kann ein 3D-Raummodell erstellt werden, welches in den Abmessungen exakt der realen Umgebung entspricht.

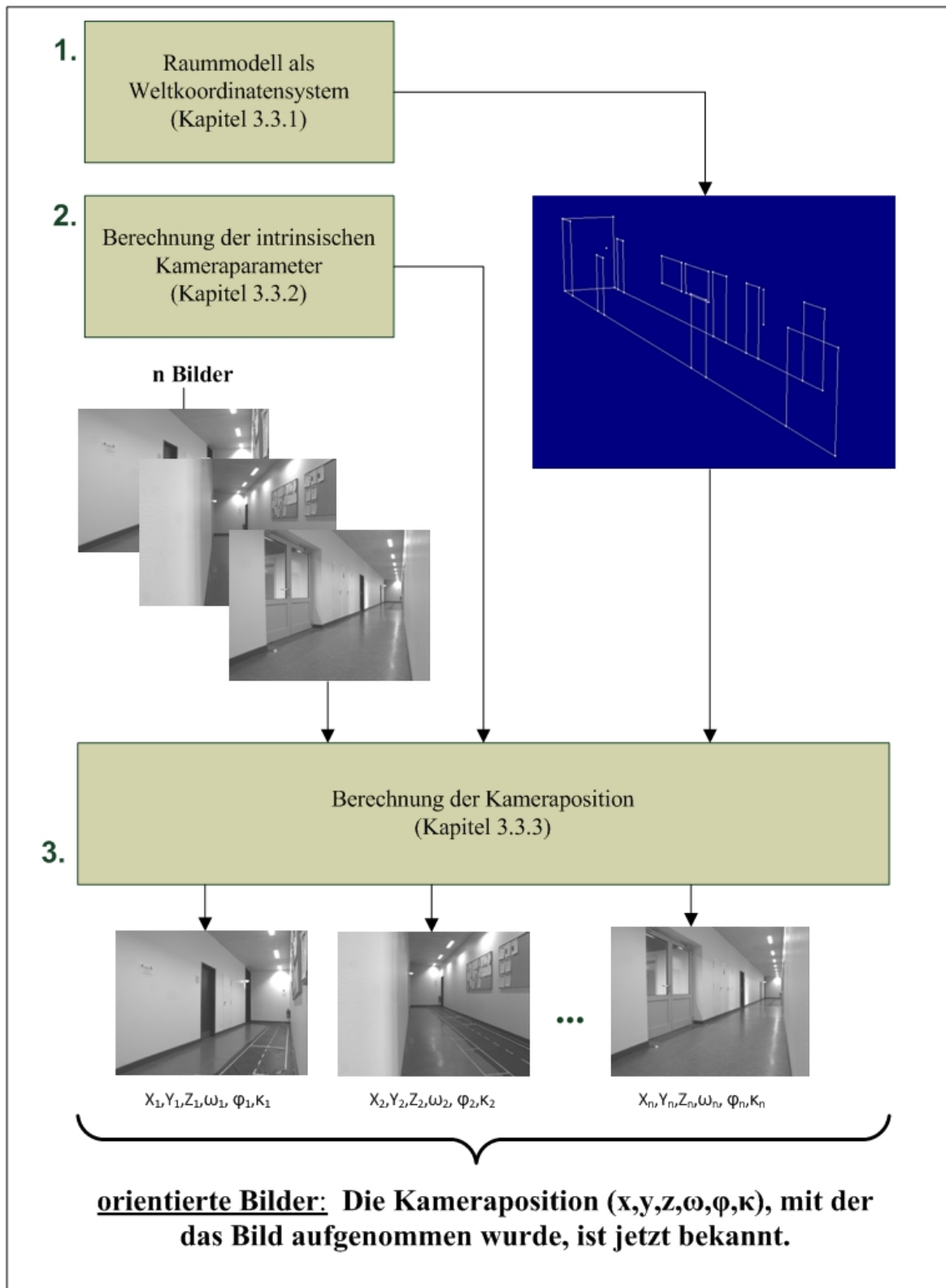


Abbildung 3.2.: Ablauf der manuellen Bildorientierung

### 3.3.2. Berechnung der intrinsischen Kameraparameter

Nachdem somit die Voraussetzung für eine Positionsbestimmung im Raum geschaffen wurde, müssen jetzt die Eigenschaften der Kamera bestimmt werden. Die intrinsischen Kameraparameter sind, da für alle Aufnahmen die gleiche Kamera eingesetzt wurde, in allen Bildern identisch. Um sie zu bestimmen, bietet der Photomodeler eine Kalibrierungsfunktion. Die mitgelieferte Kalibrierplatte (siehe Abb. 3.3) muss von jeder Seite einmal im Hoch- und Querformat fotografiert werden. Aus diesen acht Bildern ermittelt der Photomodeler dann automatisch die kompletten intrinsischen Kameraparameter.

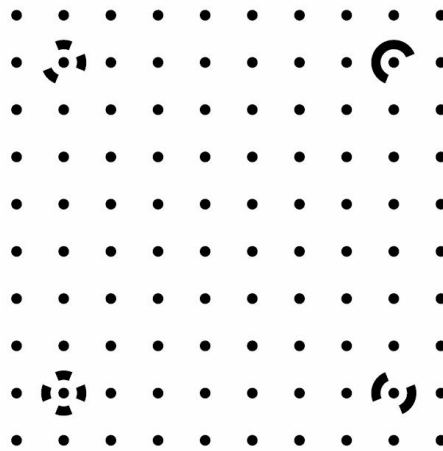


Abbildung 3.3.: Kalibrierplatte vom Photomodeler

### 3.3.3. Berechnung der Kameraposition

Die extrinsischen Kameraparameter, also die Positionen der Aufnahmekamera, sind für jedes Bild der Serie zu ermitteln. Das durch das 3D-Raummodell festgelegte Weltkoordinatensystem muss dafür in jedes Bild „eingepasst“ werden. Dafür gibt es im Photomodeler die Möglichkeit, Bilder mit dem 3D-Raummodell zu verknüpfen. In jedem Bild werden dafür möglichst viele anmessbare Punkte des 3D-Modells referenziert (siehe Abb. 3.4). Der Photomodeler errechnet, anhand des 3D-Modells und der anmessbaren Punkte im Bild, die Kameraposition in Weltkoordinaten. Nachdem für jedes Bild der Serie auf diese Art und Weise alle Kamerapositionen berechnet wurden, werden sie, zusammen mit den intrinsischen Kameraparametern, in eine Datei extrahiert.

Die errechneten Werte für die Lage und Position der Aufnahmekameras können im Photomodeler grafisch dargestellt werden (siehe Abb. 3.5). Diese Darstellung bietet eine gute Kontrolle für die Richtigkeit der äußeren Kameraparameter. In dem Testfall wurde je ein Foto



Abbildung 3.4.: Setzen der anmessbaren Punkte

von der äußeren linken bzw. rechten Flurseite in unterschiedlichen Entfernungen aufgenommen.

### 3.4. Raumpunktbestimmung der SIFT-Merkmale

Nachdem nun alle Bilder orientiert und alle notwendigen Kameraparameter verfügbar sind, müssen die SIFT-Merkmale, anhand derer später navigiert werden soll, gefunden und deren Raumkoordinaten bestimmt werden. Die Bestimmung der Raumkoordinaten von SIFT-Merkmalen läuft in folgenden, automatisierten Schritten ab:

1. Identifizieren von SIFT-Punkten
2. Vermessen dieser SIFT-Punkte

Im Diagramm [3.6](#) wird dieser Ablauf veranschaulicht.

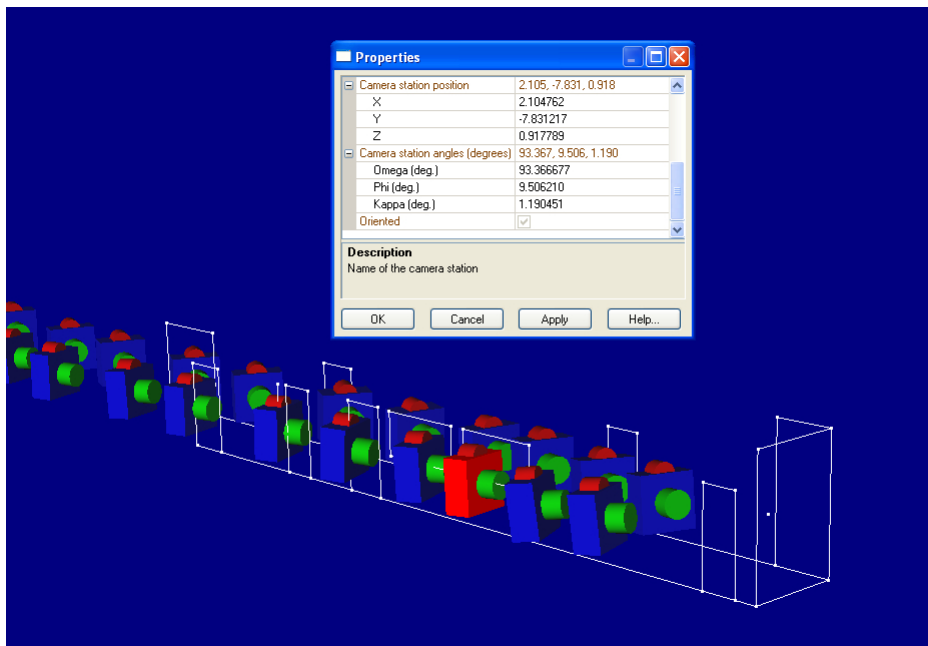


Abbildung 3.5.: Kameras im 3D-Modell

### 3.4.1. Identifikation von SIFT-Punkten

Zur Ermittlung der SIFT-Punkte in den Bildern wurde auf der Abschlussarbeit von Emmanuel Lange aufgebaut (Lange 2008). Das von Herrn Lange entwickelte Modul `SIFT++` für Impresario ist in der Lage SIFT-Punkte aus Bildern zu ermitteln (siehe Abb. 3.7). Das Modul wurde für die Zwecke dieser Arbeit leicht modifiziert und unter dem Namen `SIFT++ Extended Navi` verwendet. Die Bilder werden in den Versuchen vorher mit einem Downsampling-Modul um den Faktor 2 verkleinert. Dadurch können die SIFT-Merkmale im Bild schneller ermittelt werden. Gerade bei Kameras mit einer hohen Auflösung macht sich dieses durch eine deutlich geringere Rechenzeit bemerkbar. Die Verkleinerung des Bildes führt jedoch auch zu einer etwas geringeren Genauigkeit der SIFT-Merkmale.

Die gefundenen SIFT-Merkmale können nun im Bild angezeigt werden (Siehe Abb. 3.8).

Die Koordinaten jedes gefundenen SIFT-Merkmals werden, zusammen mit dem berechneten Merkmalsvektor, in eine Datei extrahiert. In der entwickelten C++ Software werden diese SIFT-Daten wieder eingelesen und als Objekte der Klasse `SiftPoint` in einer Liste gespeichert. Die Werte der extrahierten Kameradaten werden ebenfalls eingelesen und als `Camera` Objekt gespeichert. Weitere Ausführungen zu allen Klassenbeschreibungen finden sich im Anhang A.

Wie in den Grundlagen beschrieben, sind die SIFT-Merkmale invariant gegenüber Skalie-

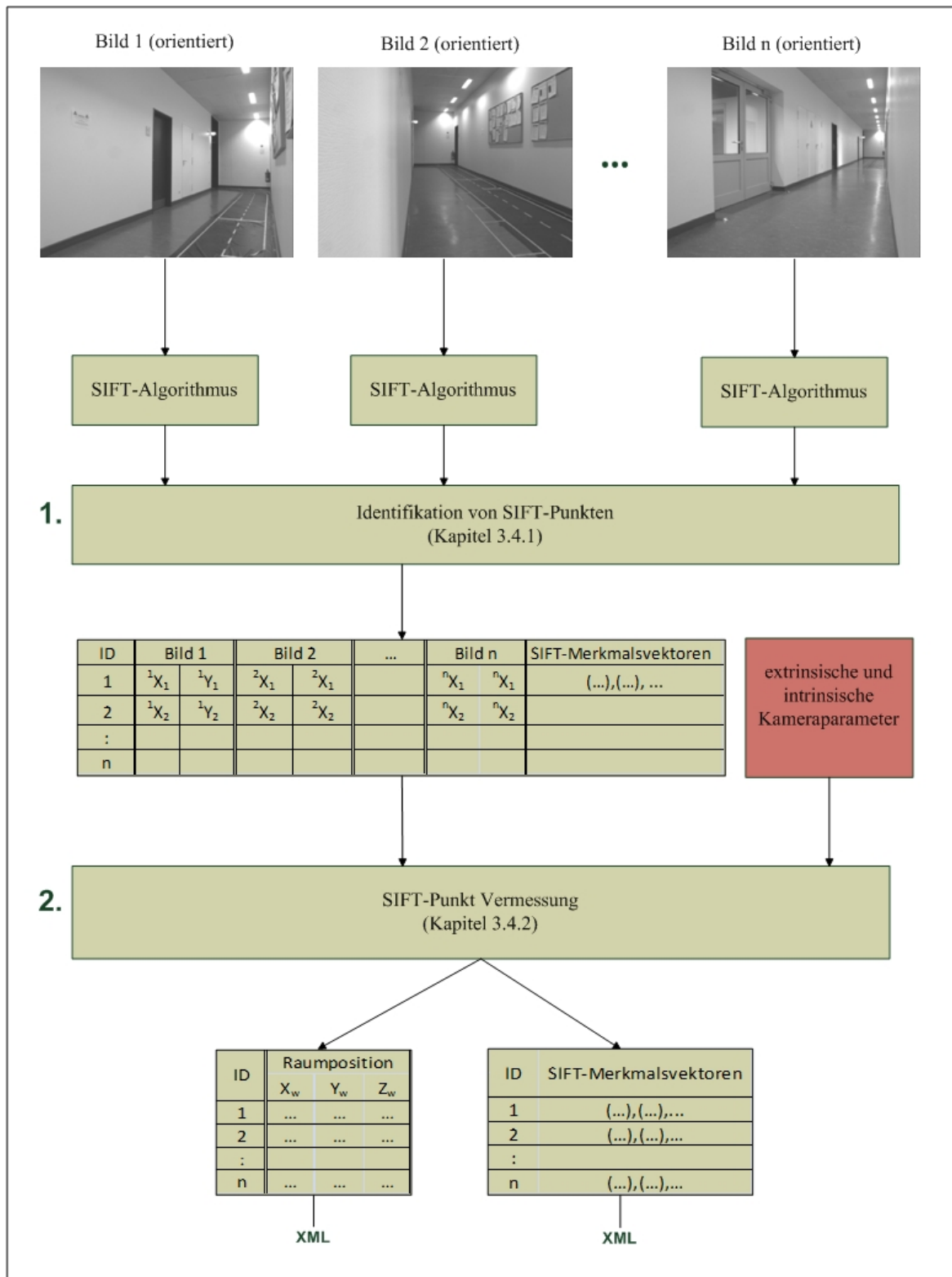


Abbildung 3.6.: Ablauf der automatisierten Raumpunktbestimmung der SIFT-Merkmale



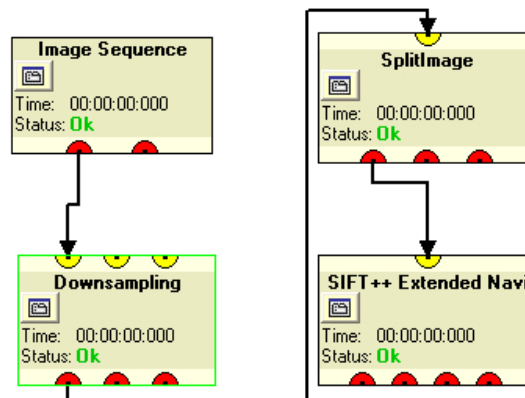


Abbildung 3.7.: Impresario-Module zur Auffindung von SIFT-Merkmalen

ung und Rotation. Dies kann zur Folge haben, dass mehrere SIFT-Merkmale an der gleichen Position im Bild erkannt wurden, alle jedoch eine unterschiedliche Rotation aufweisen. Der Merkmalsvektor dieser SIFT-Merkmale ist somit unterschiedlich, ihre Koordinaten jedoch sind identisch. Beim späteren Vergleich der Merkmalsvektoren, zur Erkennung gleicher SIFT-Merkmale in unterschiedlichen Bildern, könnte der SIFT-Punkt aus dem einen Bild in einem anderen Bild an mehreren, unterschiedlichen Positionen wiedergefunden werden. Da in diesem Projekt die Rotation der SIFT-Merkmale nicht weiter von Bedeutung ist, sondern stabile Merkmale gesucht werden, werden alle SIFT-Merkmale mit mehreren gleichen Koordinaten aber unterschiedlichem Merkmalsvektor gelöscht.

Wie bereits beschrieben bedarf es zur sicheren Navigation SIFT-Punkte, die einen hohen Wiedererkennungswert besitzen. Idealerweise sind diese in allen Bildern der Serie zu finden. Aufgrund von unterschiedlichen Abständen und Perspektiven gibt es jedoch keinen SIFT-Punkt, der dieses Kriterium erfüllt. Es finden sich aber einige Punkte, die in drei oder mehr Bildern vorkommen. Um diese zu ermitteln, werden alle Merkmalsvektoren der SIFT-Punkte, unterschiedlicher Bilder verglichen. Zum Vergleich wird die euklidische Distanz

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}$$

verwendet. Ist  $d$  kleiner als eine festgelegte Größe, werden beide SIFT-Punkte als gleich angesehen. Durch Änderung der Größe kann somit die Toleranz zur Erkennung identischer Merkmale erhöht, bzw. gesenkt werden.

Gleiche Punkte werden zu einem Objekt der Klasse `SameSiftPoint` zusammengefasst. Werden weitere SIFT-Merkmale erkannt, die den gleichen Merkmalsvektor zu diesen Punkten aufweisen, werden sie dem Objekt hinzugefügt. Können keine weiteren SIFT-Merkmale gefunden werden, wird das `SameSiftPoint`-Objekt in einer Liste gespeichert. Ausgehend von einem neuen SIFT-Merkmal werden wieder gleiche SIFT-Merkmale zusammenge-



Abbildung 3.8.: Erkannte SIFT-Merkmale im Bild

fasst. Nachdem auf diese Art und Weise mit allen SIFT-Merkmalen verfahren wurde, entsteht somit eine Liste von korrespondierenden SIFT-Punkten, die  $n$  ähnliche Merkmalsvektoren und  $n$  Bildkoordinatenpaare ( $n \geq 2$ ) aufweisen.

Für diese SIFT-Punkte gilt es nun die Raumkoordinaten zu berechnen.

### 3.4.2. SIFT-Punkt Vermessung

Die Vermessung von SIFT-Punkten basiert auf der Zentralprojektion. Ziel ist es, aus den zweidimensionalen Bildkoordinaten  $(x_b, y_b)$  der SIFT-Merkmale die dreidimensionalen Raumkoordinaten  $(X_w, Y_w, Z_w)$  zu berechnen. Da es sich bei dem Vorgang der Rückvermessung um die Umkehrung der in den Grundlagen beschriebenen Abbildung eines Raumpunktes handelt, kann man die Gleichungen

$$x_b = \frac{h_{11} \cdot X_w + h_{12} \cdot Y_w + h_{13} \cdot Z_w + h_{14}}{h_{31} \cdot X_w + h_{32} \cdot Y_w + h_{33} \cdot Z_w + h_{34}}$$

$$y_b = \frac{h_{21} \cdot X_w + h_{22} \cdot Y_w + h_{23} \cdot Z_w + h_{24}}{h_{31} \cdot X_w + h_{32} \cdot Y_w + h_{33} \cdot Z_w + h_{34}}$$

so umstellen, dass man  $(X_w, Y_w, Z_w)$  berechnen kann:

$$(x_b \cdot h_{31} - h_{11}) \cdot X_w + (x_b \cdot h_{32} - h_{12}) \cdot Y_w + (x_b \cdot h_{33} - h_{13}) \cdot Z_w = h_{14} - x_b \cdot h_{34}$$

$$(y_b \cdot h_{31} - h_{21}) \cdot X_w + (y_b \cdot h_{32} - h_{22}) \cdot Y_w + (y_b \cdot h_{33} - h_{23}) \cdot Z_w = h_{24} - y_b \cdot h_{34}$$

Die benötigten  $h_{mn}$ -Werte der Projektionsmatrix werden dafür zunächst, wie in Abschnitt 2.5 beschrieben, für jede Aufnahmekamera berechnet. Weiterhin werden, zur Steigerung der Genauigkeit, die Bildkoordinaten  $(x_b, y_b)$  in verzeichnungsfreie Punkte umgerechnet. Dies geschieht nach dem, im Abschnitt 2.3.2, beschriebenen Verfahren.

Es wird deutlich, dass dieses Gleichungssystem mit drei Unbekannten aber nur zwei Gleichungen zu keiner Lösung führt. Somit genügt nicht nur ein Bild (also ein Bildkoordinatenpaar), um die Raumkoordinaten bestimmen zu können. Es werden mindestens zwei Bilder benötigt, um  $X_w, Y_w, Z_w$  berechnen zu können. Stellt man das Gleichungssystem für  $n$  Bilder in Matrizenform dar, so erhält man:

$$\begin{bmatrix} ({}^1x_b \cdot {}^1h_{31} - {}^1h_{11}) & ({}^1x_b \cdot {}^1h_{32} - {}^1h_{12}) & ({}^1x_b \cdot {}^1h_{33} - {}^1h_{13}) \\ \vdots & \vdots & \vdots \\ ({}^nx_b \cdot {}^nh_{31} - {}^nh_{11}) & ({}^nx_b \cdot {}^nh_{32} - {}^nh_{12}) & ({}^nx_b \cdot {}^nh_{33} - {}^nh_{13}) \\ ({}^1y_b \cdot {}^1h_{31} - {}^1h_{21}) & ({}^1y_b \cdot {}^1h_{32} - {}^1h_{22}) & ({}^1y_b \cdot {}^1h_{33} - {}^1h_{23}) \\ \vdots & \vdots & \vdots \\ ({}^ny_b \cdot {}^nh_{31} - {}^nh_{21}) & ({}^ny_b \cdot {}^nh_{32} - {}^nh_{22}) & ({}^ny_b \cdot {}^nh_{33} - {}^nh_{23}) \end{bmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \begin{bmatrix} {}^1h_{14} - {}^1x_b \cdot {}^1h_{34} \\ \vdots \\ {}^nh_{14} - {}^nx_b \cdot {}^nh_{34} \\ {}^1h_{24} - {}^1y_b \cdot {}^1h_{34} \\ \vdots \\ {}^nh_{24} - {}^ny_b \cdot {}^nh_{34} \end{bmatrix}$$

Dieses Gleichungssystem ist schon bei der Nutzung von zwei Bildern (also 4 Gleichungen) überbestimmt. Das Problem lässt sich jedoch mit Hilfe einer linearen Ausgleichsrechnung lösen.

Die LTI-Lib bietet mehrere Algorithmen zur Lösung überbestimmter Gleichungssysteme. Die QR-Zerlegung mit Hilfe der Householder Transformation hat sich als sehr geeignet erwiesen und wurde in der Implementierung verwendet. So lässt sich aus den Bildkoordinaten der SIFT-Punkte in unterschiedlichen Bildern und deren Kameraparametern die Raumkoordinate des SIFT-Punktes berechnen. Die berechneten Werte werden anschließend in zwei XML Dateien extrahiert. Diese XML-Dateien sind bereits auf das folgende Navigationsmodul formatiert. Die Ergebnisse werden nun im Folgenden auf ihre Genauigkeit überprüft.

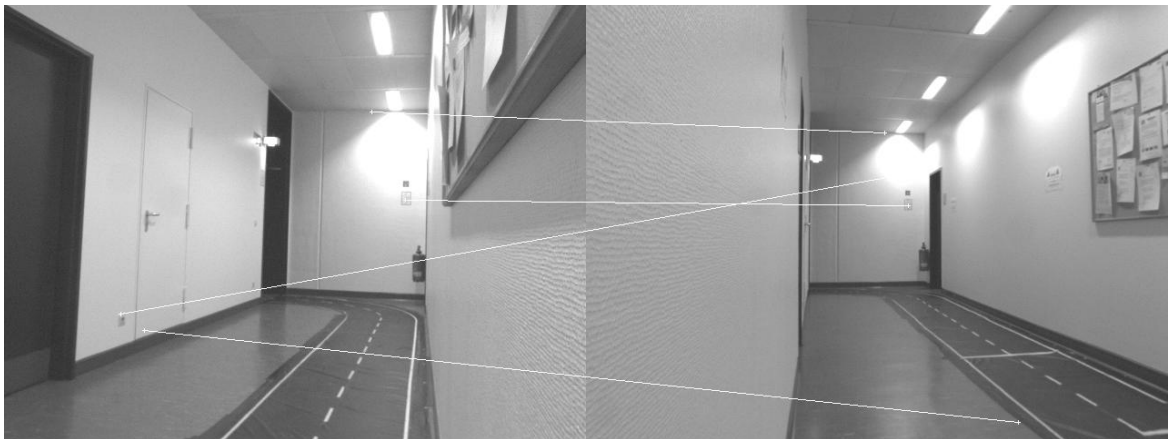


Abbildung 3.9.: Falsche Zuordnung von SIFT-Punkten

## 3.5. Genauigkeitsanalyse

Um die Genauigkeit der berechneten Raumkoordinaten der SIFT-Punkte zu analysieren, wurden aus allen 22 Bildern der Serie die SIFT-Punkte bestimmt, zusammengefasst und deren Raumkoordinaten berechnet.

### 3.5.1. Testumgebung

Bei der gewählten Messanordnung ist mit folgenden Problemen zu rechnen.

1. Die gewählte Testumgebung ist ein langer Flur, der im Verhältnis zu seiner Tiefe eine sehr geringe Breite hat. Dies führt unweigerlich zu Problemen bei der Vermessung von Positionen im Raum.
2. Ein erhebliches Problem der Testumgebung sind die weißen Wände und der Mangel an markanten Objekten. Aus diesem Grund ist es schwer eine hohe Anzahl von SIFT-Punkten zu erhalten, die sich wesentlich unterscheiden. Es besteht die Gefahr, dass beim Vergleich der Merkmalsvektoren falsche Punkte zusammengefasst werden (siehe Abb 3.9). Diese falschen Zuordnungen treten zudem auch im Zusammenhang mit Reflexionen auf dem Boden auf. Eine Möglichkeit, diese Fehler zu umgehen, ist die Toleranzgrenze beim Vergleich der Merkmalsvektoren zu erhöhen. Dies hat jedoch wiederum zur Folge, dass weniger SIFT-Punkte zusammengefasst werden können und somit am Ende die Zahl der Navigationspunkte geringer ist.

3. Gefundene SIFT-Punkte sind sehr schwer nachzumessen. Ein SIFT-Punkt auf einer weißen Wand kann vom Menschen nur sehr ungenau lokalisiert werden. Aus diesem Grund sind die Kontrollmessungen auch, je nach Position des gefundenen SIFT-Punktes, stark fehlerbehaftet.

### 3.5.2. Zuordnungssicherheit

Der erste Schritt bestand darin, zu kontrollieren ob es SIFT-Punkte gab, die falsch zusammengefasst wurden. Um diesem vorzubeugen wurden folgende Filterfunktionen eingesetzt:

- **Entfernung doppelter Koordinaten**

Wie bereits beschrieben finden sich SIFT-Merkmale mit gleichen Koordinaten und unterschiedlichen Rotationen. Diese wurden mit einem Filter entfernt.

- **Zuordnungsfehler in Abhängigkeit der euklidischen Distanz**

Damit für die Navigation nur stabile Merkmale zur Verfügung stehen, wurde die Toleranz beim Vergleich der Merkmalsvektoren entsprechend niedrig gewählt. Des Weiteren wurden nur SIFT-Merkmale verwendet, die in mindestens drei Bildern gefunden wurden.

Nach der Ausführung der genannten Filter wurden die zusammengefassten SIFT-Merkmale auf ihre richtige Zuordnung geprüft. Die Abbildung 3.10 zeigt, dass mehr als die Hälfte der SIFT-Punkte falsch zugeordnet wurden und somit unbrauchbar für die Navigation sind. Die

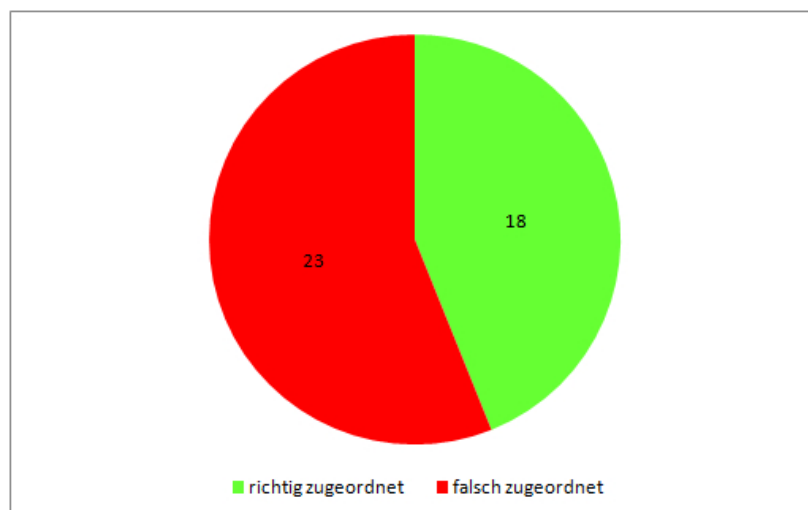


Abbildung 3.10.: Anzahl der richtigen/falschen Zuordnungen von SIFT-Merkmalen

Ursache hierfür, liegt in dem Mangel an markanten Strukturen in der Testumgebung. Es ist gleichzeitig eine periodische Beschaffenheit von Lampen, Türen und Fußleisten gegeben. In Abb. 3.11 wird deutlich, dass dadurch viele SIFT-Punkte eine ähnliche Eigenschaft aufweisen und sich somit der Merkmalsvektor kaum unterscheidet. Diese SIFT-Punkte sind daher potentielle Anwärter für eine Verwechslung und könnten beim Vergleich der Merkmalsvektoren falsch zusammengefasst werden. Um dieses Problem zu beheben wurde ein weiterer



Abbildung 3.11.: SIFT-Punkte mit hohem Verwechslungspotential

Filter eingesetzt:

- **Verwechslungsfilter**

Dieser Filter vergleicht für jedes Bild alle Merkmalsvektoren aller erkannten SIFT-Punkte mit der euklidischen Distanz. Sind zwei oder mehrere Merkmalsvektoren in einem Bild sehr ähnlich, so werden diese aus der Liste entfernt und nicht zur Berechnung verwendet.

Eine erneute Genauigkeitsüberprüfung zeigt die Wirkung der neuen Filterfunktion: Es stehen im Vergleich zum ersten Test nur noch rund 31 Prozent der SIFT-Punkte zur Navigation zur Verfügung. Diese sind jedoch zu über 90 Prozent richtig zugeordnet (Siehe Abb. 3.12).

Es wird somit deutlich, dass die Anzahl der SIFT-Merkmale die sich für die visuelle Navigation eignen, sehr gering ist. Das Diagramm 3.13 zeigt die Anzahl aller gefundenen SIFT-

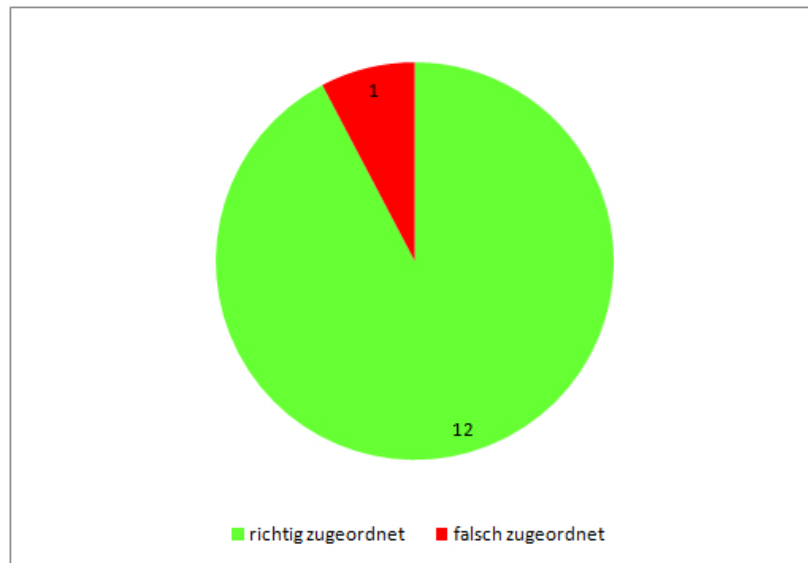


Abbildung 3.12.: Anzahl der richtigen/falschen Zuordnungen von SIFT-Merkmalen (mit Verwechslungsfilter)

Merkmale und die verbleibende Anzahl nach den jeweiligen Filtern. Am Ende bleiben nur noch rund 0,35 Prozent der ursprünglich gefundenen Merkmale übrig. In einem Raum, der mehr Struktur aufweist und damit noch mehr SIFT-Merkmale liefert, könnte eine höhere Ausbeute erzielt werden.

### 3.5.3. Ortsgenauigkeit der korrekt zugeordneten SIFT-Punkte

Die Punkte, die korrekt zugeordnet worden sind wurden mit den von Hand gemessenen Positionen im Raum verglichen. Die genauen Messwerte finden sich im Anhang B. Zur Veranschaulichung wurde die gesamte Abweichung (Summe der Abweichungen über alle Koordinaten) zusammen mit der Häufigkeit ihres Auftretens im Diagramm 3.14 dargestellt. Es wird deutlich, dass die meisten der berechneten Punkte eine Gesamtabweichung von 10 cm nicht überschreiten und somit für die Navigation sehr gut geeignet sind. Die Punkte, die eine Abweichung von mehr als 50 cm in der Summe aufweisen, lassen sich auf die ungünstige Aufnahmesituation zurückführen. Der größte Anteil der Fehlersumme aller berechneten Punkte lag dabei in der Y-Koordinate, also der Tiefe des Flures. Würde die Basisbreite der Testumgebung höher sein, bzw. läge ein größerer Winkel zwischen den Aufnahmekameras, so könnte hier eine Steigerung der Genauigkeit erreicht werden.

Die Abbildung 3.15 verdeutlicht die Problematik eines kleinen Winkels zwischen den Aufnahmekameras. Bereits ein kleiner Fehler der berechneten Kameralage verursacht eine hohe Ungenauigkeit der Raumpunktvermessung. Dieser Fehler ließe sich, durch die deutliche

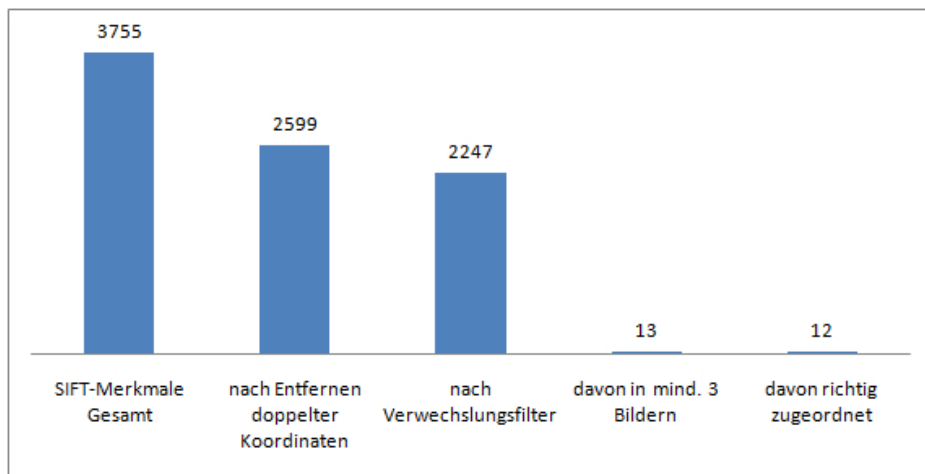


Abbildung 3.13.: Wirkung der Filterfunktionen auf die Anzahl der SIFT-Merkmale

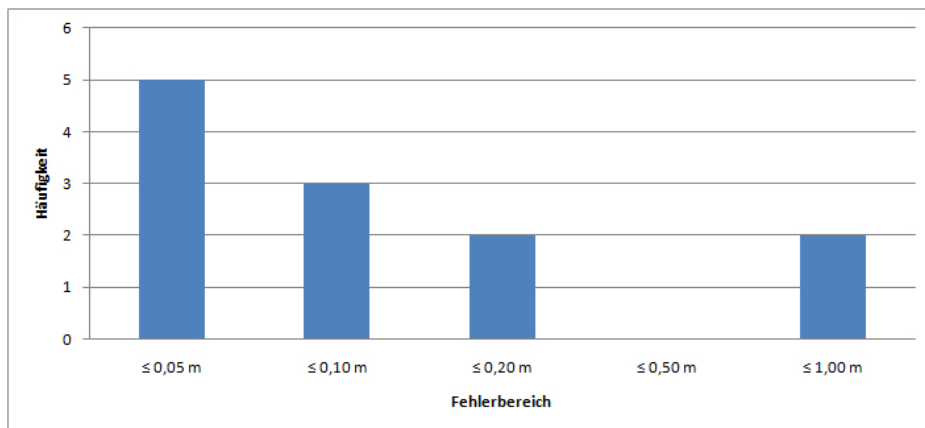


Abbildung 3.14.: Abweichungen richtig zugeordneter SIFT-Punkte

Erhöhung der Winkel zwischen den Kameras, verringern. In [Abbildung 3.16](#) wird eine gute Aufnahmesituation dargestellt. Aufgrund der Gegebenheiten der Testumgebung, lässt sich jedoch diese wünschenswerte Anordnung nicht realisieren.



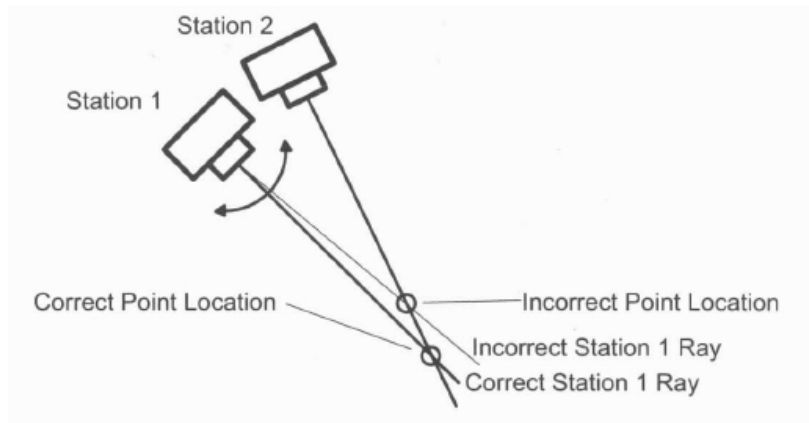


Abbildung 3.15.: Messfehler durch ungünstige Aufnahmesituation ([Systems 2007](#), S. 48)

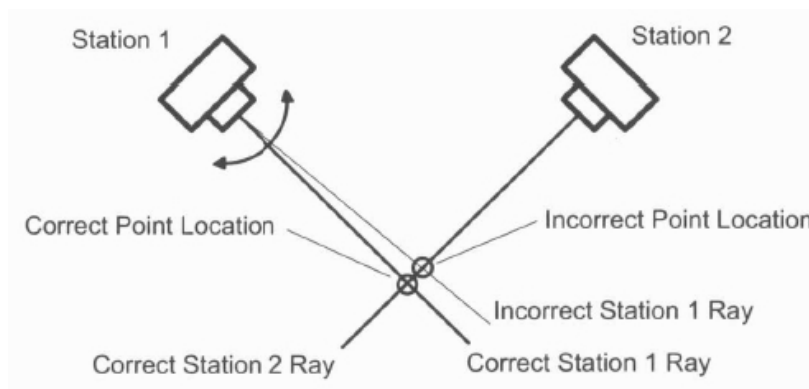


Abbildung 3.16.: Gute Aufnahmesituation durch einen großen Winkel zwischen den Kameras ([Systems 2007](#), S. 47)

## 4. Navigationsmodul

Dieses Kapitel beschreibt die Erweiterung eines bestehenden Navigationsmoduls zur Bestimmung der Kameraposition anhand von vermessenen SIFT-Merkmalen.

### 4.1. Einführung

In der Masterarbeit von Nico Manske wurde eine Möglichkeit geschaffen, ein Fahrzeug anhand von künstlichen Positionsmarken im Raum zu navigieren. Dazu wurden, aufbauend auf dem von Matthias Wilms (Wilms 2008) entwickelten Modul, zur Erkennung künstlicher Marken, weitere Module für Impresario entwickelt:

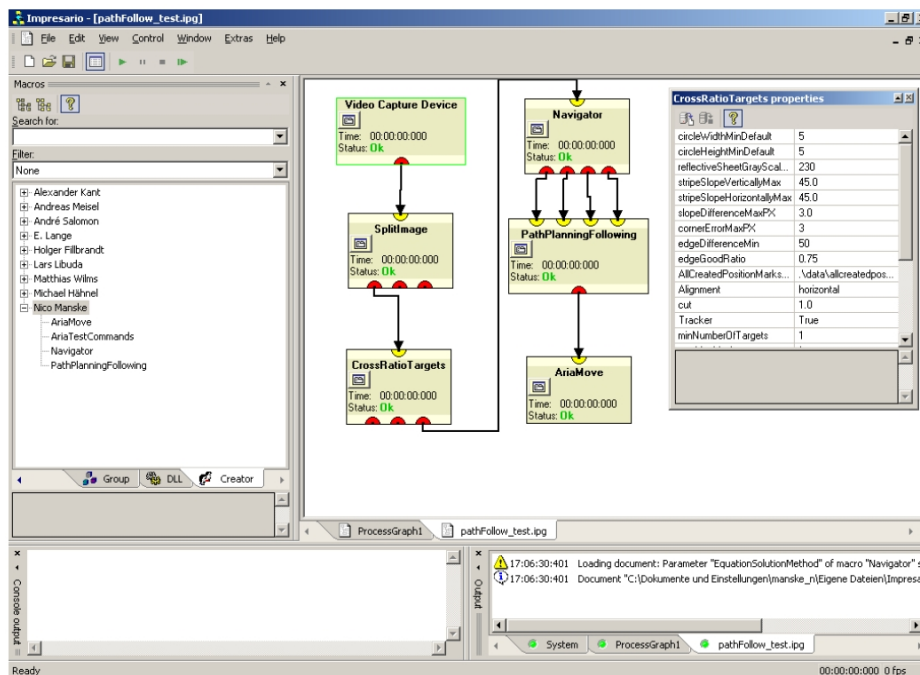


Abbildung 4.1.: Übersicht der bisherigen Navigationsmodule für Impresario (Manske 2008a, S. 58)

- **CrossRatioTargets**

Dieses von Matthias Wilms entwickelte Modul erkennt die künstlichen Positionsmarken und liefert eine Liste von Marken mit deren Bildkoordinaten.

- **Navigator**

Anhand der o.g. Liste und des von Nico Manske entwickelten Navigators kann mit Hilfe der Raumkoordinaten sowie den intrinsischen Kameraparametern der Aufnahmekamera, die aktuelle Position des Fahrzeugs berechnet werden. Die Liste der Marken und deren Positionen werden dabei über die Modulschnittstelle übergeben, die Kameraparameter und die Raumkoordinaten der Marken über XML-Dateien.

- **PathPlanningFollowing**

Dieses Modul implementiert eine vereinfachte Bahnplanung und -verfolgung für die Navigation ohne Hindernisse.

- **AriaMove**

AriaMove dient zur Steuerung des mobilen Roboters.

Damit an Stelle der künstlichen Positionsmarken mit den aus dem vorherigen Kapitel beschriebenen, vermessenen SIFT-Merkmalen navigiert werden kann, muss ein Modul entwickelt werden, welches in dem nun Folgendem beschrieben wird.

## 4.2. Navigation anhand von vermessenen SIFT-Merkmalen

Um die künstlichen Marken durch SIFT-Merkmale ersetzen zu können, wurde das CrossRatioTargets-Modul durch zwei andere Module ersetzt.

1. Das Modul von Emanuel Lange zur Auffindung von SIFT-Merkmalen wurde erweitert.
2. Ein neues Modul zur Identifizierung von SIFT-Merkmalen wurde entwickelt.

In Abb. 4.2 wird dieser Ablauf veranschaulicht.

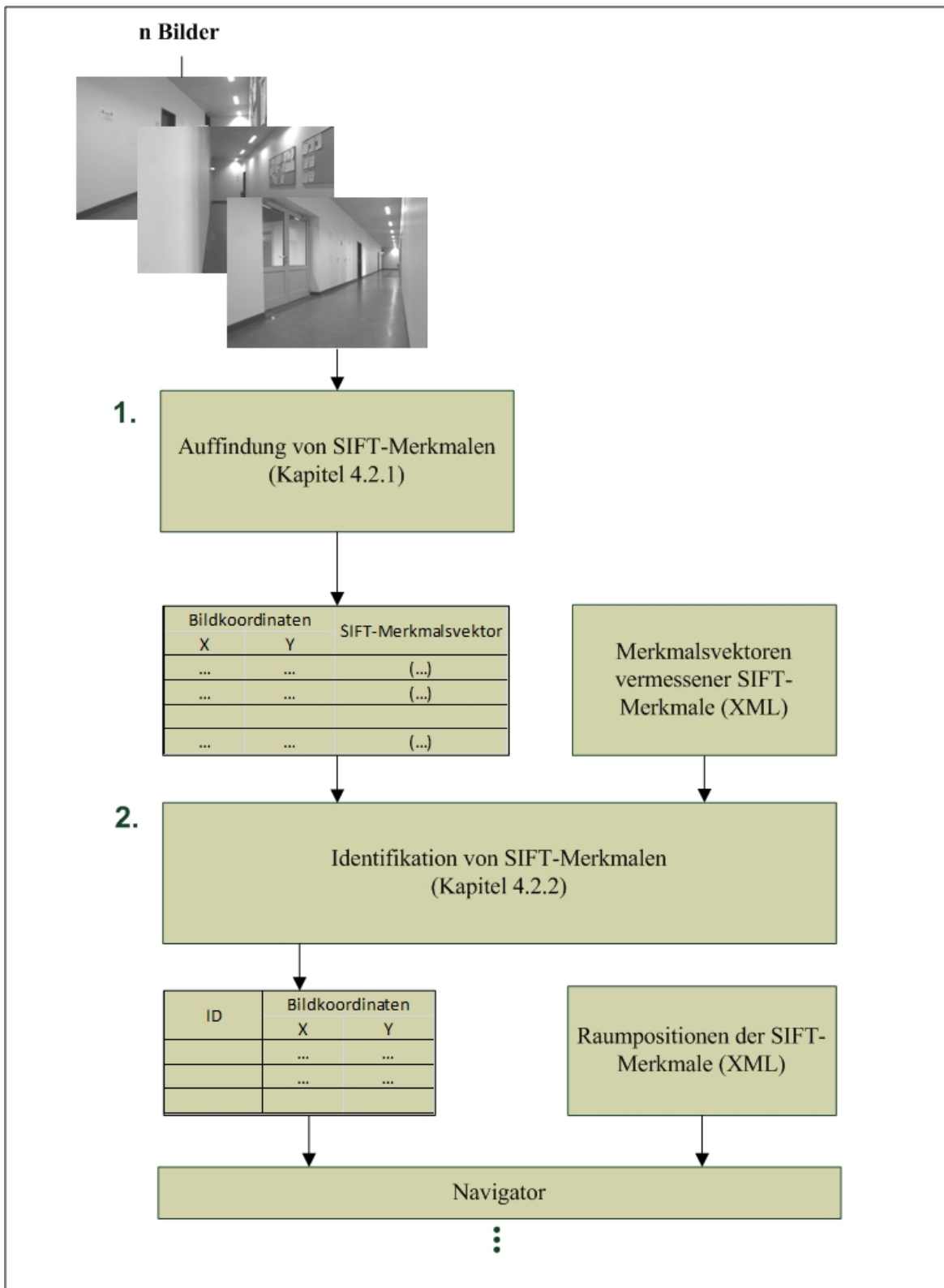


Abbildung 4.2.: Ablauf der Erweiterung für das Navigationsmodul

### 4.2.1. Auffindung von SIFT-Merkmalen

Zur Erkennung von SIFT-Merkmalen wird erneut auf das Impresario-Modul von Emanuel Lange zurückgegriffen. Es liefert weiterhin für jedes Bild die SIFT-Merkmale. Um mit diesen erkannten SIFT-Merkmalen direkt in Impresario weiter arbeiten zu können, wird das Modul um einen Ausgang erweitert. Die gefundenen SIFT-Merkmale werden nicht mehr in eine Datei extrahiert, sondern für jedes erkannte SIFT-Merkmal wird ein Objekt der Klasse `SiftPoint` erstellt und in einer Liste gespeichert. Am Ende des Modulablaufs steht somit eine komplette Liste aller gefundenen SIFT-Merkmale zur Verfügung. Diese wird für den nächsten Schritt an den neu erstellten Ausgang gelegt.

### 4.2.2. Identifikation von SIFT-Merkmalen

Nachdem nun alle SIFT-Merkmale für das aktuelle Bild ermittelt wurden, gilt es daraus alle die SIFT-Merkmale zu bestimmen die sich für die Navigation eignen. SIFT-Merkmale, die für die Navigation in Frage kommen, sind genau die, die mit dem Einrichtungsmodul vermessen wurden.

Das entwickelte Impresario-Modul `SIFT-ID` soll diese Aufgabe übernehmen und SIFT-Merkmale für die Navigation bestimmen. Dazu bietet das Modul folgende Schnittstellen an:

- Input: Eine Liste von `SiftPoint` Objekten, die in dem aktuellen Kamerabild gefunden worden sind
- Output: Eine Liste von `Mark` Objekten. Hier wurde aufgrund der Kompatibilität der vorhandene Datentyp aus der Masterarbeit von Nico Manske verwendet.

Weiterhin ist eine Liste aller vermessenen SIFT-Merkmalvektoren nötig, welche als Datei im XML-Format von dem Modul eingelesen wird (Siehe Abb. 4.3). Diese XML-Datei ist so aufgebaut, dass zu jeder ID mehrere Merkmalsvektoren zugeordnet werden. Die Raumpositionen dieser vermessenen SIFT-Merkmale können später über die ID bestimmt werden.

Zur Ermittlung der SIFT-Punkte für die Navigation wird, mit Hilfe der euklidischen Distanz, jeder Merkmalsvektor der Eingangsliste mit jedem vermessenen Merkmalsvektor der XML-Datei verglichen. Hierbei kann die Toleranz wieder variabel eingestellt werden. Ist ein SIFT-Merkmal der Eingangsliste innerhalb der Toleranz identisch mit einem der vermessenen Merkmale aus der XML-Datei, so wird diesem Merkmal die dazugehörige ID zugewiesen. Im Anschluss daran wird die ID, zusammen mit den Bildkoordinaten des SIFT-Merkmals, in einem Objekt der Klasse `Mark` gespeichert. Für jedes weitere übereinstimmende Merkmal werden jeweils weitere Objekte angelegt und alle in einer Liste gespeichert. Am Ende wird diese Liste über den Ausgang an das Navigator-Modul übergeben. Das Navigator-Modul

```

- <PositionMarks>
- <PositionMark>
  <ID>18</ID>
  <siftVector>0 1 52 46 17 1 0 0 0 17 52 21 36 3 0 0 0 1 10 57 67 0 0 0 0 27 54 16 0 0 0 11
    24 122 122 64 4 1 2 122 122 122 16 14 1 0 9 27 14 3 15 122 19 0 1 0 0 2 12 39 15 2 0 16
    6 11 29 110 122 122 44 122 19 4 1 4 7 98 122 34 1 0 0 71 85 11 20 0 0 0 13 55 42 0 0 0
    0 0 10 64 122 9 1 0 0 0 18 18 87 45 0 0 0 0 28 45 9 4 0 0 0 0 12 36 29 0</siftVector>
  <siftVector>0 2 80 47 14 1 0 0 1 27 62 14 35 4 0 0 0 2 11 40 75 3 0 1 0 0 29 33 13 9 2 1 17
    24 121 121 96 5 0 0 121 121 107 7 14 1 0 2 34 15 2 4 121 21 1 2 1 5 3 8 40 16 3 2 21 3 3 8
    121 121 121 27 121 10 1 0 5 7 104 121 42 1 0 0 87 63 5 17 0 0 0 9 25 33 17 1 0 0 0 7 53
    121 5 3 0 0 0 18 16 94 43 0 0 0 0 36 50 7 5 0 0 0 0 12 45 32 0</siftVector>
  <siftVector>0 1 62 53 16 0 0 0 0 18 57 24 32 1 0 0 0 1 10 61 57 0 0 0 0 34 56 11 0 0 0 12
    29 121 121 71 6 1 2 121 121 121 13 10 0 0 9 25 15 3 20 121 10 0 1 0 0 1 21 46 12 0 0 17
    6 9 25 106 121 121 41 121 19 4 0 4 8 96 121 31 1 0 0 76 75 6 19 0 0 0 0 19 63 18 0 0 0 0
    13 74 121 7 1 0 0 0 18 20 90 43 0 0 0 0 33 54 8 4 0 0 0 0 7 48 30 0</siftVector>
  <siftVector>0 2 77 53 16 0 0 0 1 25 64 19 33 1 0 0 0 2 9 43 67 2 0 0 0 0 27 37 13 5 1 0 15 26
    120 120 101 6 0 1 120 120 107 8 11 1 0 4 32 13 2 7 114 15 1 1 0 0 3 7 43 15 2 0 19 4 4 13
    120 120 120 31 120 10 1 0 4 7 96 120 38 0 0 0 82 58 5 19 0 0 0 5 27 35 12 0 0 0 0 13 81
    117 6 2 0 0 0 18 20 97 46 0 0 0 0 37 54 8 5 0 0 0 0 9 48 30 0</siftVector>
</PositionMark>
- <PositionMark>
  <ID>84</ID>
  ...

```

Abbildung 4.3.: Vermessene Merkmalsvektoren im XML-Format

muss nun noch die Raumkoordinaten der bereits vermessenen SIFT-Punkte erhalten, welche ihm in einer, im Bezug zur Masterarbeit von Nico Manske, strukturgleichen XML-Datei übergeben werden (Siehe Abb. 4.4). Somit kann, aus den Raumkoordinaten der SIFT-Merkmale und den über die ID zugeordneten Raumkoordinaten, die Lage und Position der Aufnahme-kamera berechnet werden.

```

- <RoomModel>
- <marks>
  <mark id="18" x="1.86556" y="0.0309961" z="1.73765" />
  <mark id="84" x="-0.0479348" y="-0.0430695" z="2.96201" />
  <mark id="105" x="1.86576" y="0.0428941" z="1.73703" />
  <mark id="130" x="0.0559296" y="-0.140684" z="0.177521" />
  <mark id="151" x="-0.0210219" y="-1.49465" z="0.167798" />
  <mark id="192" x="1.84063" y="-0.0462503" z="1.81459" />
  <mark id="232" x="0.116594" y="-0.375526" z="0.199947" />
  <mark id="769" x="-3.75101" y="6.84176" z="4.26973" />
  <mark id="780" x="0.0809654" y="-0.0598549" z="-0.172466" />
  <mark id="861" x="1.06175" y="-1.69027" z="-0.0664503" />
  <mark id="951" x="0.312576" y="-1.94424" z="-0.0283413" />
  <mark id="1069" x="1.2022" y="1.03884" z="-0.212454" />
  <mark id="1511" x="-0.0749768" y="-13.4657" z="0.26445" />
</marks>
</RoomModel>

```

Abbildung 4.4.: Raumpositionen im XML-Format

Durch die Kompatibilität der neuen mit den bereits vorhandenen Modulen, können alle Module wieder in Impresario verknüpft und ausgeführt werden.

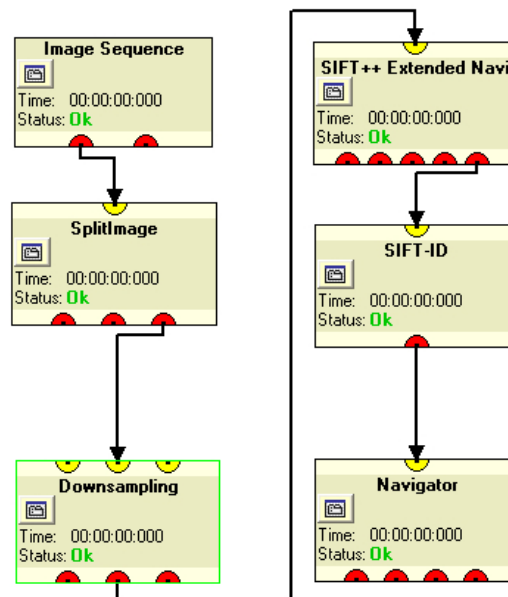


Abbildung 4.5.: Übersicht der neuen Navigationsmodule für Impresario

### 4.3. Genauigkeitsanalyse

Die Analyse der Genauigkeit für das Navigationsmodul gliedert sich in zwei Abschnitte:

1. Genauigkeitsanalyse der berechneten Kameraposition und Lage
2. Analyse der Auffindbarkeit von Navigationsmerkmalen

#### 4.3.1. Analyse der Navigationsgenauigkeit

Für die Analyse der Navigationsgenauigkeit wurden nur Bilder verwendet, die auch im Einrichtungsmodul benutzt worden sind. Dadurch ist gewährleistet, dass SIFT-Merkmale mit exakt dem gleichen Merkmalsvektor wiedergefunden werden können. Die Toleranz zur Erkennung gleicher SIFT-Merkmale im Modul `SIFT-ID` ist somit gleich null.

Die berechneten Werte stellen damit eine optimale Situation für die visuelle Navigation anhand von SIFT-Merkmalen dar. Alle berechneten Ergebnisse sind daher die bestmöglichen Werte und zeigen damit die höchste Genauigkeit für die gewählte Testumgebung auf. Abbildung 4.6 zeigt wie häufig welche Anzahl von SIFT-Merkmalen pro Bild gefunden wurde. Für die visuelle Navigation sind mindestens drei Punkte pro Bild vonnöten. Es wird deutlich,

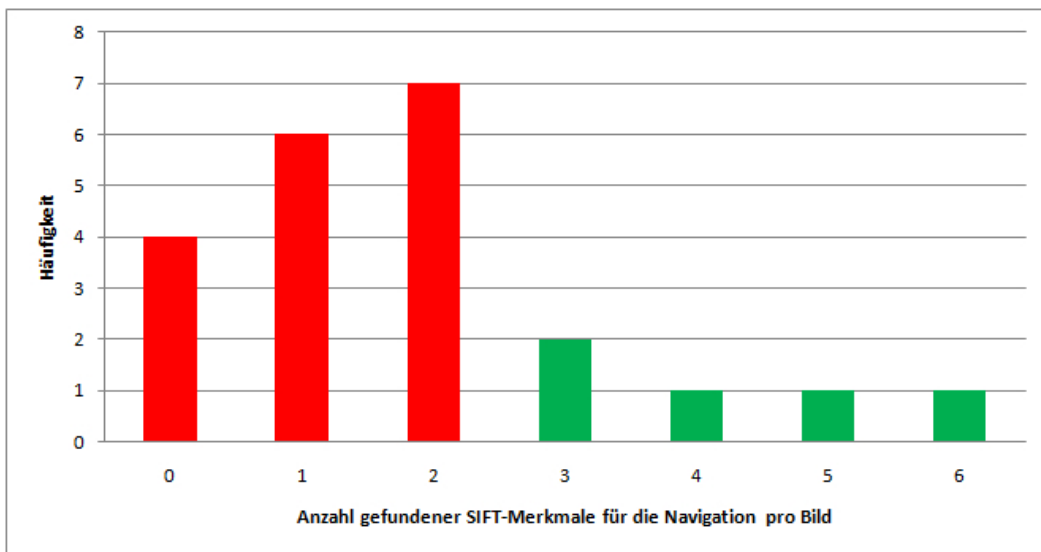


Abbildung 4.6.: Übersicht zur Auffindbarkeit von SIFT-Navigationspunkten

dass in den meisten Bildern nur zwei oder weniger SIFT-Merkmale gefunden werden konnten und diese sich somit nicht für die Navigation eignen. Nähere Ausführungen dazu finden sich im Abschnitt 4.3.2.

Die Bilder, bei denen eine Positionsbestimmung mit den SIFT-Merkmalen möglich war, werden nun auf ihre Genauigkeit analysiert. Da es sich bei den verwendeten Bildern ausschließlich um Bilder aus dem Einrichtungsmodul handelt, können als Referenz die mit dem Photomodeler errechneten Positionen der Aufnahmekameras verwendet werden. Die Genauigkeit der Photomodeler-Ergebnisse kann als deutlich genauer angenommen werden, da die Kameraposition mit einer Bündelausgleichung über viele Bilder gewonnen wird.

In der Tabelle 4.1 sind die berechneten Werte zur Kameraposition zusammen mit den Vergleichswerten aufgeführt. Es wird deutlich, dass der Positionsmessfehler mit dem Abstand

Photomodeler			Navigationsmodul		
X	Y	Z	X	Y	Z
2,0488	-4,2317	0,841	2,0848	-4,2401	0,8425
0,1203	-5,9347	0,799	1,7642	-5,9977	2,0518
2,1346	-6,0059	0,7672	2,0718	-5,9573	0,793
2,1047	-7,8312	0,9177	2,1139	-7,8728	0,8476
2,0596	-14,685	0,9119	2,4739	-19,5292	1,3166

Tabelle 4.1.: Navigationsgenauigkeit der Kameraposition. Alle Angaben in m  
der Merkmalspunkte steigt. Im Nahbereich hingegen ist der Fehler sehr gering.



Die Tabelle 4.2 zeigt die Werte zur Lage der Kamera mit den Vergleichswerten aus dem Photomodeler.

Photomodeler			Navigationsmodul		
$\omega$	$\phi$	$\kappa$	$\omega$	$\phi$	$\kappa$
97,383	17,8161	-1,1971	97,3515	18,2274	-1,3372
98,1183	-23,5387	4,1357	88,0956	-8,3857	19,27
98,4313	24,3397	-2,3619	98,3322	23,8746	-2,1018
93,3666	9,5062	1,1904	93,7942	9,4919	0,9208
88,6786	22,8814	0,9592	86,3144	22,4204	9,3707

Tabelle 4.2.: Navigationsgenauigkeit der Kameralage. Alle Angaben in  $^{\circ}$

In Abbildung 4.7 werden einige der berechneten Werte grafisch dargestellt. Dabei werden drei verschiedene Fälle unterschieden:

- Abweichung bei nur drei Bezugspunkten
- Abweichung bei sechs Bezugspunkten
- Mittlere Abweichung aller Bilder

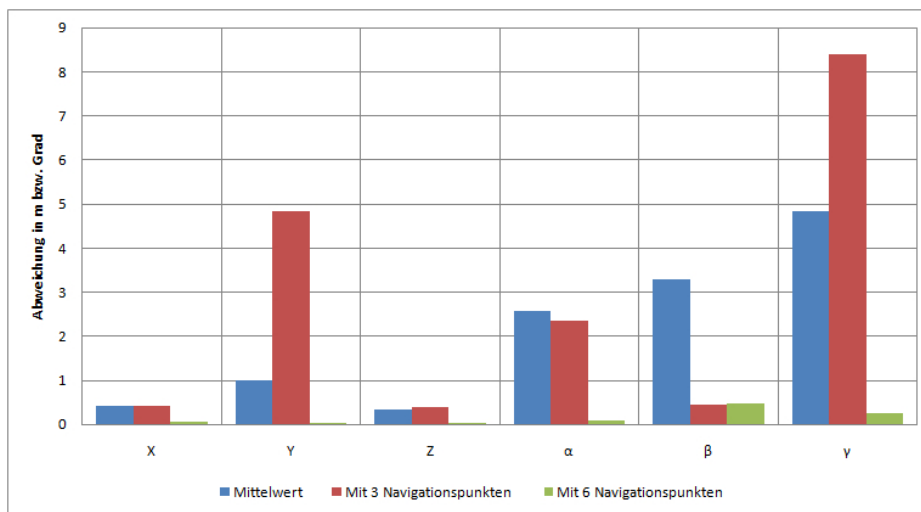


Abbildung 4.7.: Abweichungen der SIFT-Navigationspunkte

Das Diagramm zeigt, dass bei der Navigation mit nur drei Bezugspunkten, also der Mindestanzahl für die visuelle Navigation, die Abweichung sehr hoch ist. Werden jedoch sechs Punkte für die Navigation verwendet, ist die Abweichung der berechneten Kameraposition sehr gering. Im Schnitt liegt die Differenz zu den tatsächlichen Aufnahmepositionen bei unter

einem Meter. Bei der Lage der Kamera beträgt die durchschnittliche Abweichung nicht mehr als fünf Grad.

Die Untersuchung hat demnach gezeigt, dass bei einer ausreichend hohen Zahl an auffindbaren, vermessenen SIFT-Merkmalen eine sehr genaue Positionsbestimmung und damit eine sehr genaue Kameranavigation möglich ist. Ist die Zahl der Bezugspunkte jedoch minimal, so ist nur eine sehr grobe Positions- und Lagebestimmung anhand von SIFT-Merkmalen möglich.

### 4.3.2. Auffindbarkeit von SIFT-Merkmalen

Wie bereits in Abbildung 4.6 dargestellt, scheint die Wiederauffindbarkeit von vermessenen SIFT-Merkmalen im Bild das größte Problem darzustellen. Um zu testen wie gut sich SIFT-Merkmale wieder auffinden lassen, wurde eine Kamerafahrt in der Testumgebung simuliert.

Es wurden dafür 14 Aufnahmen mit der Kamera erstellt, bei denen sich die Kamera in der Mitte des Flures befand. Nach jeder Aufnahme wurde die Kamera näher in Richtung Wand bewegt. Die erstellten Bilder wurden dem Navigationsmodul übergeben um die Position der



Abbildung 4.8.: Bilder der simulierten Kamerafahrt

Kamera dadurch zu bestimmen. Da hierbei nicht mit einer exakten Übereinstimmung (euklidische Distanz der Merkmalsvektoren = 0) der Merkmalsvektoren zu rechnen ist, muss die Toleranzgrenze für die Erkennung nach oben korrigiert werden. Bei einer geringen Toleranzgrenze ( $\leq 150$ ) wurden maximal zwei SIFT-Navigationspunkte gefunden. Wie bereits erwähnt sind für die Navigation mindestens drei Referenzpunkte nötig. Die Toleranzgrenze muss also noch weiter erhöht werden.

Bei einer hohen Toleranz ( $> 200$ ) werden zwar nun ausreichend SIFT-Navigationspunkte gefunden, die meisten sind jedoch falsch zugeordnet und die berechnete Kameraposition somit falsch. Die unrichtige Zuordnung der SIFT-Merkmale liegt wieder an der ähnlichen

Beschaffenheit des Raumes. In den Kamerabildern werden viele SIFT-Merkmale erkannt, die ein hohes Verwechslungspotential aufweisen (Siehe Abb. 4.9).



Abbildung 4.9.: Verwechslungsfehler bei der Navigation

Auch in diesem Fall spiegelt sich die ungünstige Beschaffenheit der Testumgebung wieder:

1. Aufgrund mangelnder Struktur im Raum können nur wenige, eindeutige SIFT-Merkmale gefunden werden.
2. Die gefundenen Merkmale weisen alle eine sehr ähnliche Beschaffenheit auf und haben somit ein sehr hohes Verwechslungspotential.

Es konnte bei diesem Test keine Konfiguration der Toleranzgrenzen gefunden werden, die eine korrekte Positionsbestimmung möglich machte. Entweder wurden zu wenige Punkte erkannt oder die gefundenen Punkte wurden falsch zugeordnet.

# 5. Schlussbetrachtung

In diesem Kapitel soll noch einmal die komplette Arbeit zusammengefasst und ein Ausblick gegeben werden.

## 5.1. Fazit

Das angestrebte Ziel dieser Arbeit war die Ersetzung von künstlichen Positionsmarken durch SIFT-Merkmale. Durch die, im Einrichtungsmodul, berechneten Raumkoordinaten der SIFT-Merkmale sollte es, mit dem Navigationmodul, möglich sein die Position der Aufnahmekamera eines mobilen Roboters zu bestimmen. Beide Module sollten dabei auf ihre Genauigkeit hin überprüft und analysiert werden. Die zu erstellenden Module konnten erfolgreich realisiert und getestet werden. Bei den durchgeführten Genauigkeitsanalysen zeigten sich Stärken und Schwächen der Navigation mit Hilfe von SIFT-Merkmalen auf.

Die Genauigkeit der Navigation hängt sehr von den Gegebenheiten des Raumes ab, in dem navigiert werden soll. Der in dieser Arbeit gewählte Testraum hat sich hierbei, aus folgenden Gründen, als schwierige Umgebung herausgestellt:

- Der Raum weist kaum Struktur auf sondern besteht aus gleichfarbigen, musterlosen Wänden und Türen.
- Viele Gegenstände gleicher Struktur wiederholen sich periodisch und bieten somit ein hohes Verwechslungspotential.
- Der Raum hat eine geringe Basisbreite im Vergleich zu seiner Tiefe.

Diese schwierigen Eigenschaften haben zur Folge, dass bei der Navigation mit SIFT-Merkmalen im Fernbereich nur eine geringe Genauigkeit der Kameraposition erreicht werden kann. Um der Gefahr der falschen Zuordnung von SIFT-Merkmalen vorzubeugen wurden durch Filterfunktionen ein Großteil der erkannten SIFT-Punkte herausgefiltert. Dies hat zur Folge, dass am Ende nur noch eine geringe Anzahl an SIFT-Merkmalen zur Verfügung steht. Es besteht somit die Gefahr, dass bei der Navigation nicht immer die Mindestanzahl von drei SIFT-Merkmalen erkannt wird.

Im alltäglichen Einsatz sind die o.g. ungünstigen Navigationsvoraussetzungen jedoch häufig gegeben. Es konnte gezeigt werden, dass auch unter diesen schwierigen Gegebenheiten der Umgebung eine ausreichende Anzahl von SIFT-Merkmalen für die visuelle Navigation ermittelt werden kann. Weiterhin konnte festgestellt werden, dass bei SIFT-Merkmalen im Nahbereich eine hohe Positionsgenauigkeit der Aufnahmekamera erreicht werden kann.

Die Konsequenz aus den genannten Stärken und Schwächen des Verfahrens ist daher, dass sich die Navigation mit SIFT-Merkmalen durchaus für ein differentielles Navigationsverfahren eignet. In den Fällen in denen dem mobilen Roboter weniger als drei SIFT-Merkmale für die Navigation zur Verfügung stehen, könnte der Roboter, durch das mitloggen von Geschwindigkeit, Lenkwinkel etc., seine Position abschätzen. Sobald wieder ausreichend SIFT-Merkmale zur Verfügung stehen kann die genaue Position wieder neu bestimmt werden. Dabei kann die Vertrauenswürdigkeit der neu berechneten Position gut abgeschätzt werden. Befinden sich die SIFT-Merkmale in einem großen Abstand zur Kamera ist die berechnete Position eher ungenau. Befindet sich der Roboter im geringen Abstand zu den Merkmalen ist die Position genauer.

## 5.2. Ausblick

Die Untersuchungen in dieser Arbeit haben gezeigt, dass die Navigation anhand von SIFT-Merkmalen möglich ist. Zur Verbesserung könnten noch weitere Tests durchgeführt werden. Eine mögliche Optimierung wäre die Verwendung einer anderen Navigationsumgebung. Dabei sollte darauf geachtet werden, dass diese eine größere Anzahl an markanten Strukturen und Objekten aufweist. Dadurch stünden mehr SIFT-Merkmale für die Navigation zur Verfügung und es könnte eine höhere Wiederauffindbarkeit der SIFT-Merkmale realisiert werden.

Desweiteren wurde in dieser Arbeit die SIFT-Navigation nur anhand von Bildern analysiert. Diese stellen keine Herausforderungen hinsichtlich der Berechnungszeiten dar. Es müsste untersucht werden ob die Zeiten zur Berechnung auch der Echtzeitanforderung eines Videostreams genügen.

# Literaturverzeichnis

- [Blume 2008] BLUME, Holger: *Wahrscheinlichkeitsbasierte Methoden zur autonomen Führung von Fahrzeugen in unsicherer Umgebung*. 2008. – URL <http://edok01.tib.uni-hannover.de/edoks/e01dh08/569293340.pdf>. – Abruf: 2008-12-08
- [Brown und Lowe 2003] BROWN, Matthew ; LOWE, David G.: *Recognising Panoramas*. 2003. – URL <http://www.cs.ubc.ca/~mbrown/papers/iccv2003.pdf>. – Abruf: 2009-05-25
- [Duif 2006] DUIF, Thomas: *Entwicklung von Bildverarbeitungs-Algorithmen mit Impresario*. 2006. – URL [http://www.rz.rwth-aachen.de/global/show\\_document.asp?id=aaaaaaaaaavhxq](http://www.rz.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaaavhxq). – Abruf: 2009-06-22
- [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](http://www.informatik.uni-augsburg.de/lehrstuehle/mm1/publications/reports/2006_13_esc/tr2006-13.pdf). – Abruf: 2009-05-20
- [psb FTS carobot 2007] FTS CAROBOT psb: 2007. – URL [http://www.logistik-bilder.de/details.php?image\\_id=1132&sessionid=3cf7b1f307c1c8819e8f07aaa8ef4887](http://www.logistik-bilder.de/details.php?image_id=1132&sessionid=3cf7b1f307c1c8819e8f07aaa8ef4887). – Abruf: 2008-07-03
- [Harris und Stephens 1988] HARRIS, Chris ; STEPHENS, Mike: *A Combined Corner and Edge Detector*. 1988. – URL [http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A\\_Combined\\_Corner\\_and\\_Edge\\_Detector.pdf](http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A_Combined_Corner_and_Edge_Detector.pdf). – Abruf: 2009-05-20
- [Hartley und Zisserman 2003] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple View Geometry in computer vision*. Cambridge University Press, 2003. – ISBN 0-521-54051-8
- [Jähne 1997] JÄHNE, Bernd: *Digitale Bildverarbeitung*. Springer, 1997. – ISBN 3-540-61379-x
- [Laaraiedh u. a. 2009] LAARAIEDH, M. ; AVRILLON, S. ; UGUEN, B.: Hybrid Data Fusion Techniques for Localization in UWB Networks. In: *Proceedings of the 6th workshop on positioning, navigation and communication 2009* (2009), S. 51–57

- [Lange 2008] LANGE, Emmanuel: *Verfolgung markanter Raumpunkte in Video-bildsequenzen anhand skalierungs- und rotationsinvarianter Merkmale*. 2008. – URL [http://opus.haw-hamburg.de/volltexte/2008/611/pdf/bachelorarbeit\\_emmanuel\\_lange.pdf](http://opus.haw-hamburg.de/volltexte/2008/611/pdf/bachelorarbeit_emmanuel_lange.pdf). – Abruf: 2009-04-15
- [Lowe 2004] LOWE, David G.: *Distinctive Image Features from Scale-Invariant Keypoints*. 2004. – URL <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>. – Abruf: 2008-12-01
- [LTI-Lib 2005] LTI-LIB: *LTI-Lib*. 2005. – URL <http://ltilib.sourceforge.net/doc/homepage/index.shtml>
- [Manske 2008a] MANSKE, Nico: *Kamerabasierte Präzisionsnavigation mobiler Systeme im Indoor-Bereich*. Masterarbeit HAW Hamburg. 2008. – URL [http://opus.haw-hamburg.de/volltexte/2009/725/pdf/NicoManske\\_Abschlussarbeit.pdf](http://opus.haw-hamburg.de/volltexte/2009/725/pdf/NicoManske_Abschlussarbeit.pdf). – Abruf: 2009-05-20
- [Manske 2008b] MANSKE, Nico: *Verfahren zur Positionsbestimmung im Indoor-Bereich*. Seminararbeit HAW Hamburg. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/manske/bericht.pdf>. – Abruf: 2009-05-20
- [Meisel 2008] MEISEL, Andreas: *Projektive Geometrie des Raumes (3D)*. 2008. – URL [http://www.informatik.haw-hamburg.de/uploads/media/AW\\_3DBV\\_V02.pdf](http://www.informatik.haw-hamburg.de/uploads/media/AW_3DBV_V02.pdf)
- [Nehmzow 2002] NEHMZOW, Ulrich: *Mobile Robotik: Eine praktische Einführung*. Springer, 2002. – ISBN 3-540-42858-5
- [Smith 1995] SMITH, Stephen M.: *SUSAN - A New Approach to Low Level Image Processing*. 1995. – URL <http://users.fmrib.ox.ac.uk/~steve/susan/susan/susan.html>. – Abruf: 2009-05-20
- [Stegmann und Liu 2007] STEGMANN, Nadine ; LIU, Steven: *Eingangsgrößenrekonstruktion zur Trajektorienfolge mit elementarer Sensorik*. In: BERNS, Karsten (Hrsg.) ; LUKSCH, Tobias (Hrsg.): *Autonome Mobile Systeme 2007: 20. Fachgespräch Kaiserslautern, 18./19. Oktober 2007*. Berlin : Springer, 2007 (Informatik aktuell), S. 82–88. – ISBN 978-3-540-74763-5
- [Systems 2007] SYSTEMS, Eos: *PhotoModeler 6 User Guide*. 1st, Januar 2007
- [Systems 2009] SYSTEMS, Eos: *PhotoModeler*. 2009. – URL <http://www.photomodeler.com>. – Abruf: 2009-07-09

- [Wilms 2008] WILMS, Matthias: *Lokalisierung und Verfolgung von codierten Messmarken mit Hilfe von Bildverarbeitung*. Bachelorarbeit HAW Hamburg. 2008. – URL <http://opus.haw-hamburg.de/volltexte/2009/736/pdf/BAWilms.pdf>. – Ab-ruf: 2009-06-23



# A. UML Klassendiagramme

Hier finden sich die UML-Klassendiagramme der erstellten Datentypen für diese Arbeit:

- **SiftPoint**

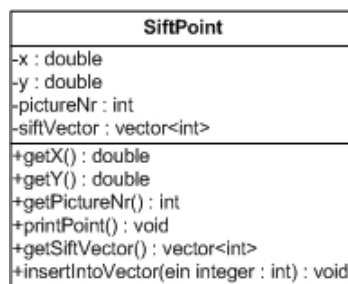


Abbildung A.1.: SiftPoint Klassenbeschreibung

Objekte dieser Klasse beschreiben einen SIFT-Punkt. Jeder SIFT-Punkt besitzt neben den Bildkoordinaten `x` und `y`, dem Merkmalsvektor `siftVector` auch eine Bildnummer `pictureNr`. Damit kann festgehalten werden, in welchem Bild dieser SIFT-Punkt gefunden worden ist.

Über die `get`-Funktionen lassen sich diese Variablen abrufen. `insertIntoVektor` dient der Erstellung des Merkmalsvektors. Mit `printPoint` werden alle Informationen des SIFT-Punktes auf der Konsole ausgegeben.

- **SameSiftPoint**

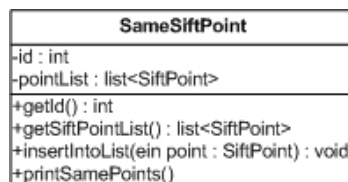


Abbildung A.2.: SameSiftPoint Klassenbeschreibung

Die Klasse `SameSiftPoint` fasst mehrere Objekte der Klasse `SiftPoint` in der `pointList` zusammen. Jeder `SameSiftPoint` hat eine eindeutige ID, welche für

die Zuordnung der berechneten Raumkoordinate nötig ist. Die Funktionen bieten hier auch die Möglichkeiten zum Abrufen der Informationen und zum Hinzufügen neuer `SiftPoint` Objekte.

- **Camera**

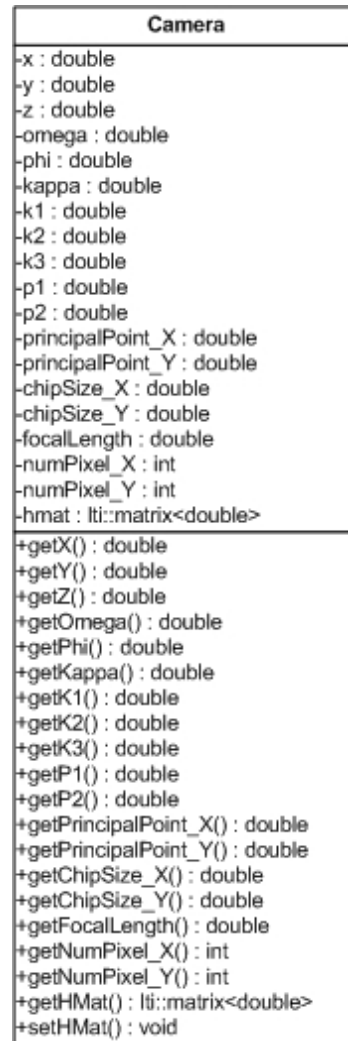


Abbildung A.3.: Camera Klassenbeschreibung

Die Klasse `Camera` enthält alle benötigten Kameraparameter. Außerdem wird für jede Aufnahmekamera die Projektionsmatrix `HMat` erstellt und gespeichert. Über die Funktionen der Klasse kann man diese Daten schließlich abrufen.

- **Mark**

Diese Hilfsklasse stammt aus den vorangegangenen Projekten und beschreibt eine

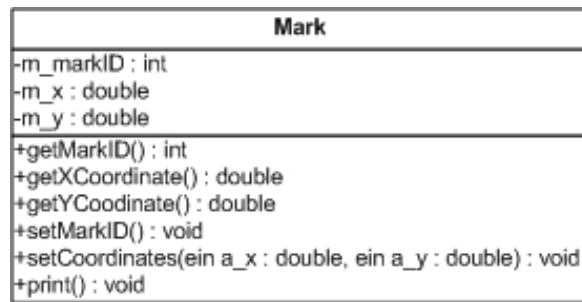


Abbildung A.4.: Mark Klassenbeschreibung

erkannte Positionsmarke (jetzt SIFT-Merkmale) mit dessen Bildkoordinaten und ID. Anhand dieser ID kann später die Raumkoordinate zugeordnet werden.

## B. Messergebnisse des Einrichtungsmoduls

In der folgenden Tabelle werden die, durch die entwickelte Software berechneten Raumkoordinaten der SIFT-Merkmale mit den, von Hand durchgeführten Kontrollmessungen aufgestellt.

Eigene Software			Kontrollmessung		
X	Y	Z	X	Y	Z
1,8655	0,0309	1,7376	1,87	0	1,73
-0,0479	-0,043	2,96	0	0	3,02
0,0559	-0,14	0,1775	0	0,01	0,15
-0,021	-1,4946	0,1677	0	-1,48	0,17
1,8657	0,0428	1,7370	1,87	0	1,73
1,8403	-0,0462	1,8145	1,85	0	1,82
0,1165	-0,3755	0,1999	0	-0,01	0,15
0,0809	-0,0598	-0,1724	0,45	-1,37	0
1,0617	-1,6907	-0,0664	0,99	-1,65	0
0,3125	-1,9442	-0,0283	0,45	-1,37	0
-0,0749	-13,4657	0,2644	0	-14,075	0,38
1,2022	1,0388	-0,2124	1,26	0	0

Tabelle B.1.: Messwerte der SIFT-Merkmale des Einrichtungsmoduls. Alle Angaben in m.

## C. Inhalt der CD

Folgende Ordner sind auf der beigelegten CD zu finden:

- **Bachelorarbeit:** Enthält die Arbeit im PDF Format.
- **Einrichtungssoftware:** Der Quellcode für das Einrichtungsmodul.
- **Arbeit Lange:** Die Bachelorarbeit von Emmanuel Lange mit dem modifizierten SIFT++ Modul für das Einrichtungs- und Navigationsmodul.
- **SIFT-ID:** Hier findet sich der Quellcode zum SIFT-ID Modul.
- **DLL:** Die erstellten DLL Dateien für Impresario.
- **Bilder:** Enthält alle genutzten Bilder für das Einrichtungsmodul, die Kalibrierungsbilder für die Kamera und die Bilder der simulierten Kamerafahrt, sowie die Photomodeler-Projektdateien
- **Grundriss:** Enthält den erstellten Grundriss der Testumgebung für Microsoft Visio 2007 und als PDF.

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. August 2009

Ort, Datum

Unterschrift