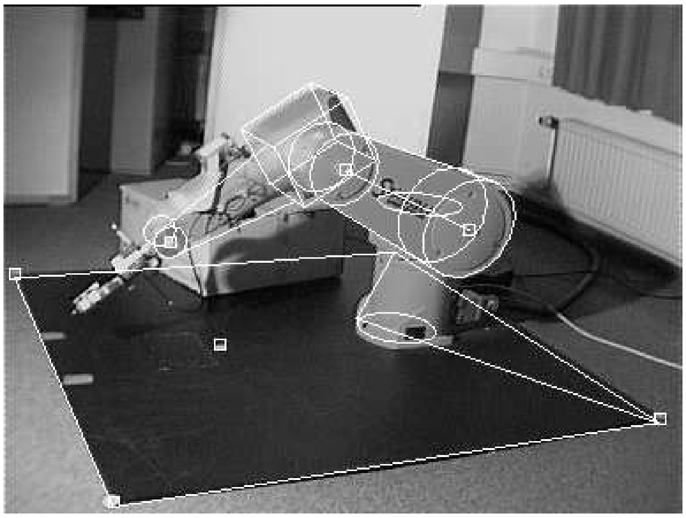
Diplomarbeit

Poseschätzung kinematischer Ketten



durchgeführt am:

Lehrstuhl für Kognitive Systeme
Institut für Informatik und Praktische Mathematik
Christian–Albrechts–Universität zu Kiel
Prof. Dr. G. Sommer

ausgearbeitet von:

Oliver Granert

Betreuer: Bodo Rosenhahn

März 2002

Zusammenfassung

Thema dieser Arbeit ist die Poseschätzung kinematischer Ketten. Ein vorgegebenes dreidimensionales CAD-Modell wird unter Verwendung von Bildinformationen an die von einem realen Objekt vorgegebene Pose angepasst. Die Bildinformationen werden in dieser Arbeit durch extrahierte Punkte und Geraden repräsentiert.

Um die dreidimensionalen kinematischen Informationen aus den zweidimensionalen projektiven Bilddaten gewinnen zu können, wird auf das CAD-Modell zurückgegriffen, das die Geometrie und Kinematik des Objektes beschreibt. Das Modellwissen führt zu Zwangsbedingungen, die die vorhandenen Bilddaten ergänzen und die Berechnung der kinematischen Konfiguration ermöglichen.

Ausgehend von den mathematischen Grundlagen zur Behandlung von kinematischen und projektiven Problemen wird ein Verfahren entwickelt, das die vorhandenen Sensordaten und Zwangsbedingungen in einem linearen Gleichungssystem zusammenfasst. Zur mathematischen Modellierung werden dazu Erkenntnisse aus der Gruppen und Lie-Theorie herangezogen.

Die Experimente am Ende der Arbeit demonstrieren die Funktion und Qualität des vorgestellten Verfahrens und zeigen die Anwendung der entwickelten C++ Klassenbibliothek.

Schlüsselwörter: Poseschätzung, Kinematik, Projektion, Lie-Algebren

Inhaltsverzeichnis

1. Einleitung **3**

2. Mathematische Grundlagen **5**
2.1. Grundlegende mathematische Begriffe 5
2.1.1. Gruppen 5
2.1.2. Algebren 6
2.2. Modelle zu Beschreibung von Rotationen 6
2.2.1. Die spezielle orthogonale Gruppe $SO(3)$ 7
2.2.2. Quaternionen 8
2.2.3. Exponentialdarstellung für Rotationen 9
2.3. Modelle zu Beschreibung affiner Transformationen 11
2.3.1. Die spezielle euklidische Gruppe $SE(3)$ 11
2.3.2. Homogene Koordinaten 12
2.3.3. Exponentialdarstellung für $SE(3)$ 13
2.4. Tangentialmengen 15
2.4.1. Die Tangentialmengen von $SO(3)$ 17
2.4.2. Die Tangentialmengen von $SE(3)$ 17
2.4.3. Eigenschaften der Tangentialmengen 18
2.5. Projektive Geometrie 20
2.5.1. Zentralprojektion 20
2.6. Punkt, Gerade und Ebene als Kern einer linearen Abbildung 24
2.6.1. 2D-Punkt 24
2.6.2. 2D-Gerade 25
2.6.3. 3D-Punkt 25
2.6.4. 3D-Gerade 25
2.6.5. 3D-Ebene 26

3. Problemanalyse **28**
3.1. Der Begriff Poseschätzung 28
3.2. Kinematische Kette 30
3.2.1. Vorwärsgerichtete oder direkte Kinematik 30
3.2.2. Inverse Kinematik 30
3.3. Modellierung der projektiven Abbildung 31
3.3.1. Kollinearität und Koplanarität 34

4. Geometrische Zwangsbedingungen zur Poseschätzung **36**
4.1. Die Punkt zu Punkt Bedingung 37
4.1.1. Punkte 0-ten Grades 37
4.1.2. Punkte 1-ten Grades 40
4.1.3. Punkte g-ten Grades 40
4.2. Die Punkt zu Sehgerade Bedingung 41
4.2.1. Punkte 0-ten Grades 41

4.2.2. Punkte 1-ten Grades 42
4.2.3. Punkte g-ten Grades 43
4.3. Die Punkt zu Sehebene Bedingung 44
4.3.1. Punkte 0-ten Grades 45
4.3.2. Punkte 1-ten Grades 45
4.3.3. Punkte g-ten Grades 45
4.4. Berechnung der Basistransformation 46
4.5. Einschränkung auf planare Bewegungen 47

5. Minimalvoraussetzungen **49**
5.1. Grundvoraussetzungen 49
5.2. Schätzung der Basiskonfiguration 49
5.3. Schätzung der Gelenkstellungen 51

6. Implementation der Poseschätzung **52**
6.1. Interne Datenstruktur 52
6.2. Externe Datenrepräsentation 52
6.2.1. Die Beschreibungssprache VRML97 53
6.3. Die Klassenstruktur der Softwareimplementation 55
6.4. Erstellen und Lösen des Gleichungssystems 56

7. Experimente **57**
7.1. Untersuchung der Kamera 57
7.2. Untersuchung des Konvergenzverhaltens 58
7.3. Poseschätzung mit der Punkt zu Punkt Bedingung 59
7.3.1. Experiment mit simulierten Daten 60
7.3.2. Experiment auf realen Bilddaten 63
7.4. Poseschätzung mit der Punkt zu Sehgerade Bedingung 67
7.4.1. Versuch mit simulierten Daten 67
7.4.2. Versuch auf realen Bilddaten 69
7.5. Roboternavigation mit der Punkt zu Sehebene Bedingung 72
7.6. Ergebnisse 75
7.6.1. Geschwindigkeit 76

8. Diskussion und Ausblick **77**
8.1. Diskussion 77
8.2. Ausblick 80

A. Zeichenerklärung **81**

B. Merkmalsextraktion auf den Bilddaten **82**
B.1. Detektion der Farbmarkierungen 82
B.2. Geradenextraktion mit Hough-Transformation 84

C. Berechnung von Exponentialausdrücken **86**

1. Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Verfolgung geometrischer Objekte und der Adaption von geometrischen Modelldaten an eine reale Szene. Der Schwerpunkt dieser Arbeit liegt dabei auf Objekten, die sich aus mehreren durch Gelenke verbundenen Teilen zusammensetzen. Man spricht hier auch von *kinematischen Ketten* [3, 23].

Anwendungen findet man zum Beispiel in der industriellen Produktion, bei der man auf die visuelle Kontrolle eines Arbeitsprozesses angewiesen ist, oder bei der Filmproduktion. Immer häufiger werden hier computergenerierte Modelle mit realen Bilddaten verknüpft. Sei es für die möglichst natürliche Nachbildung von menschlichen Bewegungen oder zur optimalen Projektion von Computermodellen in die Bildfolge einer realen Szene.

Ein aktuelles Beispiel aus den Medien ist Robert T-Online [33]. Er ist ein sogenannter Avatar, d.h. ein Computermodell, das durch die Bewegung eines realen Schauspielers eine natürlich wirkende Gestik verliehen bekommt.

Weitere Anwendungsgebiete lassen sich in der Sicherheits- und Überwachungstechnik oder der visuell gestützten Roboternavigation finden. Bei vielen dieser Aufgabenstellungen kommt es darauf an, zweidimensionale Merkmale aus Bilddaten einer Kamera zu gewinnen und mit dreidimensionalen Modelldaten abzugleichen. Die Berechnung der dreidimensionalen Modelltransformationen aus zweidimensionalen Sensordaten bezeichnet man auch als *2D-3D Poseschätzung*.

Viele Verfahren zur Poseschätzung gehen von starren Objekten aus [14, 39, 29]. Sollen die Bewegungen eines Menschen oder eines Roboters verfolgt werden, so ist das Modell eines starren Körpers nicht mehr ausreichend. Die Bewegungen, die Teile innerhalb des beobachteten Objektes durchführen, stehen in einem kinematischen Zusammenhang. Wird in diesem Fall die Poseschätzung auf das Schätzen der einzelnen Objekttransformationen zurückgeführt [40], so werden die einschränkenden kinematischen Bedingungen vernachlässigt denen die Einzelbewegungen unterliegen. Dies macht sich besonders bemerkbar, wenn die Poseschätzung mit den Daten einer Videokamera durchgeführt werden soll. Die Bilddaten sind häufig verrauscht und werden stark von der Umgebung (wie z.B. der Beleuchtung) beeinflusst. Gerade in diesem Fall ist es wichtig, möglichst viele vorhandene Informationen auszunutzen und effizient miteinander zu kombinieren. Das hier entwickelte Verfahren nutzt die kinematischen Beziehungen zwischen den Objekten und bietet die Möglichkeit, verschiedene Merkmale aus den Sensordaten in einem linearen Gleichungssystem zusammenzufassen. Die freien Parameter des Modells können dann mit einem Standardlösungsverfahren [32] bestimmt werden.

Schwerpunkt bei der Poseschätzung ist die Ermittlung von Transformationen, die die Position und Lage eines Objektes festlegen. Ein wesentlicher Teil dieser Arbeit beschäftigt sich deshalb mit der mathematischen Modellierung solcher Transformationen, die auch *starre Bewegungen* genannt werden.

Die Ermittlung der Posenparamerter erfolgt aufgrund von Merkmalen, die aus den Bilddaten einer Kamera extrahiert werden. Von der ursprünglich dreidimensionalen Szene liegt in diesem Fall nur eine zweidimensionale Bildinformation vor. Die Bestimmung der dreidimensionalen Pose erfordert eine Rekonstruktion der durch die Projektion verlorengegangenen Informationen. In einem Abschnitt über projektive Geometrie sind deshalb die Grundlagen, die zur Modellierung solcher Projektionen benötigt werden, zusammengefasst.

Der allgemeinen Einführung folgt eine genauere Analyse der Problemstellung. Aufgrund der hier angestellten Überlegungen werden dann in einem weiteren Abschnitt die beiden Konzepte *projektive Geometrie* und *kinematische Kette* miteinander verbunden und die Modellgleichungen für verschiedene geometrische Zwangsbedingungen aufgestellt.

Nach der mathematischen Modellierung werden die Verfahren unter künstlichen und realen Bedingungen untersucht. Da die Versuche auf eine umfangreiche Softwareimplementation angewiesen sind, wird zunächst auf die Implementation der Verfahren eingegangen, deren Entwicklung einen wesentlichen Teil der Arbeit ausmacht. Anschließend erfolgt dann die ausführliche Beschreibung der durchgeführten Experimente und deren Ergebnisse.

In der Diskussion werden noch einmal die Ergebnisse zusammengefasst und der Bezug zu anderen Arbeiten hergestellt.

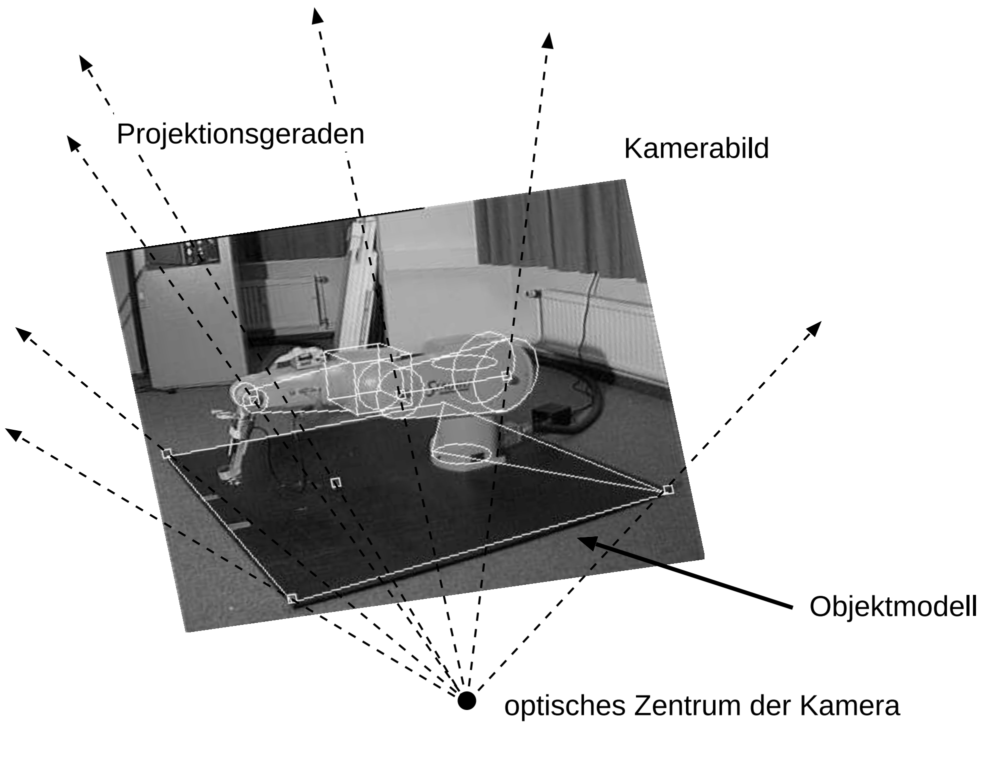


Abbildung 1: Prinzip der Poseschätzung: Anhand von zweidimensionalen Bildmerkmalen (in diesem Fall die Position von Bildpunkten) werden die Posenparameter, d.h die Lage, Orientierung und die Gelenkstellungen eines dreidimensionalen Objektes, ermittelt.

2. Mathematische Grundlagen

Dieser Abschnitt gibt eine Einführung in die mathematischen Grundlagen. Zunächst werden durch einige Definitionen die wichtigsten Grundbegriffe festgelegt. Mit Hilfe dieser Begriffe werden dann verschiedene Möglichkeiten zur Beschreibung von Rotationen vorgestellt. Anschließend wird auf den allgemeinen Fall der starren Bewegungen im Raum eingegangen und entsprechend erweiterte Modelle vorgestellt. Beispiele sollen die Vor- und Nachteile, die bei der Anwendung der entsprechenden Repräsentationen auftreten, verdeutlichen.

Die hier zusammengestellten Grundlagen beschränken sich auf die zum Verständnis wesentlichen Aspekte. Eine detailierte Beschreibung der einzelner Teilgebiete kann in der entsprechenden Literatur [2, 13, 23, 18, 3, 10, 6, 8] nachgelesen werden. Eine Übersicht über die verwendeten Notationen befindet sich im Anhang A.

2.1. Grundlegende mathematische Begriffe

2.1.1. Gruppen

Definition 2.1 (Gruppe)
Eine Gruppe besteht aus einer Menge G und einer Verknüpfung $G \times G \to G$ mit folgenden Eigenschaften:

Assoziativität $\forall a, b, c \in G: \ (ab)c = a(bc)$

Neutrales Element $\exists! 1 \in G \ \forall g \in G: \ 1g = g1 = g$

Inverses Element $\forall g \in G \ \exists! g^{-1} \in G: \ gg^{-1} = 1.$

Beispiel 2.1 (Gruppe)
Für die weiteren Untersuchungen spielt die allgemeine lineare Gruppe der invertierbaren $n \times n$ Matrizen eine wichtige Rolle. Für alle $n \in \mathbb{N}$ wird sie wie folgt definiert:

$$GL(n) := \{n \times n \text{ Matrizen } A \mid det(A) \neq 0\} \tag{2.1}$$

Definition 2.2 (abelsche Gruppe)
Eine abelsche Gruppe ist eine Gruppe deren Verknüpfung zusätzlich kommutativ ist:

Kommutativität $\forall a, b \in G: \ ab = ba$

Beispiel 2.2 (abelsche Gruppen)
$(\mathbb{Z}, +)$ *die Menge der ganzen Zahlen mit Addition*

$(\mathbb{R}, +)$ *die Menge reellen Zahlen mit Addition*

$(\mathbb{R} \setminus \{0\}, \cdot)$ *die Menge reeller Zahlen ungleich Null mit der Multiplikation*

Definition 2.3 (Untergruppe)
Eine Teilmenge U einer Gruppe G heißt Untergruppe, wenn sie folgende Eigenschaften erfüllt:

Abgeschlossenheit $\forall a, b \in U: \ ab \in U$

Neutrales Element $1 \in U$

Inverses Element $\forall a \in U: \ a^{-1} \in U$

2.1.2. Algebren

Definition 2.4 (Algebra)
Eine (assoziative) Algebra über dem Körper[1] $\mathbb{R}$ ist ein $\mathbb{R}$-Vektorraum A zusammen mit einer bilinearen (assoziativen) Abbildung $\mu : A \times A \to A$ (der sogenannten Multiplikation auf A), so dass A mit der Addition des Vektorraumes zu einem Ring[2] wird .

Definition 2.5 (Unteralgebra)
Ein Teilraum U von A, der unter der Multiplikation von A abgeschlossen ist, nennt man Unteralgebra.

Beispiel 2.3
Für jedes $n \in \mathbb{N}$ ist die Menge der reellen $n \times n$ Matrizen mit der üblichen Matrixmultiplikation eine assoziative Algebra über $\mathbb{R}$.

2.2. Modelle zu Beschreibung von Rotationen

Rotationen lassen sich durch spezielle orthogonale Matrizen beschreiben. Orthogonale Matrizen werden durch die Eigenschaft $A^T = A^{-1}$ definiert. Die Spalten- bzw. Zeilenvektoren sind orthonormal zueinander. Äquivalent dazu ist die Formulierung $A^T A = \mathbf{I}$. In Mengenschreibweise sieht die orthogonale Gruppe wie folgt aus,

$$O(n) := \{A \in GL(n) | A^T A = \mathbf{I}\}. \tag{2.2}$$

Die Determinate einer orthogonalen Matrix ist somit auf ± 1 festgelegt.
Beweis: $1 = det(\mathbf{I}) = det(A^T A) = det(A^T) det(A) = det(A)^2 \Rightarrow det(A) = \pm 1$

[1] Eine Algebra lässt sich auf beliebigen Körpern definieren. Im Rahmen dieser Arbeit wird zur Vereinfachung die Einschränkung auf den reellen Zahlenkörper vorgenommen.

[2] Ein Ring hat eine ähnliche Struktur wie ein Körper. Es gibt eine Addition mit Gruppenstruktur und eine Multiplikation. Es existiert aber nicht zu jedem Element $\neq 0$ ein multiplikativ inverses Element.

2.2.1. Die spezielle orthogonale Gruppe $SO(3)$

Die orthogonale Gruppe enthält zwei Untergruppen von $GL(n)$. Diese entstehen durch die Einschränkung auf Matrizen mit Determinate 1 bzw. -1. Im Zusammenhang mit dieser Arbeit ist die sogenannte *spezielle orthogonale Gruppe* wichtig,

$$SO(n) := \{A \in GL(n) | A^T A = \mathbf{I} \text{ und } \det(A) = 1\}. \tag{2.3}$$

Mit Hilfe dieser Matrizen können Rotationen in $\mathbb{R}^2$ und $\mathbb{R}^3$ um den Nullpunkt beschrieben werden. D.h. Matrizen in $SO(2)$ beschreiben Rotationen in der Ebene, und Matrizen in $SO(3)$ beschreiben Rotationen im Raum. Eine Drehung in der Ebene um den Winkel θ lässt sich mit der Matrix

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

beschreiben. Im Dreidimensionalen lassen sich aus dieser Matrix die drei elementaren Matrizen für Rotationen um die Basisvektoren des Vektorraumes herleiten:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Die Multiplikation mit einer dieser Matrizen lässt jeweils eine Koordinate fest und führt um die feststehende Achse eine Drehung gegen den Uhrzeigersinn durch. Soll eine Rotation um eine beliebige Achse[3] durchgeführt werden, so kann man dies durch die Konkatenation der einzelnen Rotationen erreichen. Die Parametrisierung der Rotation durch die Hintereinanderausführung von Rotationen um die Hauptachsen wird Rotation um Eulerwinkel genannt. Da die Multiplikation von Rotationsmatrizen für dreidimensionale Transformationen im allgemeinen nicht kommutativ ist ($SO(3)$ ist im Gegensatz zu $SO(2)$ keine abelsche Gruppe), muß man auf die Reihenfolge der durchgeführten Multiplikationen achten:

$$R_z(\alpha)R_y(\beta)R_z(\gamma) \neq R_z(\gamma)R_y(\beta)R_z(\alpha) \tag{2.4}$$

[3] mit *beliebige Achse* ist hier beliebige Orientierung durch den Ursprung gemeint

Ein Nachteil dieser Beschreibungsform sind die Singularitäten, die sich bemerkbar machen, wenn man von einer gegebenen Rotation

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in SO(3)$$

die Eulerwinkel α, β und γ berechnen möchte. Das folgende Beispiel verdeutlicht die Schwierigkeiten bei der Berechnung von Eulerwinkeln aus der gegebenen Rotation R.

Beispiel 2.4 (Bestimmung von Eulerwinkeln)
Für $\sin(\beta) \neq 0$ lassen sich die Eulerwinkel α, β und γ wie folgt bestimmen:

$$\begin{aligned} \beta &= atan2\left(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}\right) \\ \alpha &= atan2\left(\frac{r_{23}}{\sin(\beta)}, \frac{r_{13}}{\sin(\beta)}\right) \\ \gamma &= atan2\left(\frac{r_{32}}{\sin(\beta)}, \frac{-r_{31}}{\sin(\beta)}\right), \end{aligned} \tag{2.5}$$

$$\textit{wobei } atan2(y,x) := \begin{cases} atan(\frac{y}{x}) & \textit{falls } x > 0 \\ sign(y) \cdot (\pi - atan(|\frac{y}{x}|)) & \textit{falls } x < 0 \\ \mathbf{undef}. & \textit{falls } y = x = 0 \\ sign(y) \cdot \frac{\pi}{2} & \textit{falls } y \neq 0, x = 0 \end{cases} .$$

Das Beispiel zeigt, dass zur Berechnung dieser auf den ersten Blick einfachen Aufgabe viele Fallunterscheidungen notwendig sind. Es ist also sinnvoll, nach einer anderen Parametrisierung zu suchen.

2.2.2. Quaternionen

Eine in der Literatur [38, 23, 8] oft verwendete Beschreibung für Rotationen im dreidimensionalen Raum sind die Quaternionen $\mathbb{H}$. Sie können als eine Erweiterung der komplexen Zahlen aufgefasst werden. Wie bei der Erweiterung von den reellen Zahlen zu den komplexen Zahlen benutzt man drei imaginäre Zahlen (üblicherweise $\mathbf{i}, \mathbf{j}$ und $\mathbf{k}$ genannt). Man erhält so einen 4-dimensionalen Vektorraum der durch die Definition einer Multiplikation die Struktur einer Algebra erhält. Zur Festlegung der Rechenregeln genügt es, die Multiplikation auf den Basiselementen anzugeben,

$$\begin{gathered} a\mathbf{i} = \mathbf{i}a; \qquad a\mathbf{j} = \mathbf{j}a; \qquad a\mathbf{k} = \mathbf{k}a: \qquad a \in \mathbb{R}, \\ \mathbf{ii} = \mathbf{jj} = \mathbf{kk} = \mathbf{ijk} = -1, \\ \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \qquad \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \qquad \mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \end{gathered} \tag{2.6}$$

Aus den Gleichungen der letzen Zeile wird deutlich, dass die Multiplikation nicht kommutativ ist. Die Quaternionen ohne die Null bilden also eine nicht abelsche Gruppe.

Analog zur Konjugation der komplexen Zahlen wird die Konjugation einer Quaternion $Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ durch $Q^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$ beschrieben. Außerdem lässt sich eine Norm durch die Definition

$$||Q||^2 := QQ^* = q_0^2 + q_1^2 + q_2^2 + q_3^2 \tag{2.7}$$

festlegen. Zur Beschreibung von Rotationen werden nur Einheitsquaternionen

$$\mathbb{H}_1 := \{Q \in \mathbb{H} \quad | \quad ||Q|| = 1\} \tag{2.8}$$

benötigt. Diese bilden mit der oben definierten Multiplikation eine Unteralgebra der Quaternionen. Um eine Rotation um den Einheitsvektor $\omega = [\omega_1\ \omega_2\ \omega_3]^T$ mit dem Winkel θ zu beschreiben, wird eine Multiplikation mit der Einheitsquaternion

$$Q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(\omega_1\mathbf{i} + \omega_2\mathbf{j} + \omega_3\mathbf{k})$$

und ihrer Konjugierten durchgeführt. Die Drehung eines Punktes X (kodiert als Quaternion mit einem Realteil von Null) nach X' schreibt sich dann wie folgt,

$$X' = QXQ^*. \tag{2.9}$$

Beispiel 2.5 (Rotationsparameter einer Einheitsquaternion)
Analog zu Beispiel 2.4 soll nun die Rotationsachse ω und der Winkel θ berechnet werden. In diesem Fall wird jedoch davon ausgegangen, dass die Rotation als Einheitsquaternion Q vorliegt. Die Parameter lassen sich nun direkt aus den beiden folgenden Gleichungen berechnen,

$$\theta = 2\cos^{-1} q_0 \qquad \omega = \begin{cases} \frac{1}{\sin(\theta/2)}\,[q_1\ q_2\ q_3]^T & \text{falls } \theta \neq 0 \\ 0 & \text{sonst} \end{cases}. \tag{2.10}$$

An diesem Beispiel wird deutlich, dass durch die Quaternionenrepräsentation eine wesentliche Vereinfachung bei der Bestimmung der Rotationsparameter erreicht werden kann.

2.2.3. Exponentialdarstellung für Rotationen

Eine andere Möglichkeit der Modellierung von Rotationen bietet die Exponentialdarstellung. Hierbei wird nicht von der durchgeführten Bewegung ausgangen, sondern der dynamischen Prozeß der Bewegung genutzt (s. auch [23]). Betrachtet man einen Punkt p auf einer mit Einheitsgeschwindigkeit rotierenden Ebene, so kann der Geschwindigkeitsvektor $\dot{p}$ wie folgt bestimmt werden:

$$\dot{p}(t) = \omega \times p(t), \qquad p(t), \dot{p}(t), \omega \in \mathbb{R}^3 \text{ und } |\omega| = 1. \tag{2.11}$$

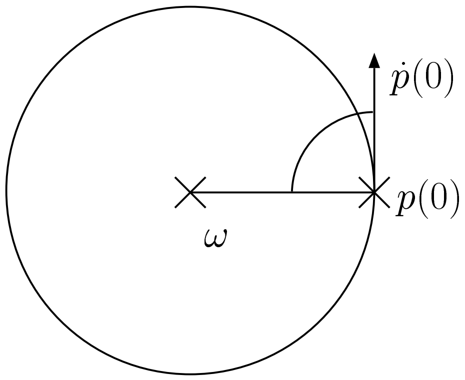


Abbildung 2: Veranschaulichung der Kreisbewegung mit Einheitsgeschwindigkeit; der Richtungsvektor ω zeigt senkrecht aus der Kreisebene; der Geschwindigkeitsvektor $\dot{p}(t)$ steht senkrecht zum Punktvektor $p(t)$ und zum Richtungsvektor ω.

Dabei entspricht ω dem normierten Normalenvektor der Ebene und $p(t)$ den Koordinaten[4] des Punktes p zum Zeitpunkt t. Da das Kreuzprodukt mit einem Vektor ein linearer Operator ist läßt es sich auch als Matrixmultiplikation schreiben,

$$\omega \times p(t) = \hat{\omega} p(t), \qquad \text{wobei } \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{2.12}$$

Durch Integration der Geschwindigkeitsgleichung erhält man einen Ausdruck, der die Position des Punktes p zu einer bestimmten Zeit t bestimmt,

$$p(t) = e^{\hat{\omega} t} p(0). \tag{2.13}$$

Dieser Ausdruck scheint zunächst etwas ungewöhnlich, da im Exponenten die Matrix $\hat{\omega} t$ auftritt. Die Reihenentwicklung der Exponentialfunktion

$$\exp(\mathbf{x}) := \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{x}^n = 1 + \mathbf{x} + \frac{\mathbf{x}^2}{2!} + \frac{\mathbf{x}^3}{3!} + \dots \tag{2.14}$$

wendet aber nur Operationen auf das eingesetzte Element $\mathbf{x}$ an, die auch auf Matrizen anwendbar sind. Weiterhin kann man zeigen [10], dass die Reihe für alle reellen Matrizen in $GL(n)$ konvergiert. Die Exponentialfunktion ist also nicht nur für reelle oder komplexe Zahlen definiert, sondern allgemein für alle reellen Matrizen[5].

Durch die Gleichung (2.13) lässt sich nun eine Rotation um eine Achse ω mit dem Winkel θ beschreiben, indem man den Punkt p für die Zeit $t = \theta$ lang rotiert. Der Grenzwert der Reihe

$$e^{\hat{\omega}\theta} = \sum_{n=0}^{\infty} \frac{1}{n!} (\hat{\omega}\theta)^n = \mathbf{I_{3\times 3}} + \hat{\omega}\theta + \frac{(\hat{\omega}\theta)^2}{2!} + \frac{(\hat{\omega}\theta)^3}{3!} + \dots \tag{2.15}$$

[4] hier sind die Koordinaten im Bezug zum Fixpunkt der Rotation gemeint

[5] Der Definitionsbereich der Exponentialfunktion kann ganz allgemein auf Banach-Algebren mit Einselement ausgeweitet werden. In diesem Fall genügt aber die Beschränkung auf die reellen Matrizen $GL(n)$, die für jedes n eine Banach-Algebra mit der Einheitsmatrix als Einselement bilden.

ist dann eine Matrix aus der Gruppe $SO(3)$. Die Menge der Matrizen, die beim Einsetzen in die Exponentialfunktion eine Matrix in $SO(n)$ erzeugen, werden in der Menge

$$so(n) := \{S \in \mathbb{R}^{n\times n} | S^T = -S\} \tag{2.16}$$

zusammengefasst. Sie entspricht der Tangentialmenge (s. Abschnitt 2.4) der speziellen orthogonalen Gruppe im neutralen Element. Man kann zeigen [23], dass es eine surjektive Abbildung von $so(3)$ nach $SO(3)$ gibt. D.h. für jede Drehung im Raum gibt es einen entsprechenden Repräsentanten in $so(3)$.

2.3. Modelle zu Beschreibung affiner Transformationen

Bisher wurden nur Rotationen um den Ursprung betrachtet. Möchte man Drehungen um beliebige Punkte im Raum zulassen, so müssen die bisherigen Modelle erweitert werden.

2.3.1. Die spezielle euklidische Gruppe $SE(3)$

Die spezielle euklidische Gruppe $SE(n), n \in \mathbb{N}$ vereinigt Rotation und Translation

$$SE(n) := \{(q, R) | q \in \mathbb{R}^n, R \in SO(n)\}. \tag{2.17}$$

Mit Hilfe der Elemente aus $SE(3)$ können nun Koordinatentransformationen durchgeführt werden, bei denen sich auch der Koordinatenursprung verschiebt.

Beispiel 2.6 (Koordinatentransformation)
Möchte man einen Punkt p_b von einem Koordinatensystem B in ein Referenzkoordinatensystem A transformieren, so wendet man die entsprechende Transformation aus $SE(3)$ auf den Punkt wie folgt an:

$$p_a = q_{ab} + R_{ab}p_b, \qquad \textit{mit } (q_{ab}, R_{ab}) \in SE(n). \tag{2.18}$$

Eine Rotation um einen beliebigen Punkt q kann dann durch die folgenden drei Schritte realisiert werden (s. Abb. 3):

i. Translation (Verschiebung) der Punkte um den Drehpunkt q, so dass der Drehpunkt im Koordiantenursprung des Referenzkoordinatensystems liegt.

ii. Rotation der Punkte um den Ursprung (nach der Translation (i) handelt es sich um eine Rotation um den Ursprung)

iii. Translation/Rücktransformation der Punkte, so dass der Drehpunkt wieder in seiner Ausgangslage liegt.

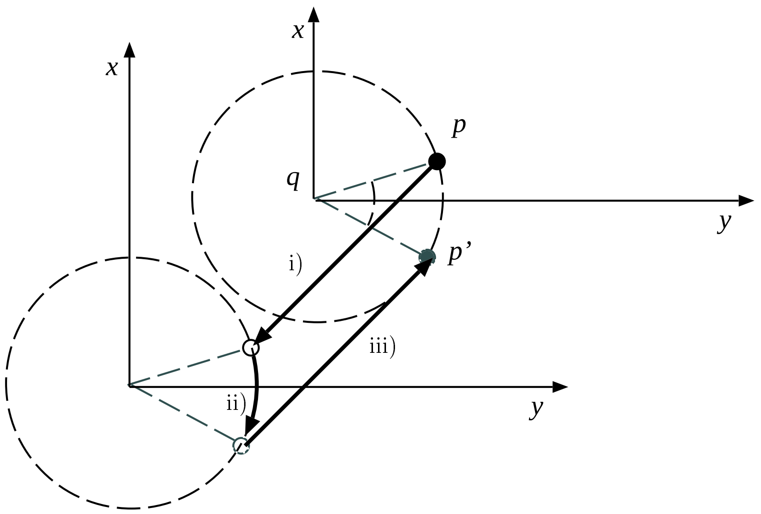


Abbildung 3: Beschreibung der Rotation um einen beliebigen Punkt.

2.3.2. Homogene Koordinaten

Eine gute Möglichkeit Rotationen und Translationen in einer kompakten Schreibweise darzustellen, bieten homogene Koordinaten. Um eine homogene Darstellung eines Punktes im $\mathbb{R}^3$ zu erhalten, erhöht man die Dimension des Vektorraumes (s. hierzu auch die Definition 2.8 des projektiven Raumes).
Einem Punkt $p = [p_1 \quad p_2 \quad p_3]^T$ ist dann der normierte homogene Punkt

$$\bar{p} = [p_1 \quad p_2 \quad p_3 \quad 1]^T \tag{2.19}$$

zugeordnet. Im Folgenden wird für die homogene Darstellung eines Punktes p die Notation $\bar{p}$ verwendet.

Bei homogener Darstellung lassen sich Richtungen von Punkten unterscheiden, da Richtungen am Ende einen Null-Eintrag besitzen. Dies wird deutlich, wenn man den Differenzvektor von zwei homogenen Punkten $\bar{p}$ und $\bar{q}$ betrachtet:

$$\bar{v} = \bar{p} - \bar{q} = [p_1 - q_1 \quad p_2 - q_2 \quad p_3 - q_3 \quad 0]^T . \tag{2.20}$$

Den Vorteil dieser Schreibweise verdeutlicht das folgende Beispiel.

Beispiel 2.7 (Homogene Koordinatentransformation)
Die Koordinatentransformation aus Beispiel 2.6 wird linear und kann durch eine Matrixmultiplikation ausgeführt werden:

$$\bar{p}_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R_{ab}} & q_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_b \\ 1 \end{bmatrix} . \tag{2.21}$$

Wendet man die Transformation jedoch auf eine Richtung an, so wirkt die Koordinatentransformation wie eine Rotation. Dies entspricht der Translationsinvarianz

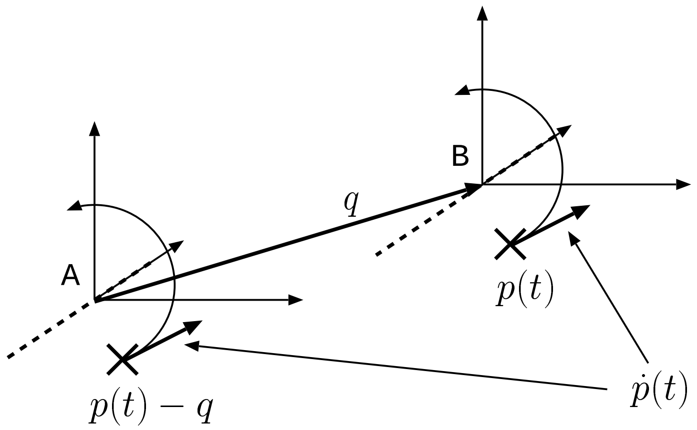


Abbildung 4: Berechnung des Geschwindigkeitsvektors $\dot{p}(t)$ erfolgt im Ursprungskoordinatensystem **A**, d.h. durch Verschiebung des Punktes $p(t)$ um den Vektor q.

von Richtungen

$$\bar{v}_a = \begin{bmatrix} v_a \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R_{ab}} & q_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R_{ab}} v_b \\ 0 \end{bmatrix}. \tag{2.22}$$

2.3.3. Exponentialdarstellung für $SE(3)$

Die Exponentialschreibweise lässt sich auch auf den durch Translation erweiterten Raum $SE(3)$ anwenden. Hierzu betrachtet man wieder die Geschwindigkeit eines Punktes p, der sich mit Einheitsgeschwindigkeit um die Achse ω ($||\omega|| = 1$) bewegt. Nun soll die Rotation nicht um den Ursprung ausgeführt werden, sondern um einen beliebigen Punkt im Raum (Abb. 4). Um die Rotationsachse festzulegen ist noch ein weiterer Punkt q auf der Rotationsachse zu wählen. Der Geschwindigkeitsvektor lässt sich dann wie folgt berechnen:

$$\dot{p}(t) = \omega \times (p(t) - q). \tag{2.23}$$

In homogener Matrixschreibweise sieht der Ausdruck dann so aus:

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & -\omega \times q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \hat{\xi} \begin{bmatrix} p \\ 1 \end{bmatrix}, \qquad \text{mit } \hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times q \\ 0 & 0 \end{bmatrix}. \tag{2.24}$$

Wie schon bei den reinen Rotationen, kann man nun mit Hilfe der Exponentialfunktion die Position des Punktes p zu einem Zeitpunkt t bestimmen:

$$\bar{p}(t) = e^{\hat{\xi} t} \bar{p}(0). \tag{2.25}$$

Die Menge aller Matrizen, die eine solche Transformation darstellen, wird mit $se(3)$ bezeichnet

$$se(3) := \{(v, \hat{\omega}), \hat{\omega} \in so(3), v \in \mathbb{R}^3\}. \tag{2.26}$$

Sie entspricht der Tangentialmenge der speziellen euklidischen Gruppe $SE(3)$ in der Einheitsmatrix (s. Abschnitt 2.4).

Bemerkung 2.1 *Die gleichzeitige Durchführung von Drehung und Translation kann als Schraubenbewegung dargestellt werden, deshalb werden die Elemente von $so(3)$ auch als* **Twist** *(Verwindung) oder* **Screw** *(Schraube) bezeichnet.*

Die Tabelle 1 zeigt eine Übersicht der vorgestellten Modelle. Dabei lassen sich die in einer Spalte aufgeführten Repräsentationen ineinander überführen.

Beschreibungsformen für Rotationen	Beschreibungsformen für die allgemeine starre Bewegung
$SO(3) = \{A \in \mathbb{R}^{3\times3} \mid A^T A = I_{3\times3}, detA = +1\}$ Matrixmultiplikation beschreibt eine Rotation um den Ursprung des Koordiantensystems.	$SE(3) = \{(p, R) \mid p \in \mathbb{R}^3, R \in SO(3)\}$ Matrixmultiplikation beschreibt eine Rotation um eine beliebige Achse im Raum.
$so(3) = \{S \in \mathbb{R}^{3\times3} \mid S^T = -S\}$ Matrixmultiplikation des Exponentials eines Elementes beschreibt eine Rotation um den Ursprung des Koordinatensystems.	$se(3) = \{(v, \hat{\omega} \in so(3)\}$ Matrixmultiplikation des Exponentials eines Elementes beschreibt eine Rotation um eine beliebige Achse im Raum.
$\mathbb{H}_1 = \{Q \in \mathbb{H} \mid \|\|Q\|\| = 1\}$ Quaternionenmultiplikation QXQ^* beschreibt eine Rotation des Punktes $X \in \mathbb{H}$ um den Ursprung des Koordinatensystems.	$\hat{\mathbb{H}} = \{Q_1 + \epsilon Q_2 \mid \epsilon^2 = 0; Q_1, Q_2 \in \mathbb{H}\}$ Dualquaternionen können eine Rotation um eine beliebige Achse im Raum beschreiben.

Tabelle 1: Verschiedene Repräsentationen zur Beschreibung von Rotationen und affinen Transformationen

2.4. Tangentialmengen

In diesem Abschnitt sollen die beiden Mengen $so(3)$ und $se(3)$ etwas genauer betrachtet und wichtige Eigenschaften zusammengefasst werden. Da die beiden Räume spezielle Tangentialmengen der Gruppen $SO(3)$ und $SE(3)$ sind, werden zunächst die allgemeinen Begriffe der Tangentialmenge und des Tangentialvektors definiert. Anschließend wird dann näher auf die Eigenschaften dieser Mengen eingegangen. Insbesondere soll dabei deutlich werden, warum sich gerade die Taylor-Approximation zur 1. Ordnung eignet um die Zwangsbedingungen in Abschnitt 4 zu vereinfachen.

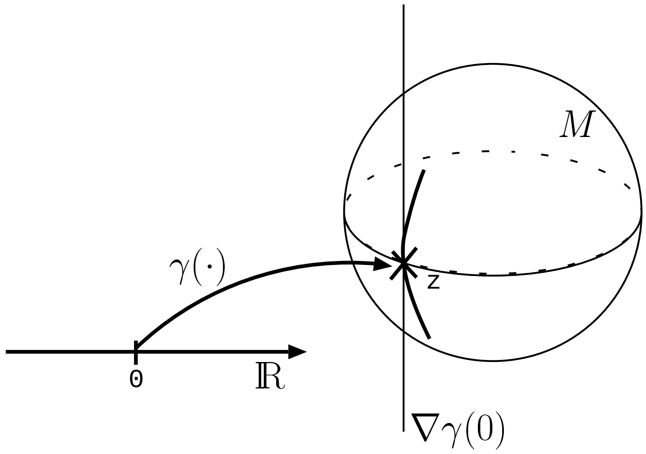


Abbildung 5: Beispiel der Konstruktion einer Tangentialmenge an einer Kugeloberfläche im Punkt z durch Ableitung der Funktionen $\gamma \in \Gamma_z$.

In diesem Abschnitt sei V ein $\mathbb{R}$-Banach-Raum[6], M eine Teilmenge von V und $z \in M$.

Definition 2.6 (Tangentialmenge und Tangentialvektoren)

i. *Mit $\Gamma_z M$ sei die Menge aller V-wertigen Abbildungen γ mit den folgenden Eigenschaften bezeichnet:*

   a) Def (γ) ist ein offenes Intervall in $\mathbb{R}$ um 0

   b) Bild $(\gamma) \subset M$

   c) γ ist in 0 differenzierbar

   d) $\gamma(0) = z$

ii. *Die Menge $T_z M := \{\gamma'(0) | \gamma \in \Gamma_z M\}$ nennt man die Tangentialmenge von M in z. Ihre Elemente heißen Tangentialvektoren von M in z.*

Die Konstruktion von Tangentialmengen wird in Abb. 5 veranschaulicht.

[6] Ein Banach-Raum ist ein vollständiger, normierter Vektorraum. Beispiele sind die Räume $\mathbb{R}^n$ mit der Euklidischen Norm oder $\mathbb{R}^{n\times n}$ mit der Operatornorm $sup\{\frac{Ax}{||x||} | x \in \mathbb{R}^n\}$.

Beispiel 2.8 (Die Tangentialmengen des Einheitskreises)
Als $\mathbb{R}$-Banach-Raum V betrachte man in diesem Fall den 2-dimensionalen Vektorraum $\mathbb{R}^2$ und als Teilmenge M den Einheitskreis $\{v \in \mathbb{R}^2 \mid ||v|| = 1\}$. Für die Tangentialmenge im Punkt $v \in M$ gilt dann:

$$T_v M = \{X \in \mathbb{R}^2 | < v, X >= 0\}$$

Die Tangentialmenge entspricht also dem Senkrechtraum von v.
Beweis: *"$\subseteq$" : Sei $X \in T_v M$ und sei $\gamma \in \Gamma_v M$ mit $X = \gamma'(0)$. Für jedes $t \in Def(\gamma)$ gilt $1 =< \gamma(t), \gamma(t) >= \gamma_1(t)^2 + \gamma_2(t)^2$. Durch Ableiten nach t in 0 erhält man:*

$$0 = \gamma_1'(0) \cdot 2\gamma_1(0) + \gamma_2'(0) \cdot 2\gamma_2(0) = 2 < \gamma(0), \gamma'(0) >= 2 < v, X >$$

"$\supseteq$" : Sei nun $X \in \mathbb{R}^2$ mit $< v, X >= 0$. Dann erhält man mit

$$\gamma : \ ]-1,1[\to M, t \mapsto \frac{v + tX}{||v + tX||}$$

eine in t=0 differenzierbare Funktion. Sie ist in $\Gamma_v M$ enthalten und es gilt $\gamma'(0) = X$. Damit ist $X \in T_v M$ gezeigt.

Satz 2.1
Ist X ein Tangentialvektor von M in v (d.h. $X \in T_v M$), so ist für alle $a \in \mathbb{R}$ auch $aX \in T_v M$.
Beweis: *Sei $X \in T_v M$ und $a \in \mathbb{R}$. Dann ex. ein $\gamma \in \Gamma_z M$ mit $\gamma'(0) = X$. Betrachtet man nun die Abbildung $\gamma_{new}(t) = \gamma(at)$. Dann ist auch $\gamma_{new} \in \Gamma_z M$ und die Ableitung $\gamma_{new}'(0) = aX$.*

Bemerkung 2.2 *Im allgemeinen ist $T_z M$ kein Teilraum von V, da nicht gewährleistet ist, dass die Summe von Tangentialvektoren wieder in der Tangentialmenge liegt. Ist M allerdings eine Untergruppe von $GL(n)$, so zeigt der folgende Satz die Teilraumeigenschaft der Tangentialmengen.*

Satz 2.2
Sei G eine Untergruppe von $GL(n)$. Dann ist für jedes $g \in G$ die Menge $T_g G$ ein Teilraum der Algebra der $n \times n$ Matrizen.
Beweis: *Seien $X, Y \in T_g G$, dann ex. $\gamma, \delta \in \Gamma_g G$, so dass $\gamma'(0) = X$ und $\delta'(0) = Y$. Betrachtet man nun die Abbildung $\beta : t \mapsto \gamma(t) g^{-1} \delta(t)$ auf dem Definitionsbereich Def(γ)$\cap$Def(δ). Dann gilt,*

$$Bild(\beta) \subset Bild(\gamma) g^{-1} Bild(\delta) \subset G,$$

$$\beta(0) = g g^{-1} g = g,$$

$$\beta'(0) \underset{Produktregel}{=} \gamma'(0) g^{-1} \delta(0) + \gamma(0) g^{-1} \delta'(0) = X g^{-1} g + g g^{-1} Y = X + Y.$$

2.4.1. Die Tangentialmengen von $SO(3)$

In diesem Abschnitt soll die Tangentialmenge $so(3)$ der Gruppe $SO(3)$ in der Einheitsmatrix berechnet werden.

Betrachtet man die Matrizen zur Rotation um Eulerwinkel als eine Abbildung von $\mathbb{R}$ nach $SO(3)$, so gilt:

$$R_x(0) = R_y(0) = R_z(0) = I_{3\times 3} \tag{2.27}$$

$$R_x(t), R_y(t), R_z(t) \in SO(3) \text{ für alle } t \in \mathbb{R} \tag{2.28}$$

Damit sind $R_x(\cdot), R_y(\cdot), R_z(\cdot) \in \Gamma_{I_{3\times 3}} SO(3)$. Die Ableitungen der drei Funktionen im Punkt 0 können durch die folgenden drei schiefsymmetrischen Matrizen dargestellt werden:

$$R'_x(0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \; R'_y(0) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \; R'_z(0) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Die Sätze 2.1 und 2.2 liefern nun die übrigen Tangentialvektoren durch die Linearkombination der drei Ableitungen:

$$\omega_x R'_x(0) + \omega_y R'_y(0) + \omega_z R'_z(0) \in so(3), \text{ wobei } \omega_x, \omega_y, \omega_z \in \mathbb{R}.$$

2.4.2. Die Tangentialmengen von $SE(3)$

Nun soll auch die Tangentialmenge $se(3)$ der Gruppe $SE(3)$ in der Einheitsmatrix berechnet werden.

In diesem Fall betrachtet man die erweiterten Matrizen zur Beschreibung von Rotation und Translation als eine Abbildung von $\mathbb{R}$ nach $SE(3)$. Es gibt nun die sechs Grundfunktionen:

$$\overline{R_x}(t) = \begin{bmatrix} R_x(t) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \; \overline{R_y}(t) = \begin{bmatrix} R_y(t) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \; \overline{R_z}(t) = \begin{bmatrix} R_z(t) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

und

$$\overline{T_x}(t) = \begin{bmatrix} I_{3\times 3} & (t\ 0\ 0)^T \\ \mathbf{0} & 1 \end{bmatrix}, \overline{T_y}(t) = \begin{bmatrix} I_{3\times 3} & (0\ t\ 0)^T \\ \mathbf{0} & 1 \end{bmatrix}, \overline{T_z}(t) = \begin{bmatrix} I_{3\times 3} & (0\ 0\ t)^T \\ \mathbf{0} & 1 \end{bmatrix}.$$

Es gilt:

$$\overline{R_x}(0) = \overline{R_y}(0) = \overline{R_z}(0) = \overline{T_x}(0) = \overline{T_y}(0) = \overline{T_z}(0) = I_{4\times 4}. \tag{2.29}$$

Damit sind die Funktionen in $\Gamma_{I_{4\times 4}} SE(3)$, und ihre Ableitungen im Punkt 0 werden durch die folgenden Matrizen dargestellt:

$$\overline{R_x}'(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \overline{R_y}'(0) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \overline{R_z}'(0) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\overline{T_x}'(0) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \overline{T_y}'(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \overline{T_z}'(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Die Sätze 2.1 und 2.2 liefern wieder die übrigen Tangentialvektoren durch Linearkombination:

$$\omega_x \overline{R_x}'(0) + \omega_y \overline{R_y}'(0) + \omega_z \overline{R_z}'(0) + \nu_x \overline{T_x}'(0) + \nu_y \overline{T_y}'(0) + \nu_z \overline{T_z}'(0) \in se(3),$$

wobei $\omega_x, \omega_y, \omega_z, \nu_x, \nu_y, \nu_z \in \mathbb{R}$.

2.4.3. Eigenschaften der Tangentialmengen

Im Folgenden werden einige Eigenschaften der betrachteten Tangentialmengen vorgestellt, die zur Vereinfachung der später durchgeführten Rechnungen beitragen. Außerdem motivieren sie, warum bei der Vereinfachnung der Zwangsbedingungen gerade die Taylor-Approximation zur 1. Ordnung gewählt wird.

Auf den Tangentialmengen der Gruppen lässt sich eine Algebrenstruktur definieren:

Definition 2.7 (Lie-Algebra)
Eine Lie-Algebra über $\mathbb{R}$ ist ein Paar $(L, [\cdot, \cdot\cdot])$, dass sich aus dem reellen Vektorraum L und einer bilinearen Abbildung $[\cdot, \cdot\cdot] : L \times L \to L$ (der sogenannten Lie-Klammer) zusammensetzt. Dabei erfüllt die Lie-Klammer die folgenden Rechenregeln:

Alternation $\forall X \in L \qquad [X, X] = 0$

Jacobi-Identität $\forall X, Y, Z \in L \qquad [X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$

Beispiel 2.9
Mit der Definition $[X, Y] := XY - YX$ bilden die Tangentialmengen $so(3)$ und $se(3)$ Lie-Algebren. Sie werden deshalb auch als die Lie-Algebra von $SO(3)$ bzw. $SE(3)$ bezeichnet.

Diese Arbeit beschäftigt sich mit der Modellierung kinematischer Ketten. Dabei geht es im wesentlichen um die Hintereinanderausführung von Bewegungen im Raum. Schon bei der Beschreibung der Rotationen mit Eulerwinkeln musste jedoch auf die Reihenfolge der durchgeführten Bewegungen geachtet werden. Gerade diese Antikommutativität macht den Umgang mit den Transformationen im Raum schwierig. Für $A, B \in \mathbb{R}^{n \times n}$ mit $AB = BA$ gilt

$$\exp(A)\exp(B) = \exp(A + B). \tag{2.30}$$

Die multiplikative Hintereinanderausführung kommutativer Matrizen kann also in eine Addition im Exponenten überführt werden. Der Zusammenhang mit den Tangentialmengen wird deutlich, wenn die Bedingung $AB = BA$ in $[A, B] = 0$ umgeformt wird. Die Elemente einer Lie-Algebra stehen also im Zusammenhang mit der

Kommutativität der zugrundeliegenden Gruppe. Betrachtet man die Exponentialfunktion wie in Definition 2.6 als Funktion von einem offenen reellen Intervall um Null nach $GL(n)$, dann entspricht die Reihendarstellung der Exponentialfunktion der Taylor-Entwicklung zum Entwicklungspunkt Null. Die Beziehung aus Gleichung 2.30 lässt sich jetzt für beliebige Matrizen formulieren[7]:

Approximation zur 1. Ordnung $t \to 0$:

$$t \mapsto \exp(tA)\ \exp(tB) = \exp(t(A + B + \mathbf{R}_{>1})) \tag{2.31}$$

Approximation zur 2. Ordnung $t \to 0$:

$$t \mapsto \exp(tA)\ \exp(tB) = \exp(t(A + B + \frac{1}{2}[A, B] + \mathbf{R}_{>2})) \tag{2.32}$$

Approximation zur 3. Ordnung $t \to 0$:

$$\begin{aligned} t \mapsto \exp(tA)\ \exp(tB) = \exp(t(A + B + \frac{1}{2}[A, B] \\ + \frac{1}{12}([A, [A, B]] + [B, [B, A]]) + \mathbf{R}_{>3})) \end{aligned} \tag{2.33}$$

Die Abhängigkeit von der Lie-Klammer nimmt also bei der Approximation zu einer höheren Ordnung zu. Zur Vereinfachung der Berechnung kinematischer Ketten eignet sich daher die Approximation zur 1. Ordnung am besten. In diesem Fall kann die nicht kommutative Matrixmultiplikation in die kommutative Addition überführt werden. An dem Experiment in Abschnitt 7.2 wird deutlich, dass die sehr grobe Approximation ausreicht, um fast alle benötigten Transformationen iterativ anzunähern.

Die Eigenschaft, dass in den Tangentialmengen bereits viele Struktureigenschaften einer Gruppe enthalten sind, ist Grundlage der Lie-Theorie und wird in der entsprechenden Literatur, wie [13] und [10] genauer untersucht.

[7] Der Term $\mathbf{R}_{>i}$ entspricht den Restgliedern der Taylorreihe mit einer Ordnung größer als i.

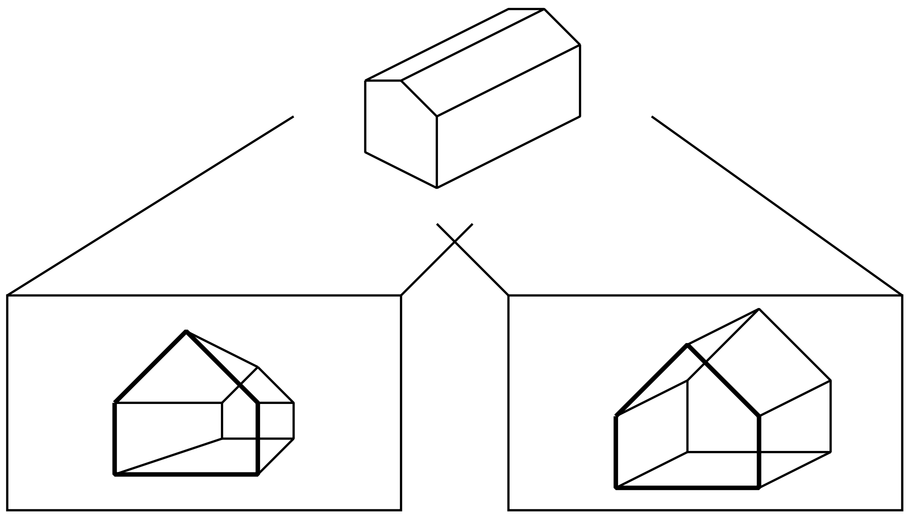


Abbildung 6: Gegenüberstellung: perspektivische und parallele Projektion

2.5. Projektive Geometrie

In diesem Abschnitt wird die Modellierung der geometrischen Abbildungseigenschaft einer Kamera betrachtet. Im wesentlichen bildet eine Kamera den dreidimensionalen Raum auf eine zweidimensionale Bildebene ab. Diese Art von Transformation, bei der eine Dimensionsreduktion durchgeführt wird, nennt man auch *Projektion*.

2.5.1. Zentralprojektion

Das Modell der *Lochkamera* bildet die grundlegenden Eigenschaften einer Kamera nach. Die mit diesem Modell realisierbaren Projektionen lassen sich in zwei Grundklassen einteilen: perspektivische und parallele Projektionen[8]. Zur Modellierung der Aufnahmetechnik einer Kamera ist die perspektivische Projektion besser geeigent. Nur so kann die Größenänderung eines Objektes bei Variation des Abstandes von der Kamera berücksichtigt werden (Abb. 6).

Möchte man einen dreidimensionalen Punkt $p = [x\ y\ z]^T$ auf eine Bildebene mit dem Abstand d von der x-y-Ebene projizieren, so lässt sich dies mit Hilfe der Verhältnisse in den Ähnlichkeitsdreiecken aus Abb. 7 formulieren,

$$\frac{x_{proj}}{d} = \frac{x}{z}, \qquad \frac{y_{proj}}{d} = \frac{y}{z}. \tag{2.34}$$

Um die Abbildung als Matrixmultiplikation darstellen zu können, bietet es sich an, die Aufgabe im projektiven Raum zu formulieren. Eine detailierte Beschreibung der

[8]Die beiden Klassen unterscheiden sich durch die Lage des Projektionszentrums zur Bildebene. Bei der parallelen Projektion ist der Abstand von Projektionszentrum und Bildebene unendlich, d.h. die Projektionsstrahlen sind parallel zueinander.

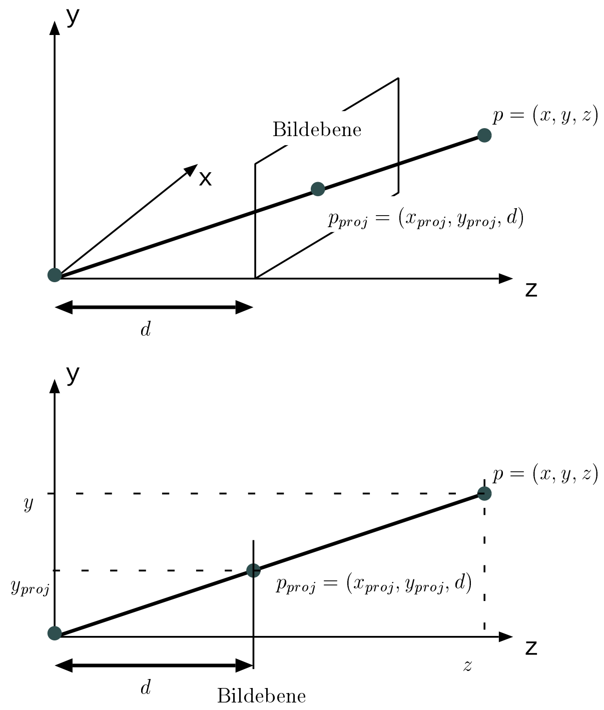


Abbildung 7: Berechnung der perspektivischen Projektion

Modellierung im projektiven Raum ist in dem Buch von Faugeras [6] zu finden. Im folgenden sind die zum Verständnis wesentlichen Eigenschaften zusammengefasst.

Die projektive Geometrie kann als eine Verallgemeinerung der Euklidischen Geometrie angesehen werden. Grundlegend für die projektive Geometrie ist die Beobachtung, dass Punkte, die auf einem Projektionsstrahl liegen, auf einen gemeinsamen Bildpunkt abgebildet werden. Außerdem wird berücksichtigt, dass die Projektionsstrahlen in einem optischen Zentrum zusammenlaufen. Diese Eigenschaften werden durch die Einführung des projektiven Raumes nachgebildet.

Definition 2.8 (Projektiver Raum)
Jeder Punkt $\bar{x} = [x_1\ x_2\ \ldots\ x_{n+1}]^T$ eines n dimensionalen projektiven Raumes $\mathbb{P}$ wird durch einen $n+1$ dimensionalen Vektor repräsentiert, wobei mindestens eine Koordinate ungleich Null ist.

Die $n+1$ Koordinaten eines Punktes im projektiven Raum werden auch homogene Koordinaten genannt. Zwei Vektoren $\bar{x}, \bar{y}$ beschreiben den gleichen Punkt, wenn ein Skalar λ mit $\bar{x} = \lambda\bar{y}$ existiert. Man hat hier die Eigenschaft, dass eine Menge von Vektoren des projektiven Raumes einen einzigen Punkt im euklidischen Raum repräsentiert. Häufig wird dies durch Einführung der Äquivalenzrelation

$$\forall\ \overline{x}, \overline{y} \in \mathbb{P} : \overline{x} \sim \overline{y} \Leftrightarrow \exists \lambda \in \mathbb{R} : \overline{x} = \lambda\overline{y} \tag{2.35}$$

ausgedrückt.

Genau diese Eigenschaft kann man auch bei der Abbildung von dreidimensionalen Punkten auf die Bildebene der Kamera beobachten. Alle Punkte, die auf einem gemeinsamen Sehstrahl liegen, werden auf einen Punkt der Bildebene abgebildet.

Die Projektion eines Punktes nach dem Lochkameramodell kann durch eine Linksmultiplikation mit der 3×4 Matrix

$$M_{per} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.36}$$

beschrieben werden. Mit dieser Matrix werden homogene dreidimensionale Punkte auf die entsprechenden homogenen Punkte der Bildebene transformiert.

Beispiel 2.10 (Projektion nach dem Lochkameramodell)
Multipliziert man den Punkt $\bar{p} = [x\ y\ z\ 1]^T$ mit der Matrix M_{per}, so erhält man einen unnormierten homogenen Punkt

$$M_{per}\bar{p} = \begin{bmatrix} d\,x & d\,y & z \end{bmatrix}^T. \tag{2.37}$$

Dividiert man nun die Einträge durch z[9]*, so ergibt sich der normierte homogene Vektor*

$$\overline{p_{proj}} = \frac{1}{z}(M_{per}\bar{p}) = \begin{bmatrix} \frac{d\,x}{z} & \frac{d\,y}{z} & 1 \end{bmatrix}^T. \tag{2.38}$$

Die Einträge des Vektors entsprechen dann den Koordinaten des projizierten Punktes p.

Bisher wurde die Projektion nur auf eine Bildebene durchgeführt, die parallel zur x-y-Ebene ausgerichtet ist und deren Projektionszentrum im Ursprung des Koordinatensystems liegt. Das Modell lässt sich jedoch recht einfach auf den allgemeinen Fall erweitern, indem der Projektion eine affine Transformation vorgeschaltet wird. Die Parameter dieser Transformation werden auch externe Parameter genannt:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.39}$$

Zur Konvertierung der Weltkoordinateneinheiten in Bildschirm- oder Pixelkoordinaten wird noch eine weitere Transformation zugelassen (s. auch [5]). Man erweitert dazu die Gleichungen 2.34 um Verschiebungs-, Skalierungsparameter:

$$x_{proj} = s_x \frac{x}{z} - s_{xy} \frac{y}{z} + x_0; \qquad y_{proj} = s_y \frac{y}{z} + y_0. \tag{2.40}$$

[9] Die Division durch z entspricht einer Multiplikation mit dem Skalar $\frac{1}{z}$. Der resultierende Vektor beschreibt somit den gleichen projektiven Punkt.

Die erweiterte Projektionsmatrix hat jetzt die Form:

$$P_{scale} = \begin{bmatrix} s_x & s_{xy} & x_0 & 0 \\ 0 & s_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.41}$$

Ihre Parameter werden als interne Parameter bezeichnet, da sie unabhängig von der Position der Kamera sind. Die Multiplikation der beiden Matrizen R und P_{scale} führt zu einer Matrix, die die grundsätzlichen Abbildungseigenschaften einer Kamera nachbildet

$$P = P_{scale}R. \tag{2.42}$$

Linsenverzerrungen bleiben bei dieser Modellierung allerdings unberücksichtigt und führen zu den in Abschnitt 7.1 untersuchten Fehlern. Zur Kompensation radialer Linsenverzerrungen bietet sich das Kalibrierverfahren von Tsai [37] an. Bei diesem Verfahren wird ein Verzerrungskoeffizient κ eingeführt mit dem die Linsenverzerrung weitgehend ausgeglichen wird. Die Berücksichtigung der Linsenverzerrung führt allerdings zu einer nichtlinearen Projektionsabbildung.

In dieser Arbeit wird auf eine Kompensation der Linsenverzerrung verzichtet, da sich die Kalibration dadurch vereinfacht[10].

Zur Bestimmung der freien Parameter der Projektionsmatrix werden die Gleichungen nach den Unbekannten umgestellt (siehe [18] oder [5]). Mit Hilfe eines lineares Ausgleichsverfahren [32] können dann, aus einer Anzahl von 2D-3D Punktkorrespondenzen, die gesuchten Parameter ermittelt werden.

[10]Da die Schätzung der Posenparameter (Abschnitt 4) unabhängig von der Merkmalsextraktion durchgeführt wird, kann eine entsprechende Kompensation der Verzerrungen nachträglich in die Bildvorverarbeitung integriert werden.

2.6. Punkt, Gerade und Ebene als Kern einer linearen Abbildung

In diesem Abschnitt wird gezeigt, wie durch die Ausnutzung des projektiven Raumes Punkte, Gerade und Ebenen in einer einheitlichen Form repräsentiert werden können. Die hier vorgestellte einheitliche Beschreibung wird sich im Folgenden als sehr nützlich erweisen und die Formulierung der in Abschnitt 4 zu entwickelnden Zwangsbedingungen vereinfachen. Grundlage der hier betrachteten Beziehungen ist der Kern oder Nullraum einer linearen Abbildung [7, 2].

Definition 2.9 (Der Kern einer linearen Abbildung)
Als Kern einer linearen Abbildung $L : X \to Y$ von einem Vektorraum X in den Vektorraum Y wird die Menge aller Elemente $x \in X$ bezeichnet mit $L(x) = 0$

$$\mathbf{Kern}\,(L) := \{x \in X | L(x) = 0\}. \tag{2.43}$$

Um Punkte, Geraden und Ebenen als Kerne von linearen Abbildungen beschreiben zu können, muß die Modellierung in dem unter Definition 2.8 eingeführten projektiven Raum erfolgen. Das wird sofort klar, wenn man sich die in der Definition einer linearen Abbildung geforderte Eigenschaft

$$aL(x) = L(ax) \text{ für alle } x \in X, a \in \mathbb{R} \tag{2.44}$$

betrachtet. Soll der Kern von L zum Beispiel eine Gerade beschreiben und p ein Punkt des Euklidischen Raumes auf der Geraden darstellen, so folgt aus $L(p) = 0$ sofort

$$L(\lambda p) = \lambda L(p) = \lambda 0 = 0 \text{ für alle } \lambda \in \mathbb{R}. \tag{2.45}$$

Damit ist $0 \in L$, und jede Gerade, die als Kern einer linearen Abbildung beschrieben wird, verläuft durch den Ursprung. Bettet man den Euklidischen Raum jedoch in den projektiven Raum ein, so lassen sich die drei Entitäten durch den Kern einer linearen Abbildung beschreiben. Zunächst wird der zweidimensionale Raum betrachtet.

2.6.1. 2D-Punkt

Als erstes soll der Punkt $q = [q_x \; q_y]^T$ durch die lineare Abbildung $P_{2D}^{(q)}$ repräsentiert werden. Die Abbildung

$$P_{2D}^{(q)} : \lambda \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \mapsto \lambda \begin{bmatrix} 1 & 0 & -q_x \\ 0 & 1 & -q_y \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \lambda \in \mathbb{R} \setminus \{0\} \tag{2.46}$$

ist eine lineare Abbildung mit der Eigenschaft $\forall \lambda \in \mathbb{R} \setminus \{0\} \quad P_{2D}^{(q)}(\lambda \bar{q}) = 0$. Damit werden alle homogenen Repräsentanten des Punktes q auf Null abgebildet. Die Menge $\mathbf{Kern}\left(P_{2D}^{(q)}\right) \setminus \{0\}$ beschreibt also eine Gerade des zweidimensionalen Euklidischen Raumes.

2.6.2. 2D-Gerade

Eine Gerade in der Ebene kann durch die Gleichung

$$p_x \; cos \; \alpha + p_y \; sin \; \alpha - d = 0 \tag{2.47}$$

beschrieben werden (s. auch Anhang B.2). Die Matrix $[cos \; \alpha \; sin \; \alpha \; - d]$ bildet die Gleichung als Matrixmultiplikation mit einem homogenen Punkt nach, so dass der Kern der Abbildung

$$G_{2D}^{(\alpha,d)} : \lambda \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \mapsto \lambda \begin{bmatrix} cos \; \alpha & sin \; \alpha & -d \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \lambda \in \mathbb{R} \setminus \{0\} \tag{2.48}$$

genau die Repräsentanten des projektiven Raumes enthält, denen Punkte auf der entsprechenden Gerade im Euklidischen Raum zugeordnet sind.

Im Dreidimensionalen gibt es drei geometrische Entitäten, die als Kern einer linearen Abbildung beschrieben werden können.

2.6.3. 3D-Punkt

Der dreidimensionale Punkt $q = (q_x, q_y, q_z)^T$ kann durch die Abbildung

$$P_{3D}^{(q)} : \lambda \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \mapsto \lambda \begin{bmatrix} 1 & 0 & 0 & -q_x \\ 0 & 1 & 0 & -q_y \\ 0 & 0 & 1 & -q_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \lambda \in \mathbb{R} \setminus \{0\} \tag{2.49}$$

dargestellt werden. Es gilt dabei $P_{3D}^{(q)}(\bar{p}) = 0$, wenn der Punkt $\bar{p}$ ein projektiver Repräsentant von q ist.

2.6.4. 3D-Gerade

Eine Gerade im Raum kann durch zwei dreidimensionale Vektoren beschrieben werden. Der erste Vektor $r \in \mathbb{R}^3$ beschreibt die Richtung, der zweite das sogenannte Moment $m \in \mathbb{R}^3$ der Geraden. Das Moment ist durch das Kreuzprodukt

$$m = p \times r \tag{2.50}$$

eines Punktes p auf der Geraden und dem Richtungsvektor r festgelegt. Bei der Berechnung wird meist von einem normalisierten Richtungsvektor ausgegangen, um eine eindeutige Darstellung zu erreichen. Ein Punkt, der auf einer so beschriebenen Geraden liegt, muß dann die Gleichung

$$x \times r - m = 0 \tag{2.51}$$

erfüllen. Die Parametrisierung einer Geraden durch Richtung und Moment wird auch Plücker - Koordinatendarstellung genannt. Dieser Name basiert auf der Vorstellung, eine Gerade als fünfdimensionalen Punkt des sechsdimensionalen homogenen Plücker - Raumes darzustellen. Weitere Informationen zur Plücker - Koordinatendarstellung sind in dem Buch von Blaschke [3] oder in dem Online-Tutorial von Thouis Jones [36] zu finden.

Die Vorteile der Plücker - Koordinatendarstellung einer Geraden resultieren aus der Parameterlosigkeit der Darstellung. Um zu prüfen, ob ein Punkt x auf der Geraden liegt, benötigt die *Punkt - Richtungsform* einen Parameter λ:

$$\exists \lambda \in \mathbb{R} \; x = p + \lambda r, \tag{2.52}$$

wobei p ein Punkt auf der Geraden ist. Benutzt man die implizite Form der Plücker-Koordinatendarstellung, so ist nur die Gleichheit 2.51 nachzuprüfen.

Betrachtet man die Gleichung 2.51 und schreibt das Kreuzprodukt in der aus Abschnitt 2.2.3 bekannten Matrixform, dann erhält man durch anschließendes Homogenisieren die lineare Abbildung:

$$G_{3D}^{(r,m)} : \lambda \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \mapsto \lambda \begin{bmatrix} 0 & r_z & -r_y & -m_x \\ -r_z & 0 & r_x & -m_y \\ r_y & -r_x & 0 & -m_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \lambda \in \mathbb{R} \setminus \{0\}. \tag{2.53}$$

Da die Abbildung so konstruiert ist, dass sie alle homogenisierten Punkte der Geraden auf Null abbildet, entspricht die Abbildung der gesuchten Geradenbeschreibung. Es gilt nämlich genau dann

$$G_{3D}^{(r,m)}(\bar{p}) = 0, \tag{2.54}$$

wenn p auf der Geraden liegt.

Bemerkung 2.3 *Die Matrixdarstellung einer Geraden stimmt (bis auf Vorzeichen) mit der eines Twists aus se(3) überein, da die Ableitung im Raum $SE(3)$ einer linearen Abbildung der Form 2.53 entspricht.*

2.6.5. 3D-Ebene

Auch Ebenen können in einer impliziten Form beschrieben werden. Sie wird Hesseform genannt und legt die Ebene durch Normalenvektor n und einen Skalar δ_h (Hessedistanz) fest. Um festzustellen, ob ein Punkt p auf der Geraden liegt, ist die Gleichheit

$$n \cdot p - \delta_h = 0 \tag{2.55}$$

zu prüfen.

Daraus lässt sich die folgende Beschreibung der Ebene als Kern einer linearen Abbildung herleiten:

$$E_{3D}^{(n,\delta_h)} : \lambda \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \mapsto \lambda \begin{bmatrix} n_x & n_y & n_z & -\delta_h \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \lambda \in \mathbb{R} \setminus \{0\}. \tag{2.56}$$

Es gilt $E_{3D}^{(n,\delta_h)}(\bar{p}) = 0$ für jeden Punkt p der Ebene.

3. Problemanalyse

In diesem Abschnitt soll die Aufgabe dieser Arbeit genauer spezifiziert werden. Dabei wird zunächst auf den Begriff der Poseschätzung eingegangen und verschiedene Ansätze vorgestellt. Anschließend werden dann die in dieser Arbeit geforderten Voraussetzungen und Ziele festgelegt und auf die daraus resultierenden Probleme eingegangen. Zu den auftretenden Problemen werden Lösungsansätze aus der Literatur besprochen und der in dieser Arbeit gewählte Ansatz begründet.

3.1. Der Begriff Poseschätzung

Die Berechnung der Lage und Orientierung eines Objektes bezeichnet man als Poseschätzung. Im wesentlichen lassen sich drei verschiedene Ansätze unterscheiden. Bei der 2D-2D Poseschätzung geht es um die Bestimmung der Lage und Orientierung eines Objektes in der zweidimensionalen Ebene. Zur Bestimmung der Parameter liegen zweidimensionale Informationen vor.

Eine Erweiterung ist die 3D-3D Poseschätzung. Hier wird auf Grund von dreidimensionalen Daten die Lage und Orientierung eines 3D-Objektes im 3D-Raum bestimmt. Die Schwierigkeiten bei dieser Aufgabe ist die mit der höheren Dimension verbundene Anzahl von freien Parametern. Eine Koordinatentransformation im Raum hat sechs freie Parameter, in der Ebene sind hingegen nur drei freie Parameter zu bestimmmen.

Diese Arbeit beschäftigt sich im wesentlichen mit der 2D-3D Poseschätzung. Es geht dabei um die Bestimmung der Lage und Orientierung eines Objektes im dreidimensionalen Raum aus den zweidimensionalen Informationen einer Kamera. Da mit der Dimensionsreduktion durch die Kamera ein Informationsverlust verbunden ist, kann man nicht erwarten, dass sich die dreidimensionalen Daten ohne zusätzliche Information rekonstruieren lassen. Benutzt man jedoch ein kalibriertes Stereosystem oder greift auf geometrische Informationen über das beobachtete Objekt zurück, so lassen sich dreidimensionale Informationen zurückgewinnen.

Viele Arbeiten, die sich mit dem Problem der 2D-3D Poseschätzung befassen und auf die geometrischen Beziehungen eines dreidimensionalen Objektes zurückgreifen, gehen von einem starren Körper aus. Die Abstände und Winkel innerhalb des Objektes werden somit als konstant vorausgesetzt. Die dreidimensionalen Informationen können dann, durch Ausnutzung dieser Invarianten, aus den geometrischen Verhältnissen der Merkmale im zweidimensionalen Bild rekonstruiert werden. Anwendung hierzu sind zum Beispiel in [14, 39, 29] zu finden.

In dieser Arbeit soll das Modell eines starren Körpers erweitert werden, so dass die Poseschätzung von dreidimensionalen Objekten mit internen kinematischen Abhängigkeiten möglich ist. Damit sind Objekte gemeint, die sich aus mehreren Teilobjekten (im folgenden Glieder genannt) zusammensetzen und durch Gelenke miteinander verbunden sind. Unter den freien Parametern bei der Poseschätzung sind somit nicht nur Lage und Orientierung des Objektes zu verstehen, sondern auch Werte für die Stellung der internen Gelenke (Abb. 8).

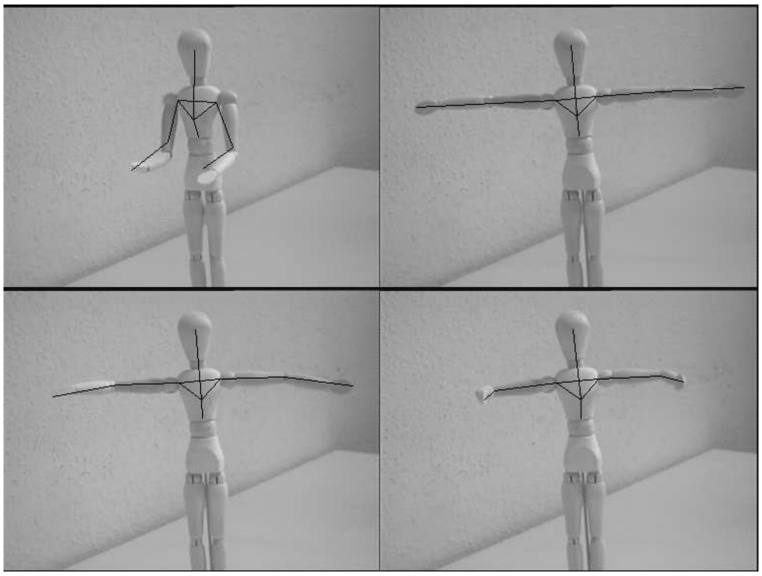


Abbildung 8: Beispiel für die 2D-3D Poseschätzung: Die Bilder zeigen die Berechnung der Pose für das dreidimensionale Modell einer Holzpuppe. Die Schätzung von Lage, Orientierung und Gelenkstellung erfolgte auf Grundlage von manuell vorgegebenen 2D-3D Punktkorrespondenzen.

Um eine solche Parameterschätzung durchzuführen, gibt es verschiedene Ansätze. Weik und Liedtke stellen in [40] ein hierarchisches System vor. Hier wird zunächst die Lage und Orientierung des Basiskörpers bestimmt, um dann auf dieser Grundlage Schritt für Schritt die Stellung der einzelnen Gelenke zu berechnen.

Benutzt man einen solchen hierarchischen Ansatz, lassen sich aber nur schwer die internen Abhängigkeiten nachbilden. Die Lage und Orientierung eines Körperteils am Ende einer Gelenkkette beschränkt schließlich auch die möglichen Konfigurationen des Basiskörpers. Diese Abhängigkeit ist bei einem hierarchischem Ansatz explizit zu formulieren.

Die vorliegende Arbeit basiert deshalb auf einem Ansatz, der von Bregler und Malik in [4] verwendet wird. Dort werden die externen und internen Bewegungen mit Hilfe einer Exponentialabbildung beschrieben. Durch Approximation wird dann ein lineares Gleichungssystem erstellt, dass die gesamte Kinematik des Objektes infinitesimal beschreibt. Dieses kann durch ein lineares Ausgleichsproblem gelöst werden.

Bregler und Malik benutzen in ihrer Arbeit jedoch den optischen Fluss und beschränken sich auf die orthographische Projektion, um die Gleichungen linearisieren zu können. Dies hat den Nachteil, dass sich die beobachteten Objekte nur bis zu einer Tiefe ausdehnen dürfen, in der die perspektivischen Verzerrungen vernachlässigt werden können. Außerdem lassen sich durch die Verwendung des optischen Flusses nur relativ kleine Bewegungen verfolgen.

Das hier zu untersuchende Problem der 2D-3D Poseschätzung setzt sich aus der Modellierung kinematischer Ketten und der Nachbildung der Abbildungseigenschaften einer Kamera zusammen. In den beiden folgenden Abschnitten sollen deshalb die beiden Teilaspekte genauer betrachtet werden.

3.2. Kinematische Kette

Unter einer kinematischen Kette versteht man ein von mehreren starren Körpern gebildetes System. Dabei sind die einzelnen Körper durch Gelenke miteinander verbunden. Die Bewegungen einzelner Punkte des Körpers unterliegen also den kinematischen Gesetzen des vorgegebenen Systems. Bei der Berechnung solcher Systeme wird zwischen der direkten und der inversen Kinematik unterschieden.

3.2.1. Vorwärsgerichtete oder direkte Kinematik

In diesem Fall ist die Stellung der Gelenke bekannt, und es soll die Position der Körper berechnet werden. Die Berechnungen zur Bestimmung der direkten Kinematik lassen sich relativ einfach durchführen. Das folgende Beispiel demonstriert die Berechnung einer kinematischen Kette, wobei die Gelenke durch Twists (s. Abschnitt 2.3.3) parametrisiert sind.

Beispiel 3.1 (Berechnung der direkten Kinematik)
Gegeben sei ein System mit n+1 starren Körpern, die durch Gelenke $\xi_1, \xi_2, \ldots \xi_n$ verbunden sind (s. Abb. 15 auf Seite 37). Weiterhin sei die Stellung der Gelenke durch die Parameter $\theta_1, \theta_2, \ldots \theta_n$ festgelegt. Möchte man nun, ausgehend von der initialen Konfiguration $(\theta_1, \theta_2, \ldots \theta_n = 0)$ des Systems, die Position p' eines Punktes p bestimmen, so lässt sich diese auf folgende Weise berechnen:

$$\begin{aligned}
p' &= p && \textit{für Punkte im Grundkörper}\\
p' &= e^{\hat{\xi}_1\theta_1}p && \textit{für Punkte nach dem 1. Gelenk}\\
p' &= e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}p && \textit{für Punkte nach dem 2. Gelenk}\\
p' &= e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}\ldots e^{\hat{\xi}_i\theta_i}p && \textit{für Punkte nach dem i-ten Gelenk}
\end{aligned}$$

Die Exponentialausdrücke entsprechen dabei Transformationsmatrizen (s. Abschnitt C), die eine starre Bewegung beschreiben. Die eigentliche Berechnung der Ausdrücke kann somit durch Matrixmultiplikationen durchgeführt werden.

3.2.2. Inverse Kinematik

Anders sieht es bei Berechnung der inversen Kinematik aus. Möchte man die Gelenkstellungen berechnen, die zu einer Verschiebung der Punkte im Objekt geführt haben oder sollen, so stößt man auf einige Probleme (s. auch Beispiel 2.4). Durch die Hintereinanderschaltung mehrerer Gelenke erhält man nicht lineare und nicht eindeutig lösbare Gleichungssysteme, d.h. die gleiche Position eines Punktes kann durch verschiedene Konfigurationen des Systems erreicht werden (Abb. 9). In der

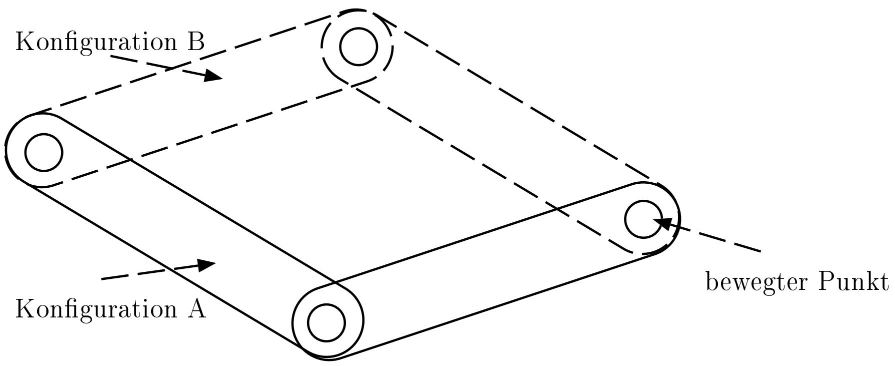


Abbildung 9: Mehrdeutigkeiten erschweren die Bestimmung der freien Parameter bei der inversen Kinematik

Robotik werden die Gleichungen zur Berechnung der Bewegungsparameter oft für einen ganz bestimmten Robotertyp aufgestellt. Man kann in diesem Fall spezielle Eingenschaften des Manipulators ausnutzen und die Aufgabenstellung in einfachere Teilprobleme unterteilen. Diese Methode ist auch unter dem Namen Paden-Kahan Zerlegung bekannt [23].

Sollen ganz allgemeine kinematische Ketten berechnen werden, so lässt sich diese Aufteilung in Teilprobleme nicht mehr so einfach durchführen. Schwierigkeiten treten zum Beispiel dann auf, wenn sich die Rotationsachsen nicht schneiden. In diesem Fall existiert kein Fixpunkt mit dem sich die Berechnungen vereinfachen lassen.

Die explizite Lösung führt im allgemeinen zu Gleichungssystemen, die sich aus trigonometrischen Polynomen mit hohem Grad zusammensetzen. Bis heute ist aber kein stabiles und zugleich schnelles Verfahren bekannt um dieses Problem zu lösen. Einen Ansatz zur Lösung solcher Gleichungssysteme findet man zum Beispiel in [22].

In dieser Arbeit wird deshalb ein iteratives Vefahren zur Berechnung der inversen Kinematik gewählt. Da bei der Analyse von Bildfolgen oft nur kleine Bewegungen auftreten, genügt es, die Bewegung lokal zu approximieren und sich dann iterativ an die real ausgeführte Bewegung anzunähern. Die Abbildung 10 zeigt das Prinzip des Verfahrens in der Ebene. Insbesondere bietet sich in diesem Fall die Modellierung der Bewegung in Exponentialdarstellung (d.h. in der mit der Bewegungsgruppe verbundenen Lie-Algebra se(3)) an, da so eine direkte Verbindung zu den Tangentialmengen der Bewegungsgruppe hergestellt wird (s. auch Abschnitt 2.4 über Tangentialmengen).

3.3. Modellierung der projektiven Abbildung

Im Abschnitt über die mathematischen Grundlagen zur projektiven Geometrie wurde im wesentlichen auf die Projektion von einem höherdimensionalen Raum in einen niedriger Dimension eingegangen. Bei der Poseschätzung handelt es sich um ein

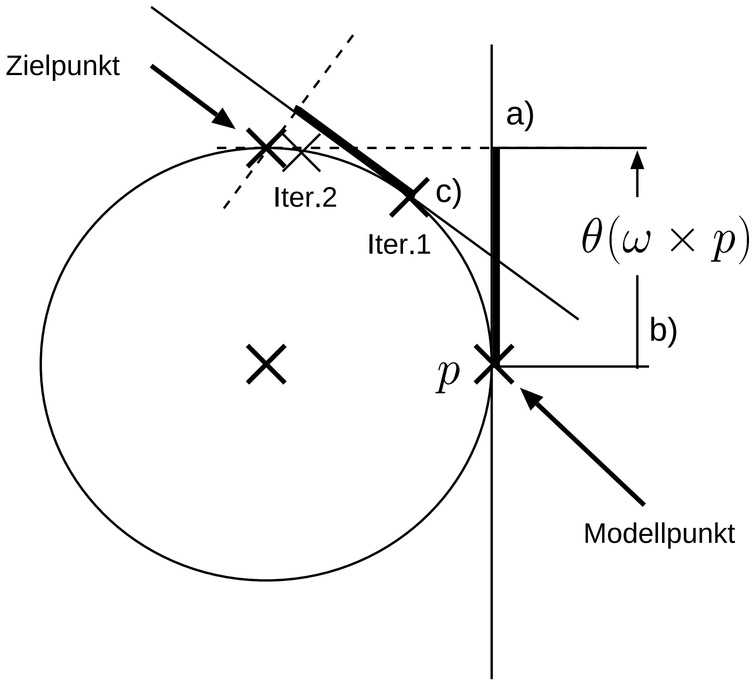


Abbildung 10: Berechnung der Kinematik durch iterative Annäherung des Modellpunktes an den Zielpunkt:
a) Bestimmung des minimalen Abstandes von Zielpunkt und Tangentialgerade;
b) Bestimmung des Winkels (Der Skalierungsfaktor θ des Richtungsvektors $\omega \times p$ der Tangentialgerade entspricht dem Winkel im Bogenmaß);
c) Berechnung der neuen Pose für den nächsten Iterationsschritt

beobachtendes System. In diesem Fall geht es nicht mehr um eine Abbildung die Punkte auf Punkte abbildet, sondern um eine Relation, die einem Punkt in der Projektionsebene eine Menge von Punkten zuordnet. Man kann dies auch als Abbildung von Entitäten unterschiedlicher Art auffassen. Einem Punkt in der Bildebene ist zum Beispiel eindeutig eine Sehgerade zugeordnet. Genauso ist einer Geraden in der Bildebene eine Sehebene zugeordnet (s. Abb. 11).

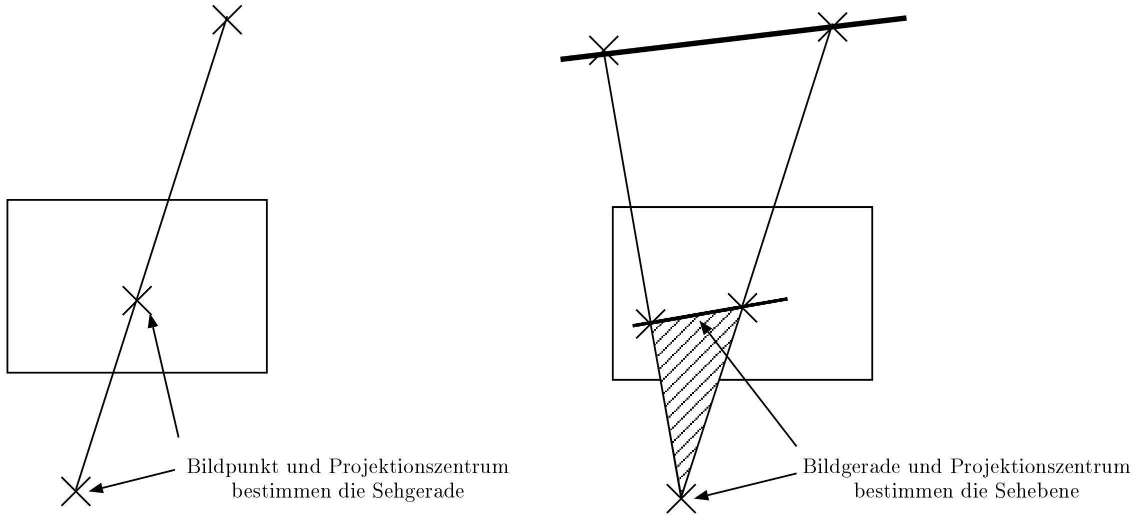


Abbildung 11: Objekten in der Bildebene sind durch die Projektion höherdimensionale Objekte im Raum zugeordnet.

Abhängig von den beiden Betrachtungsweisen ergeben sich zwei verschiedene Vorgehensweisen:

1) Projektion des *dreidimensionalen* Modells in die Bildebene und abgleichen des *projizierten* Modells mit den extrahierten *zweidimensionalen* Bildmerkmalen, oder

2) Berechnung *dreidimensionaler* Entitäten aus den *zweidimensionalen* Bildmerkmalen und abgleichen des *dreidimensionalen* Modells mit den *dreidimensionalen* Entitäten.

Bei der Entscheidung für eine Vorgehensweise muß berücksichtigt werden, dass die ermittelten Daten nicht exakt sind. Ungenauigkeiten bei der Detektion, Fehler bei der Kalibration der Kamera oder Abweichungen der Modellgeometrie vom realen Objekt haben signifikaten Einfluß auf die Berechnung der Pose.

Es ist also wichtig, ein Verfahren zu wählen, das mit Abweichungen umgehen kann und ein Maß für die Güte einer Schätzung mitbringt. Denn ohne die Möglichkeit Abweichungen von Modell- und Messdaten quantisieren zu können ist die Schätzung der Bewegungsparameter nicht durchzuführen.

In dieser Arbeit wurde daher die zweite Vorgehensweise gewählt. Die Berechnung der dreidimensionalen Entitäten kann in diesem Fall bereits vor der eigentlichen Poseschätzung durchgeführt werden. Die Optimierungsaufgabe wird dadurch

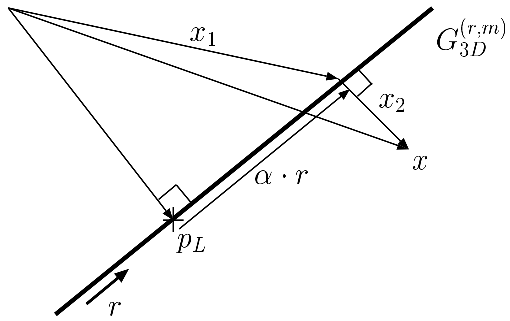


Abbildung 12: Parametriesierung einer Geraden durch Richtung r und Moment m

unabhängig von der projektiven Abbildung. Einerseits wird so Rechenzeit eingespart, da die zur Berechnung der Kinematik benötigen Iterationen unabhängig von der Projektion werden. Andererseits kann zur Optimierung das Prinzip von *Kollinearität* und *Koplanarität* ausgenutzt werden, so dass ein geeignetes Abstandsmaß zur Optimierung genutzt werden kann.

3.3.1. Kollinearität und Koplanarität

In dieser Arbeit wird als Fehlermaß der minimale Euklidische Abstand eines Punktes vom korrespondierenden geometrischen Objekt gewählt. Dieses Maß ist eindeutig, da es sich bei den hier betrachteten Objekten, wie Punkt, Linie und Gerade, um Objekte mit konvexen Punktmengen handelt. Zur Berechnung bieten sich deshalb die impliziten Darstellungen von Gerade und Ebene (s. Abschnitt 2.6) an, da in dieser Beschreibungsform bereits die Abstände zur Festlegung des Objektes genutzt werden (siehe auch [27] und [36]).

Verwendet man die Pückerdarstellung einer Gerade (s. Abschnitt 2.6.4), so ist sie durch den Einheitsvektor r für die Richtung und den Momentvektor m eindeutig bestimmt,

$$x \times r - m = 0, \qquad \text{falls } x \text{ Element der Geraden } G_{3D}^{(r,m)} \text{ ist.} \tag{3.57}$$

Diese implizite Darstellung der Geraden enthält bereits einen Term, der den minimalen Abstand (senkrechter Abstand) eines Punktes zur Geraden repräsentiert. Liegt der Punkt x nicht auf der Geraden $G_{3D}^{(r,m)}$, dann ergibt sich die in Abbildung 12 dargestellte Situation. In Formeln geschrieben bedeutet das:

$$\begin{aligned} x \times r - m &= (x1 + x2) \times r - m \\ &= (p_L + \alpha\, r + x2) \times r - m \\ &= p_L \times r + \alpha\, r \times r + x2 \times r - m \\ &= m + 0 + x_2 \times r - m = x_2. \end{aligned} \tag{3.58}$$

Auf der rechten Seite der Gleichung ergibt sich also ein Vektor x_2. Er steht senkrecht auf der Geraden und sein Betrag $|x_2|$ entspricht dem minimalen Euklidischen

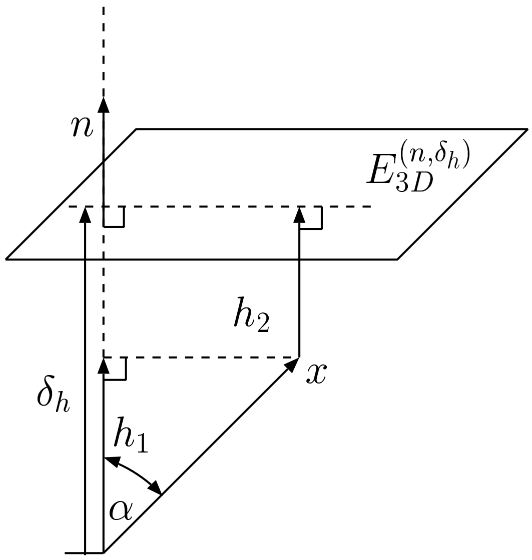


Abbildung 13: Parametriesierung einer Ebene durch Normalenvektor n und Hessedistanz δ_h

Abstand des Punktes x zur Geraden. Im projektiven Raum entspricht das dem Ausdruck:

$$G_{3D}^{(r,m)}(\bar{x}) = x_2. \tag{3.59}$$

Zur Bestimmung des minimalen Abstandes eines Punktes zu einer Ebene kann die ebenfalls implizit definierte Hesse-Form aus Abschnitt 2.6.5 benutzen werden. Ist n der Normalenvektor und p ein Punkt auf der Ebene, dann entspricht die Hessedistanz δ_h dem Vektorprodukt $\delta_h = n \cdot p$.

Wird ein Punkt x außerhalb der Ebene in die Gleichung eingesetzt, dann ergibt sich ein Skalar h_2, dessen Absolutbetrag gerade dem Minimalabstand des Punktes x von der Ebene entspricht. Die Verhältnisse sind in der Abbildung 13 wiedergegeben und in der Gleichung 3.60 formal zusammengefasst.

$$n \cdot x - \delta_h = |n| \cdot |x| \cdot \cos(\alpha) - \delta_h = h1 - \delta_h = h_2 \tag{3.60}$$

In der in Abschnitt 2.6.5 eingeführten Schreibweise gilt somit:

$$E_{3D}^{(n,\delta_h)}(\bar{x}) = h_2. \tag{3.61}$$

4. Geometrische Zwangsbedingungen zur Poseschätzung

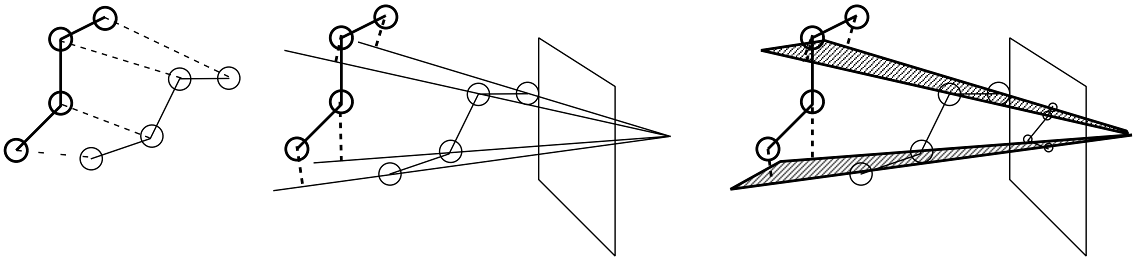


Abbildung 14: Die drei Zwangsbedingungen im Überblick:
a) Punkt zu Punkt zur 3D-3D Poseschätzung;
b+c) Punkt zu Sehgerade (Sehebene) zur 2D-3D Poseschätzung;
Die verstärkt eingezeichnete Figur stellt das Referenzmodell dar. Die dünne Figur deutet die vorgegebene Pose an.

In diesem Abschnitt werden geometrische Zwangsbedingungen entwickelt, die die Verbindung zwischen den ermittelten Bilddaten und dem dreidimensionalen kinematischen Modell herstellen. Wie in [27] wird davon ausgegangen, dass man in den Bilddaten einer kalibrierten Kamera Punkte und Geraden detektieren kann, die mit einem dreidimensionalen Modellpunkt korrespondieren. Die Grundidee dabei ist, dass die zu einem Modellpunkt korrespondierenden Merkmale auf einer geometrischen Einheit höherer Ordnung liegen (Abb. 11). Einem Punkt im Videobild einer kalibrierten Kamera ist zum Beispiel ein Sehstrahl zugeordnet, auf dem der transformierte Modellpunkt liegen muß. Ein weiteres Beispiel ist eine Gerade, die man z.B. durch eine Hough-Transformation erhalten kann. Ihr ist eine Sehebene zugeordnet auf der sowohl die Gerade selbst, als auch ein ihr zugeordneter transformierter Modellpunkt liegt.

Im Folgenden werden die im Bild detektierten Merkmale (d.h. Punkte und Graden) nur noch durch ihre höheren Einheiten repräsentiert. Betrachtet man auch den Fall direkt dreidimensionale Punkte in Beziehung zu setzen (ohne Projektion), so erhält man drei verschiedene geometrische Zwangsbedingungen (s. auch Abb. 14):

a) Punkt - Punkt: setzt zwei 3D-Punkte in Beziehung

b) Punkt - Sehgerade: (Kollinearität) setzt 3D-Modellpunkt und 3D-Gerade in Beziehung

c) Punkt - Sehebene: (Koplanarität) setzt 3D-Modellpunkt und 3D-Ebene in Beziehung

In den folgenden Abschnitten werden die aus den drei Fällen resultierenden Gleichungen hergeleitet und genauer untersucht. Bei den Berechnungen wird die in Abbildung 15 veranschaulichte Notation verwendet.

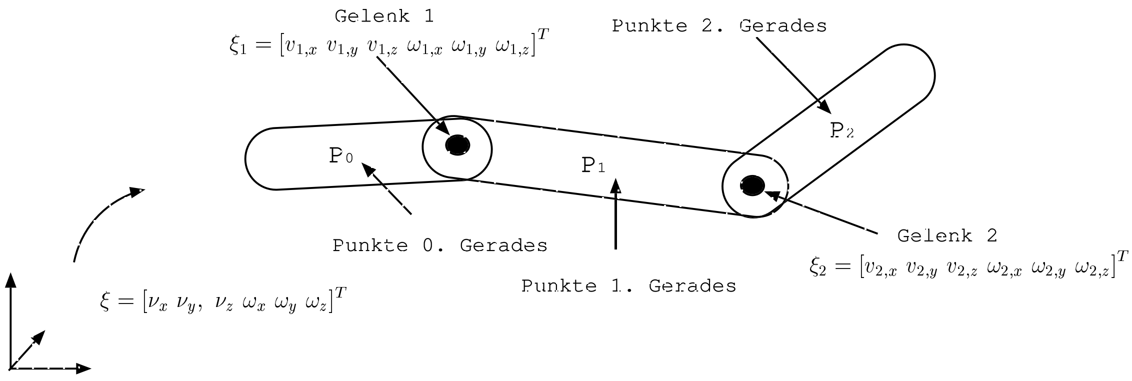


Abbildung 15: Zeigt die bei den Berechnungen verwendete Notation.

4.1. Die Punkt zu Punkt Bedingung

Die Punkt zu Punkt Bedingung beschreibt die Verbindung zwischen zwei dreidimensionalen Punkten. Diese Bedingungen werden benötigt, wenn keine perspektivische Projektion benutzt wird, bzw. aus anderen Vorverarbeitungsschritten bereits dreidimensionale Punktkoordinaten vorliegen. Als Eingabedaten werden in diesem Fall die dreidimensionalen Koordinaten des zu verfolgenden Objektes und die korrespondierenden Modelldaten (d.h. Punkte und die Lage der Gelenkachsen) benötigt. Als Ergebnis sollen die Bewegungsparameter geliefert werden. Im folgenden werden nun die Gleichungen nach dem Grad des Punktes, d.h. nach der Abhängigkeit eines Punktes von den Gelenken, untersucht.

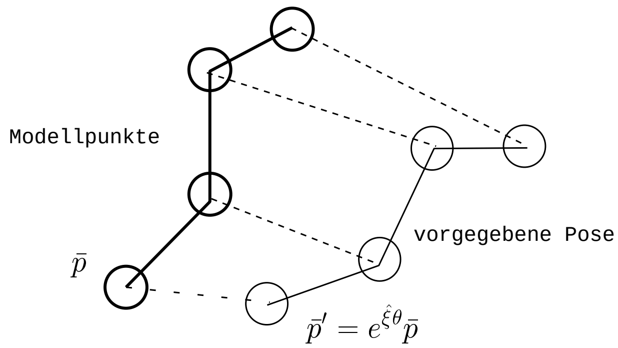


Abbildung 16: Die Punkt zu Punkt Zwangsbedingung: 3D Punkt aus dem Referenzmodell und der vorgegebenen 3D Szene werden durch die Exponentialabbildung zueinander in Beziehung gesetzt.

4.1.1. Punkte 0-ten Grades

In diesem Fall sind die Punkte nur von der Grundposition des Objektes abhängig. Da es keine feste Drehachse oder andere einschränkenden Bedingungen für die Bewegung der Modellpunkte gibt, müssen in diesem Fall alle Parameter einer starren Bewegung berechnet werden. Ist $p = [p_x\ p_y\ p_z]^T$ ein Modellpunkt 0-ten Grades (d.h. von keinem internen Gelenk abhängig), $p' = [p'_x\ p'_y\ p'_z]^T$ der zu verfolgende Punkt und $\xi\theta$ der gesuchte Twist, der die starre Bewegung beschreibt. Dann lässt sich die Zwangsbedingung wie folgt formulieren:

$$\bar{p'} = e^{\hat{\xi}\theta}\bar{p} \tag{4.62}$$

In der aus Abschnitt 2.6 bekannten Form lautet die Bedingung:

$$P_{3D}^{(p')}\left(e^{\hat{\xi}\theta}\bar{p}\right) = 0, \tag{4.63}$$

oder in Worten:

Der Modellpunkt wird durch die Transformation $e^{\hat{\xi}\theta}$ auf den Zielpunkt p' abgebildet.

Die Exponentialfunktion ist nicht linear und die Bestimmung der gesuchten Parameter $\hat{\xi}\theta$ nicht trivial. Zur Vereinfachung betrachtet man deshalb nun die ersten zwei Glieder der Taylorentwicklung (Entwicklung zur 1. Ordnung) zum Entwicklungspunkt $\theta = 0$ und erhält den Term:

$$P_{3D}^{(p')}\left((\mathbf{I} + \hat{\xi}\theta)\bar{p}\right) = 0 \tag{4.64}$$

An dieser Stelle werden die Vorteile der Beschreibung von Bewegungstransformationen im Raum $se(3)$ deutlich. Da die Beschreibung im Exponenten bereits infinitesimal erfolgt (s. auch Abschnitt 2.4), beschränkt sich die Approximation zur 1. Ordnung auf das einfache hinschreiben der ersten zwei Glieder der Exponentialreihe:

$$\bar{p'} = e^{\hat{\xi}\theta}\bar{p} = \left(\sum_{k=0}^{\infty}\frac{(\hat{\xi}\theta)^k}{k!}\right)\bar{p} \approx (\mathbf{I} + \hat{\xi}\theta)\bar{p} \tag{4.65}$$

Geometrisch betrachtet beschreibt dieser Term eine Gerade durch den Punkt p, deren Richtung und Lage im Raum durch die Twistparameter ξ festgelegt werden. Das wird besonders deutlich, wenn man die Gleichung 4.64 in die Punkt-Richtungs-Form einer Geraden umformt:

$$\bar{p'} = (\mathbf{I} + \hat{\xi}\theta)\bar{p} = \bar{p} + \theta(\hat{\xi}\bar{p}) \tag{4.66}$$

Zu bestimmen ist in diesem Fall aber nicht nur der Parameter θ, sondern auch der Twist ξ, der Position und Lage der Geraden festlegt. Die gestellte Aufgabe lautet jetzt:

Suche den Tangentialvektor zur vorgegebenen starren Bewegung.

Die Gleichung 4.64 entspricht der Matrixschreibweise:

$$\begin{bmatrix} 1 & 0 & 0 & -p'_x \\ 0 & 1 & 0 & -p'_y \\ 0 & 0 & 1 & -p'_z \end{bmatrix} \begin{bmatrix} 1 & -\theta\omega_z & \theta\omega_y & \theta v_x \\ \theta\omega_z & 1 & -\theta\omega_x & \theta v_y \\ -\theta\omega_y & \theta\omega_x & 1 & \theta v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = 0. \tag{4.67}$$

Bemerkung 4.1 *Durch die Taylor-Approximation und die Abhängigkeit der Punktkoordinaten p' von verrauschten Sensordaten ist nicht zu erwarten, dass die Gleichung wirklich erfüllt wird. Im Abschnitt 3.3.1 wurde jedoch schon gezeigt, dass der Betrag des Terms auf der linken Seite einen Wert liefert, der als Fehlermaß genutzt werden kann. Die Aufgabe lautet somit:*

$$min_{\hat{\xi}\theta \in se(3)} \mid P_{3D}^{(p')}(\hat{\xi}\theta)\bar{p} \mid . \tag{4.68}$$

Um die Minimierung durchzuführen, wird das Gleichungssystem nach den Unbekannten umgestellt. Die Gleichung hat dann die folgende Form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \begin{bmatrix} \theta v_x \\ \theta v_y \\ \theta v_z \\ \theta\omega_x \\ \theta\omega_y \\ \theta\omega_z \end{bmatrix} = \begin{bmatrix} p'_x - p_x \\ p'_y - p_y \\ p'_z - p_z \end{bmatrix}. \tag{4.69}$$

Durch die Approximation und das entsprechende Umstellen der Gleichung wird die nichtlineare Zwangsbedingung 4.62 in ein unterbestimmtes lineares Gleichungssystem der Form $Ax = b$ gebracht. Um eine eindeutige Lösung zu erhalten, muss das Gleichungssystem ergänzt werden. Zur Lösung sind in diesem Fall mindestens drei Punktkorrespondenzen nötig (siehe hierzu auch Abschnitt 5). Die aus den einzelnen Korrespondenzen generierten Matrizen können dann zu einem Gleichungssystem zusammengesetzt werden. Die Aufgabe ist damit auf das Lösen eines linearen Ausgleichsproblems zurückgeführt worden. Der genaue Vorgang wird im Laufe des Abschnittes noch deutlicher werden.

Zur Vereinfachung der Schreibweise wird für die Systemmatrix der Punkte 0. Grades die Bezeichnung

$$\mathbf{PP_0} := \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \tag{4.70}$$

und für den Systemvektor die Bezeichnung

$$PPb := [p'_x - p_x \; p'_y - p_y \; p'_z - p_z]^T \tag{4.71}$$

eingeführt.

4.1.2. Punkte 1-ten Grades

Bisher beschränkte sich die Modellierung auf die Bewegungsschätzung eines starren Körpers. Nun wird das Modell auf Objekte mit einem internen Gelenk erweitert. Für Punkte 1. Gerades, d.h. Modellpunkte die durch ein Gelenk mit dem Basiskoordinatensystem des Grundkörpers in Verbindung stehen, setzt sich die Bewegungstransformation aus der bisher betrachteten Basistransformation und der Gelenktransformation zusammen. Die resultierende Zwangsbedingung lautet:

$$P_{3D}^{(p')}\left(e^{\hat{\xi}\theta}e^{\hat{\xi}_1\theta_1}\bar{p}\right) = 0 \tag{4.72}$$

Betrachtet man wieder die Glieder der Taylorreihenentwicklung zum Entwicklungspunkt $\theta = \theta_1 = 0$ bis zur ersten Ordnung, so erhält man den folgenden Ausdruck:

$$P_{3D}^{(p')}\left((\mathbf{I} + \hat{\xi}\theta + \hat{\xi}_1\theta_1)\bar{p}\right) = 0 \tag{4.73}$$

In diesem Schritt wird der Vorteil der gewählten Parametrisierung wieder besonders deutlich. Die nichtkommutative Hintereinanderausführung der Transformationen in Gleichung 4.72 wird durch die lineare Approximation in eine kommutative Linearkombination von Elementen aus $se(3)$ überführt. Da in diesem Fall die Lage der Rotationsachse des 1. Gelenks bekannt ist, liegen die Einträge der Matrix ξ_1 bereits fest. Die freien Parameter sind also $\xi = (v_{0,x}, v_{0,y}, v_{0,z}, \omega_{0,x}, \omega_{0,y}, \omega_{0,z})^T$, θ und θ_1. Stellt man die Formel 4.73 analog zu Abschnitt 4.1.1 nach den Unbekannten um, so ergibt sich die folgende Systemmatrix:

$$\mathbf{PP_1} := \begin{bmatrix} & -p_y\omega_{1,z} + p_z\omega_{1,y} + v_{1,x} \\ \mathbf{PP_0} & -p_z\omega_{1,x} + p_x\omega_{1,z} + v_{1,y} \\ & -p_x\omega_{1,y} + p_y\omega_{1,x} + v_{1,z} \end{bmatrix} \tag{4.74}$$

Der Systemvektor auf der rechten Seite des Gleichungssystems entspricht wieder dem Vektor PPb, also der Differenz von Referenz- und Modellpunkt. Die Parameter des Lösungsvektors stimmen mit denen des ersten Falls überein, er enthält aber die zusätzliche Unbekannte θ_1:

$$\mathbf{PP_1}\begin{bmatrix} \theta v_x \\ \theta v_y \\ \theta v_z \\ \theta\omega_x \\ \theta\omega_y \\ \theta\omega_z \\ \theta_1 \end{bmatrix} = PPb \tag{4.75}$$

4.1.3. Punkte g-ten Grades

Verallgemeinert man die Transformationsgleichung aus dem letzten Abschnitt auf Punkte g-ten Grades, so kann man aus der Gleichung

$$P_{3D}^{(p')}\left(e^{\hat{\xi}\theta}e^{\hat{\xi}_1\theta_1}\cdots e^{\hat{\xi}_g\theta_g}\bar{p}\right) = 0 \tag{4.76}$$

nach der Taylorreihenapproximation die Systemmatrix für Punkte g-ten Grades aufstellen.

$$\mathbf{PP_g} := \begin{bmatrix} \mathbf{PP_{(g-1)}} & \begin{matrix} -p_y\omega_{g,z} + p_z\omega_{g,y} + v_{g,x} \\ -p_z\omega_{g,x} + p_x\omega_{g,z} + v_{g,y} \\ -p_x\omega_{g,y} + p_y\omega_{g,x} + v_{g,z} \end{matrix} \end{bmatrix} \tag{4.77}$$

Die rechte Seite des Gleichungssystems berechnet, unabhängig vom Grad des Punktes, den Differenzvektor zwischen Modellpunkt p und dem zu verfolgenden Punkt p'.

Jedes Gelenk erweitert also das Grundgleichungssystem um einen weiteren freien Parameter. Dabei geht die Abhängigkeit der Punkte höheren Grades von anderen Gelenken nicht verloren. Durch das Lösen des Gleichungssystems mit Hilfe eines linearen Ausgleichsverfahrens[11] haben die Punkte höheren Grades auch Einfluß auf die Parameter der Gelenke kleineren Grades. Die Abhängigkeiten sind somit direkt im Gleichungssystem enthalten. Auf eine explizite Formulierung der kinematischen Beziehungen, wie in dem hierarchischen Ansatz von Weik und Liedtke [40], kann in diesem Fall verzichtet werden.

4.2. Die Punkt zu Sehgerade Bedingung

Die Punkt zu Sehgerade Bedingung beschreibt den Zusammenhang zwischen einem Modellpunkt und der aus einem Bildpunkt resultierenden Sehgeraden. Dabei wird die Sehgerade, die durch Moment $m = [m_x\ m_y\ m_z]^T$ und Richtung $r = [r_x\ r_y\ r_z]^T$ festgelegt ist und der zugehörige Modellpunkt $p = [p_x\ p_y\ p_z]^T$ als bekannt vorausgesetzt. Gesucht ist nun eine affine Transformation, die den Modellpunkt auf die Projektionsgerade abbildet (s. Abb. 17).

4.2.1. Punkte 0-ten Grades

Modellpunkte 0-ten Grades unterliegen jetzt der Zwangsbedingung:

$$G_{3D}^{(r,m)}\left(e^{\hat{\xi}\theta}\bar{p}\right) = 0 \tag{4.78}$$

Wie bei der Punkt zu Punkt Bedingung kann auch hier wieder eine Taylorapproximation der Exponentialfunktion bis zur 1. Ordnung durchgeführt werden. Dies führt wieder zu einer linearen Approximation der Gleichung:

$$G_{3D}^{(r,m)}\left((\mathbf{I} + \hat{\xi}\theta)\bar{p}\right) = 0. \tag{4.79}$$

In Matrixform entspricht das der Gleichung:

$$\begin{bmatrix} 0 & r_z & -r_y & -m_x \\ -r_z & 0 & r_x & -m_y \\ r_y & -r_x & 0 & -m_z \end{bmatrix} \begin{bmatrix} 1 & -\theta\omega_z & \theta\omega_y & \theta v_x \\ \theta\omega_z & 1 & -\theta\omega_x & \theta v_y \\ -\theta\omega_y & \theta\omega_x & 1 & \theta v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = 0. \tag{4.80}$$

[11] Um eine numerisch stabile Lösung zu erreichen, wird in der Implementation die Householder-Reduktion [32] verwendet.

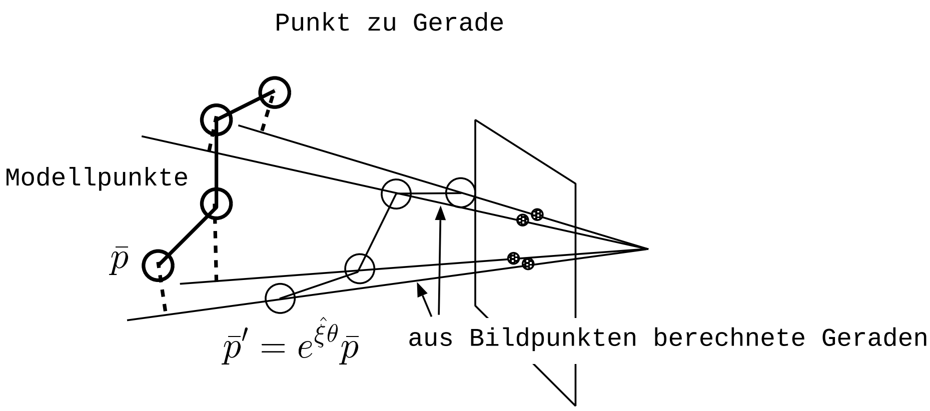


Abbildung 17: Die Punkt zu Sehgerade Zwangsbedingung: 3D Punkt aus dem Referenzmodell wird mit dem Sehstrahl (Gerade) des korrespondierenden Bildpunktes in Beziehung gesetzt.

Auch hier kann die Beschreibung der Rotationen mit Elementen aus $se(3)$ ausgenutzt und die Taylorapproximation auf das Hinschreiben der ersten zwei Glieder der Exponentialreihe reduziert werden. Stellt man nach den Unbekannten um, so erhält man das folgende Gleichungssystem:

$$\mathbf{PG_0}\,\xi\theta = PGb, \tag{4.81}$$

wobei

$$\mathbf{PG_0} := \left[\begin{array}{cccc} & -p_zr_z - p_yr_y & p_xr_y & p_xr_z \\ -\hat{\mathbf{r}} & p_yr_x & -p_xr_x - p_zr_z & p_yr_z \\ & p_zr_x & p_zr_y & -p_yr_y - p_xr_x \end{array}\right], \tag{4.82}$$

mit

$$\hat{\mathbf{r}} := \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{4.83}$$

und

$$PGb := \begin{bmatrix} p_zr_y - p_yr_z + m_x \\ p_xr_z - p_zr_x + m_y \\ p_yr_x - p_xr_y + m_z \end{bmatrix}. \tag{4.84}$$

4.2.2. Punkte 1-ten Grades

Wie im Fall der Punkt zu Punkt Korrespondenz lässt sich das Prinzip auf Punkte erweitern, die von einem Gelenk abhängig sind. Für Punkte 1-ten Grades lautet die Zwangsbedingung:

$$G_{3D}^{(r,m)}\left(e^{\hat{\xi}\theta}e^{\hat{\xi}_1\theta_1}\bar{p}\right) = 0 \tag{4.85}$$

Auch in diesem Fall zeigt sich nach der Taylorapproximation, dass sich die nicht-kommutative Multiplikation von Transformationsmatrizen in eine kommutative Addition der Elemente aus $se(3)$ verwandelt. Die hieraus resultierende Systemmatrix setzt sich, wie im Fall der Punkt zu Punkt Bedingung, aus der Systemmatrix für Punkte 0-ten Grades und einer zusätzlichen Spalte zusammen. Um die Schreibweise zu vereinfachen werden die Hilfsterme H1 bis H3 eingeführt:

$$\begin{aligned} H1 &:= p_z\omega_{1,y} - p_y\omega_{1,z} + v_{1,x}, \\ H2 &:= p_x\omega_{1,z} - p_z\omega_{1,x} + v_{1,y}, \\ H3 &:= p_y\omega_{1,x} - p_x\omega_{1,y} + v_{1,z}. \end{aligned} \tag{4.86}$$

Die zusammengesetzte Systemmatrix sieht dann wie folgt aus:

$$\mathbf{PG_1} := \begin{bmatrix} & H2\ r_z - H3\ r_y \\ \mathbf{PG_0} & H3\ r_x - H1\ r_z \\ & H1\ r_y - H2\ r_x \end{bmatrix}. \tag{4.87}$$

4.2.3. Punkte g-ten Grades

Für die Softwareimplementation ist es wichtig den allgemeinen Fall zu betrachten, um die Systematik in den zu generierenden Gleichungen herauszuarbeiten. Im Folgenden wird deshalb wieder ein Modellpunkt betrachtet, der über $g \in \mathbb{N}$ Gelenke mit dem Grundkörper verbunden ist. Da die Herleitung analog zu den bisher betrachteten Fällen erfolgt, ist hier nur die Systemmatrix angegeben:

$$\begin{aligned} H1_g &:= p_z\omega_{g,y} - p_y\omega_{g,z} + v_{g,x}, \\ H2_g &:= p_x\omega_{g,z} - p_z\omega_{g,x} + v_{g,y}, \\ H3_g &:= p_y\omega_{g,x} - p_x\omega_{g,y} + v_{g,z}, \end{aligned} \tag{4.88}$$

$$\mathbf{PG_g} := \begin{bmatrix} & H2_g\ r_z - H3_g\ r_y \\ \mathbf{PG_{(g-1)}} & H3_g\ r_x - H1_g\ r_z \\ & H1_g\ r_y - H2_g\ r_x \end{bmatrix}. \tag{4.89}$$

Das Gleichungssystem für eine Korrespondenz bei der Punkt zu Sehgerade Bedingung hat also die Form:

$$\mathbf{PG_g}\left[\theta v_x\ \theta v_y\ \theta v_z\ \theta\omega_x\ \theta\omega_y\ \theta\omega_z\ \theta_1\ \ldots\ \theta_g\right]^T = \begin{bmatrix} p_z r_y - p_y r_z + m_x \\ p_x r_z - p_z r_x + m_y \\ p_y r_x - p_x r_y + m_z \end{bmatrix} \tag{4.90}$$

Wie bei der Punkt zu Punkt Bedingung sind zur Bestimmung der Transformationsparameter mehrere Korrespondenzen erforderlich (s. auch Abschnitt 5). Zur Lösung müssen wieder mehrere Gleichungssysteme zu einem überbestimmten Gleichungssystem zusammengesetzt werden.

4.3. Die Punkt zu Sehebene Bedingung

Die bisher betrachteten Zwangsbedingungen nutzen Punktinfomationen aus (auch die Sehgeraden im vorherigen Abschnitt werden aus Punktinformationen im Bild gewonnen). Die Extraktion von Kanten und Geraden aus Bildern lässt sich jedoch häufig exakter und sicherer durchführen, da zur Bestimmung der Lage und Position dieser Entitäten ein ganzer Bildbereich genutzt wird. Die Verfahren sind dadurch meist unempfindlicher gegenüber verrauschten Daten. Um die bei der Merkmalsextraktion gewonnenen Informationen ausnutzen zu können, müssen die zweidimensionalen Geradeninformationen mit dem Modell in Verbindung gebracht werden.

Das Bild einer Geraden im Kamerabild entspricht einer Ebene im dreidimensionalen Raum. Wird diese wie in Abschnitt 2.6.5 beschrieben, durch ihren Normalenvektor $n = [n_x \; n_y \; n_z]^T$ und die Hessedistanz $\delta_h \in \mathbb{R}$ parametrisiert, so erhält man die folgende Zwangsbedingung für die Punkt zu Sehebene Beziehung:

$$E_{3D}^{(n,\delta_h)}\left(e^{\hat{\xi}\theta} \dots e^{\hat{\xi}_g\theta_g}\bar{p}\right) = 0 \tag{4.91}$$

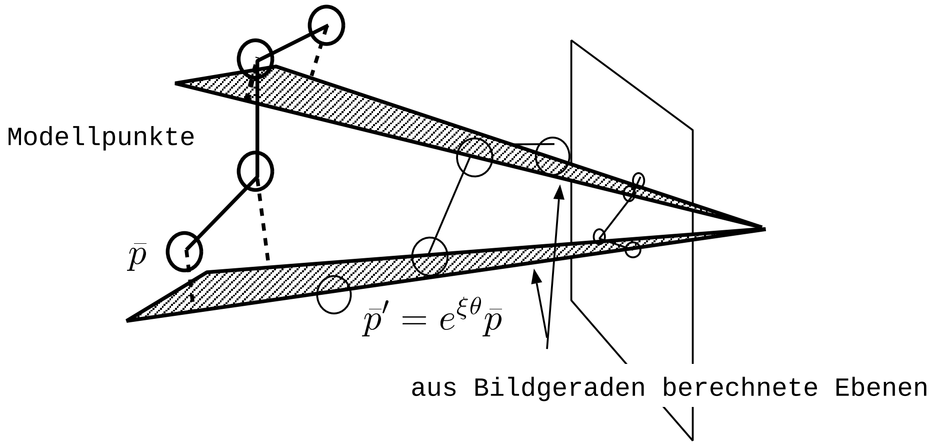


Abbildung 18: Die Punkt zu Sehebene Zwangsbedingung: 3D Punkt aus dem Referenzmodell wird mit der Sehebene des korrespondierenden Bildpunktes in Beziehung gesetzt.

Wieder kann durch Taylorapproximation und Umstellen der Gleichung nach den Unbekannten eine Liniarisierung des Problems erreicht werden. Da die Umformungen analog zu den beiden letzten Abschnitten erfolgen, werden hier nur noch die zur Lösung benötigten Matrizen notiert. Der Systemvektor der Punkt zu Sehebene Bedingung hat auf der rechten Seite die Form

$$PEb := \delta_h - n_x p_x - n_y p_y - n_z p_z. \tag{4.92}$$

Die Systemmatrizen für die verschiedenen Abhängigkeiten von den Gelenken sind in den folgenden Abschnitten 4.3.1, 4.3.2 und 4.3.3 angegeben.

4.3.1. Punkte 0-ten Grades

Zunächst wieder die zur Basistransformation gehörige Systemmatrix:

$$\mathbf{PE_0} := \begin{bmatrix} n_x & n_y & n_z & n_z p_y - n_y p_z & n_x p_z - n_z p_x & n_y p_x - n_x p_y \end{bmatrix}. \tag{4.93}$$

4.3.2. Punkte 1-ten Grades

Die Erweiterung des System für Objekte mit einem internen Gelenk hat die Form:

$$\begin{aligned} H1 &:= \omega_{1,x}(n_z p_y - n_y p_z), \\ H2 &:= \omega_{1,y}(n_x p_z - n_z p_x), \\ H3 &:= \omega_{1,z}(n_y p_x - n_x p_y), \\ NV_1 &:= n_x v_{1,x} + n_y v_{1,y} + n_z v_{1,z}, \end{aligned} \tag{4.94}$$

$$\mathbf{PE_1} := \begin{bmatrix} \mathbf{PE_0} & NV_1 + H1 + H2 + H3 \end{bmatrix}. \tag{4.95}$$

4.3.3. Punkte g-ten Grades

Die zur Implementation wichtige Verallgemeinerung führt dann zur allgemeinen Systemmatrix der Punkt zu Sehebenen Bedingung:

$$\begin{aligned} H1_g &:= \omega_{g,x}(n_z p_y - n_y p_z), \\ H2_g &:= \omega_{g,y}(n_x p_z - n_z p_x), \\ H3_g &:= \omega_{g,z}(n_y p_x - n_x p_y), \\ NV_g &:= n_x v_{g,x} + n_y v_{g,y} + n_z v_{g,z}, \end{aligned} \tag{4.96}$$

$$\mathbf{PE_g} := \begin{bmatrix} \mathbf{PE_{(g-1)}} & NV_g + H1_g + H2_g + H3_g \end{bmatrix}. \tag{4.97}$$

Bemerkenswert ist, dass die Korrespondenzen von Punkten, Geraden und Ebenen durch die im Abschnitt 2.6 eingeführte Beschreibung einheitlich behandelt werden können. Die resultierenden Gleichungssysteme können sogar, wie in Abb. 19 angedeutet, kombiniert werden. Das Verfahren ist dadurch sehr flexibel und kann in verschiedenen Bereichen eingesetzt werden. In Situationen, bei denen verschiedene Informationen über eine Szene zur Verfügung stehen und diese miteinander kombiniert werden sollen, kann dies durch einfaches Aneinanderreihen der beschriebenen Matrizen erreicht werden.

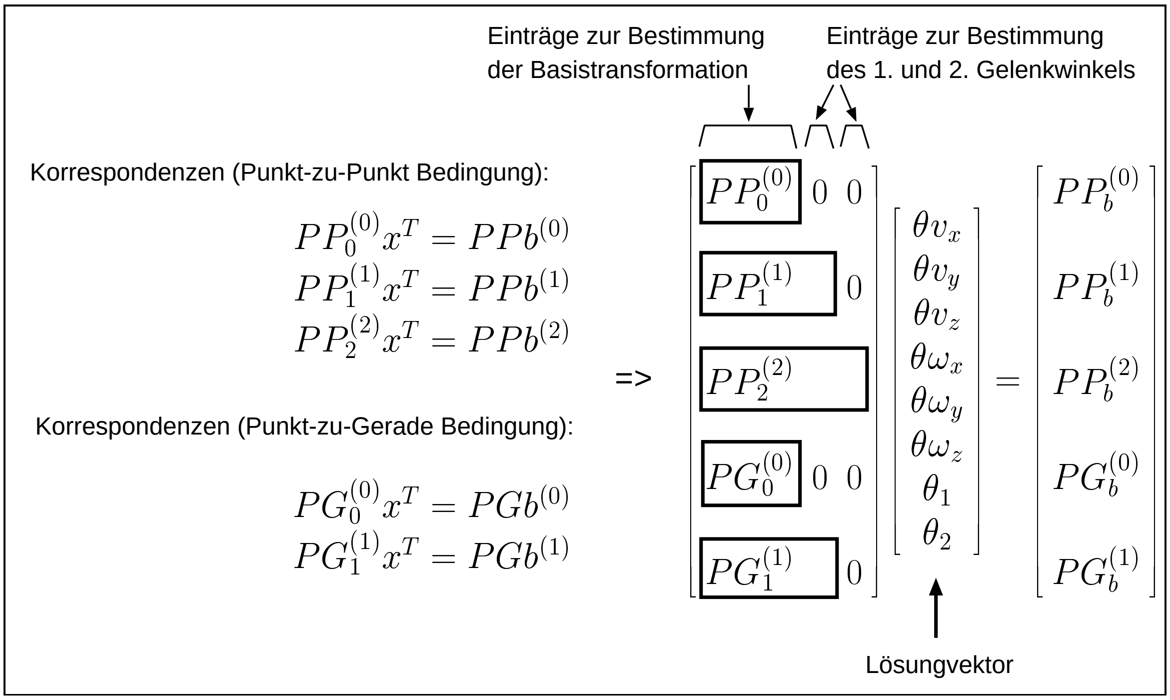


Abbildung 19: Zusammensetzen des Gleichungssystems: Einträge in den Spalten der Systemmatrix, bei denen keine Abhängigkeiten vorliegen, werden mit **0** besetzt. Der in runden Klammern angegebene Index entspricht dem Grad des zugehörigen Modellpunktes.

4.4. Berechnung der Basistransformation

In den Gleichungssystemen werden sechs Parameter ($\theta v_x, \theta v_y, \theta v_z, \theta \omega_x, \theta \omega_y, \theta \omega_z$) zur Beschreibung der Basistransformation ermittelt. Zur Berechnung der neuen Konfiguration wird der Winkel aus dem Lösungsvektor bestimmt und die Twistparameter normiert:

$$\theta = ||\omega||, \hat{\xi} = \frac{1}{\theta}(\theta \hat{\xi}). \tag{4.98}$$

Die entsprechende Transformationsmatrix lässt sich jetzt mit dem im Anhang C beschriebenen Verfahren bestimmen.

4.5. Einschränkung auf planare Bewegungen

Die in der Arbeit vorgestellten Gleichungssysteme gehen von einem Objekt aus, das sich im Raum frei bewegen kann. Möchte man das System bei der Navigation eines Roboters benutzen, um die Position des Roboters in Bezug zu einer Landmarke zu schätzen, so kann man die möglichen Freiheitsgerade weiter einschränken. Bei der

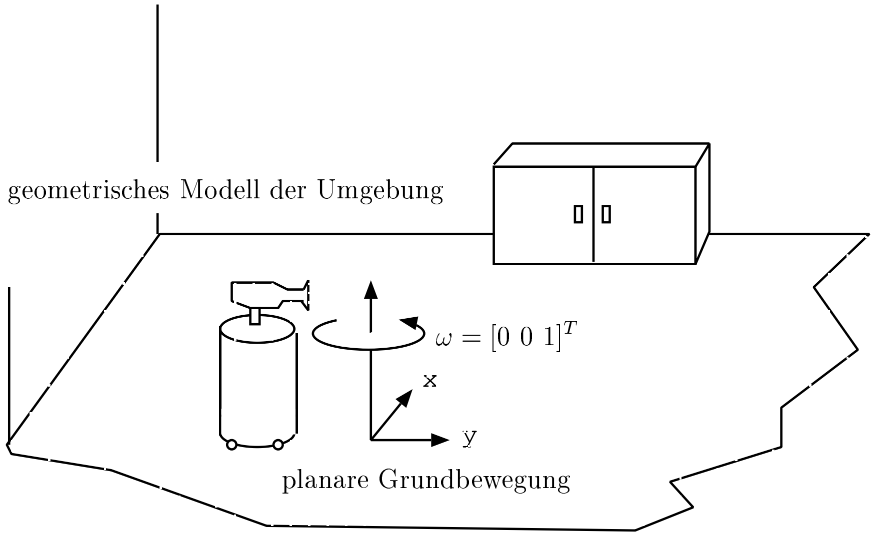


Abbildung 20: Vereinfachung der Bewegungsschätzung durch Einschränkung der Grundtransformation auf die x-y-Ebene.

Roboternavigation kann die mögliche Bewegung auf die x-y-Ebene eingeschränkt werden. Durch Festlegen des Richtungsvektors $\omega = [0\ 0\ 1]^T$ und die Einschränkung auf Translationen in der Ebene $v = [v_x\ v_y\ 0]^T$ erhält man für die Basistransformation die folgende Twistbeschreibung:

$$\hat{\xi}\theta = \theta \begin{bmatrix} 0 & -1 & 0 & v_x \\ 1 & 0 & 0 & v_y \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.99}$$

Die zur Bestimmung der Basistransformation benötigten Parameter haben sich durch die Einschränkung von sechs auf drei reduziert. Bei der hier vorgestellten Roboternavigation (s. Experiment im Abschnitt 7.5 und [9]) , die die Punkt zu Sehebenen Beziehung benutzt, kommt die Poseschätzung dadurch mit bis zu drei Korrespondenzen weniger aus. Das Umstellen der Gleichungen und die Approximation erfolgen analog zum allgemeinen Fall und sollen hier nicht wiederholt werden.

In der Tabelle 2 sind die Systemmatrizen für den Fall einer planaren Bewegung aufgeführt. Die Bezeichnungen sind wie im allgemeinen Fall gewählt. Der Lösungsvektor hat die Form $[v_x \quad v_y \quad \theta \quad \theta_1 \quad \theta_2 \quad \ldots]^T$.

Punkt zu Punkt
$\mathbf{PP_0^{planar}} := \begin{bmatrix} 1 & 0 & p_y \\ 0 & 1 & p_x \\ 0 & 0 & 0 \end{bmatrix}$ $\mathbf{PP_g^{planar}} := \begin{bmatrix} \mathbf{PP_{g-1}^{planar}} & \begin{matrix} \omega_{g,y}p_z - p_y\omega_{g,z} + v_{g,x} \\ \omega_{g,z}p_x - \omega_{g,x}p_z + v_{g,y} \\ \omega_{g,x}p_y - \omega_{g,y}p_x + v_{g,z} \end{matrix} \end{bmatrix}$ $PPb^{planar} := \begin{bmatrix} p'_x - p_x & p'_y - p_y & p'_z - p_z \end{bmatrix}^T$
Punkt zu Sehgerade
$\mathbf{PL_0^{planar}} := \begin{bmatrix} 0 & d_z & d_z p_x \\ -d_z & 0 & d_z p_y \end{bmatrix}$ $\mathbf{PG_g^{planar}} := \begin{bmatrix} \mathbf{PG_{g-1}^{planar}} & \begin{matrix} d_z p_x \omega_{g,z} + d_y \omega_{g,y} p_x - d_z \omega_{g,x} p_z + d_z v_{g,y} - d_y v_{g,z} - d_y \omega_{g,x} p_y \\ d_z p_y \omega_{g,z} - d_x \omega_{g,y} p_x - d_z \omega_{g,y} p_z - d_z v_{g,x} + d_x v_{g,z} + d_x \omega_{g,x} p_y \end{matrix} \end{bmatrix}$ $PGb^{planar} := [d_y p_z + d_y v_{1,z} - d_z p_y + m_x \qquad d_z p_x - d_x v_{1,z} + m_y - d_x p_z]$
Punkt zu Sehebene
$\mathbf{PE_0^{planar}} := [n_x \qquad n_y \qquad -n_x p_y + n_y p_x \qquad]$ $\mathbf{PE_g^{planar}} := \begin{bmatrix} \mathbf{PE_{g-1}^{planar}} & n_x O1 + n_y O2 + n_z O3 \end{bmatrix}$ $O1 := \omega_{g,y}p_z - \omega_{g,z}p_y + v_{g,x}$ $O2 := \omega_{g,z}p_x - \omega_{g,x}p_z + v_{g,y}$ $O3 := \omega_{g,x}p_y - \omega_{g,y}p_x + v_{g,z}$ $PEb^{planar} := [\delta_h - n_x p_x - n_y p_y - n_z p_z]$

Tabelle 2: Zusammenfassung der Systemmatrizen für den planaren Fall.

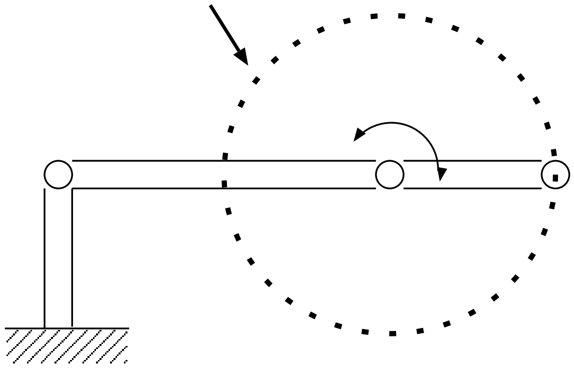


Abbildung 21: Undefinierte Pose bei fehlender Korrespondenz für das Endglied einer kinematischen Kette

5. Minimalvoraussetzungen

Bei der Anwendung des Verfahrens ist es wichtig, genügend Informationen zur Lösung des Gleichungssystems zur Verfügung zu haben. In diesem Abschnitt sollen deshalb die wesentlichen Voraussetzungen, die zur Bestimmung der Posenparamerter nötig sind, zusammengestellt werden. Die hier aufgeführten Voraussetzungen beschränken sich im allgemeinen Fall auf notwendige Bedingungen, da zur Untersuchung der hinreichenden Bedingungen viele Fallunterscheidungen nötig sind.

Für eine spezielle Konfiguration genügt es den Rang der Systemmatrix zu berechenen. Lösbar ist das Gleichungssystem, wenn der Rang mit der Anzahl der Spalten (entspricht der Anzahl der Unbekannten) übereinstimmt. Die allgemeinen Vorraussetzungen zur Lösbarkeit eines linearen Ausgleichsproblems werden in verschiedenen Büchern der Numerik (wie z.B. [32]) behandelt.

5.1. Grundvoraussetzungen

Bei der Bestimmung der Pose von Objekten mit kinematischen Ketten ist es notwendig, dass Informationen über das Anfang- und Endglied der Kette vorhanden sind. D.h. zur Schätzung muß sowohl für das Basisglied wie auch für die Endglieder einer Kette jeweils mindestens eine Korrespondenz angegeben werden. Die Abbildung 21 veranschaulicht die Freiheitsgrade bei fehlender Endkorrespondenz.

5.2. Schätzung der Basiskonfiguration

Zur Bestimmung der Basiskonfiguration ist die Lage des Objektkoordinatensystems zum Weltkoordinatensystem zu bestimmen. Diese wird durch sechs Parameter beschrieben. Im Fall der Punkt zu Punkt Bedingung sind also mindestens drei Punktkorrespondenzen notwendig um eine Schätzung durchführen zu können (jeder Punkt

$$\mathbf{PP_0} := \left[\boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}} \boxed{\begin{matrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{matrix}} \right]$$

Einheitsmatrix => Rang 3

Matrizen für das Kreuzprodukt => max. Rang 2

$$\mathbf{PG_0} := \left[\boxed{-\hat{\mathbf{r}}} \begin{matrix} -p_z r_z - p_y r_y & p_x r_y & p_x r_z \\ p_y r_x & -p_x r_x - p_z r_z & p_y r_z \\ p_z r_x & p_z r_y & -p_y r_y - p_x r_x \end{matrix} \right]$$

mit

$$\hat{\mathbf{r}} := \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

Matrix mit einer Zeile => max. Rang 1

$$\mathbf{PE_0} := \begin{bmatrix} n_x & n_y & n_z & n_z p_y - n_y p_z & n_x p_z - n_z p_x & n_y p_x - n_x p_y \end{bmatrix}.$$

Abbildung 22: Matrizen zur Bestimmung der Basistransformation

Punkt zu Punkt	Punkt zu Sehgerade	Punkt zu Sehebene
min. 3	min. 3	min. 6

Tabelle 3: Notwendige Anzahl von Korrespondenzen zur Ermittlung einer Basistransformation

legt die Lage einer Koordinatenachse fest).

Betrachtet man die entsprechende Systemmatrix $\mathbf{PP_0}$ (Abb. 22), so findet man auf den ersten Spalten eine 3×3 Einheitsmatrix. Die erste Punktinformation liefert damit bereits ein Gleichungssystem mit dem Rang 3. Eine weitere Punktkorrespondenz liefert in den ersten drei Spalten keine neuen linear unabhängigen Zeilenvektoren. Der nun entscheidene zweite Block der Matrix $\mathbf{PP_0}$ entspricht dem Aufbau einer Matrix für das Kreuzprodukt. Sie hat einen maximalen Rang von 2. Der durch zwei Punktkorrespondenzen erreichbare Rang ist somit 5. Zur Lösung des linearen Ausgleichsproblem ist deshalb mindestens eine weitere Korrespondenz erforderlich um den Rang der Matrix auf 6 zu erhöhen.

Auch bei der Punkt zu Sehgerade Bedingung kann eine Poseschätzung mit drei Korrespondenzen durchgeführt werden. Jede Korrespondenz kann den Rang der Systemmatrix um maximal 2 erhöhen, da der erste dreier Block einer Kreuzproduktmatrix entspricht.

Bei der Punkt zu Sehebene Bedingung werden hingegen wenigstens sechs Vorgaben benötigt. Die Bedingung liefert für jede Korrespondenz nur eine neue Zeile (s. $\mathbf{PE_0}$ in Abb. 22). Der Rang der Systemmatrix erhöht sich dadurch maximal um den Rang 1.

Punkt zu Punkt	Punkt zu Sehgerade	Punkt zu Sehebene
min. < 1	min. < 1	min. 1

Tabelle 4: Notwendige Anzahl von Korrespondenzen zur Schätzung einer Gelenkstellung

5.3. Schätzung der Gelenkstellungen

Bei der Schätzung von Objekten mit internen Gelenken wird das Gleichungssystem für jedes Gelenk um eine Unbekannte erweitert, d.h. jedes Gelenk erweitert die Systemmatrix um eine Spalte. Bei der Punkt zu Punkt und der Punkt zu Sehgerade Bedingung kann man somit im günstigsten Fall durch Hinzufügen nur einer weiteren Korrespondenz, die Stellungen von zwei zusätzlichen Gelenken schätzen. Bei Benutzung der Punkt zu Sehebene Bedingung ist jedoch für jedes Gelenk mindestens eine Korrespondenz nötig, da hier der Rang der Systemmatrix um maximal 1 erhöht wird.

6. Implementation der Poseschätzung

Zu den Aufgaben dieser Arbeit gehört die Implementation und Integration der entwickelten Verfahren zur Poseschätzung in die KiVi-Grap Umgebung [17]. Da die Poseschätzung auf einem Modell des zu verfolgenden Objektes beruht, stellt sich die Frage nach einer geeigneten Datenstruktur zur Verwaltung der Objektgeometrie. Bei der Wahl der Datenstruktur sollte dabei auf Erweiterbarkeit, Kompatibilität und einfache Anwendbarkeit geachtet werden, damit nachfolgende Arbeiten auf den Strukturen aufbauen können.

Ein weiterer Punkt, der bei der Implementation beachtet werden muss, ist die Effizienz der Datenstruktur in Bezug auf die Zugriffsgeschwindigkeit. Dies spielt besonders dann eine Rolle, wenn die Verfahren im Online-Betrieb, wie zum Beispiel bei der Roboternavigation, eingesetzt werden sollen. Alle Anforderungen gleichzeitig zu erfüllen ist häufig nicht möglich, bei der Implementation musste deshalb oft ein Kompromiss zwischen Geschwindigkeit, Speicherverbrauch und Übersichtlichkeit der Programmstruktur gefunden werden. In diesem Abschnitt sollen die wichtigsten Komponenten der Implementation vorgstellt und die Grundideen erläutert werden.

6.1. Interne Datenstruktur

Das bisher betrachtete Modell einer kinematischen Kette beschränkte sich auf die Untersuchung einer einzelnen Gelenkkette. Um das Modell eines Menschen oder eines Tieres darstellen zu können, ist es wichtig mehrere kinematische Ketten gleichzeitig behandeln zu können. Zur Verwaltung der Abhängigkeiten bietet sich in diesem Fall eine Baumstruktur an (s. Abb. 23). Sie wird durch eine verkette Liste implementiert, deren Elemente selbst wieder Listen enthalten können.

6.2. Externe Datenrepräsentation

Um die Anwendung flexibel zu gestalten und den Datenaustausch zu anderen Anwendungen möglich zu machen, müssen Funktionen zum Laden und Speichern der Objekte vorhanden sein. Um von Anfang an eine gute Grundlage für die weitere Nutzung der Arbeit zu legen, sollte das verwendete Dateiformat folgende Anforderungen erfüllen:

i. Beschreibung verschiedener Objektgeometrien (Punkte, Linien, Flächen, Körper)

ii. Beschreibung der Zusammenhänge zwischen den Objekten
(Hierarchien und Abhängigkeiten zwischen den Objekten müssen ausgedrückt werden können)

iii. Einfache Modellierung von neuen Objekten

iv. Erweiterbarkeit

v. Kompatiblität zu anderen Programmen

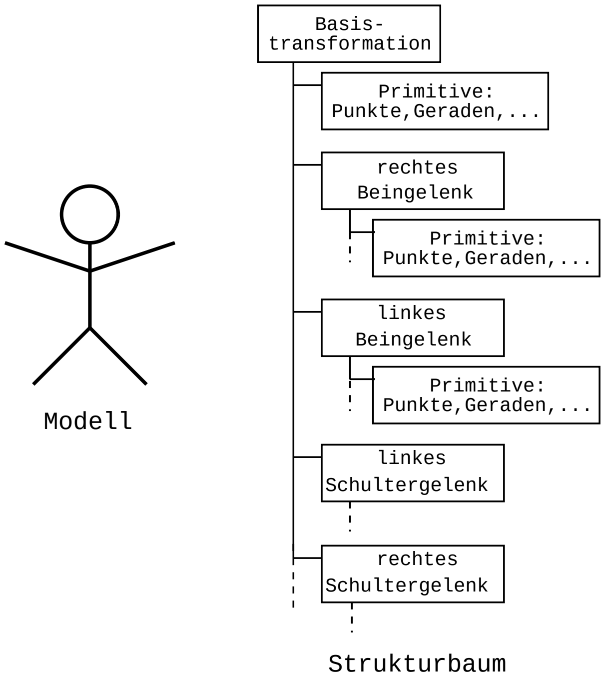


Abbildung 23: Beispiel einer Baumstruktur zur internen Verwaltung der Modelldaten.

vi. Möglichkeiten zur graphischen Visualisierung der Objekte

Bei Recherchen im Internet [28] und Büchern zur Computergrafik [19, 8], stellte sich das offene und standartisierte VRML-Format (Virtual Reality Modeling Language) [35] als die für diese Aufgabe am besten geeignete Beschreibungssprache heraus. Es bietet ein von Menschen leicht lesbares, textbasiertes Dateiformat, festgelegte Definition einiger Standartprimitiva, wie z.B. Punkt, Linie, Kugel und die Möglichkeit mit Hilfe von Prototypen eigene geometrische Objekte anzulegen. Ein weiterer Vorteil ist die Kompatiblität zu anderen Programmen und die für diese Anwendung besonders wichtige Möglichkeit der hierarchischen Strukturierung.

6.2.1. Die Beschreibungssprache VRML97

In diesem Abschnitt soll ein kurzer Überblick über die Beschreibungssprache VRML gegeben werden. Außerdem soll auf die Anpassungen und Einschränkungen bei der Implementation eingegangen werden, die in dieser Arbeit gemacht wurden. Eine nach dem ISO/IEC 14772 (VRML97) [35] Standard geschriebene VRML Datei besteht aus einer Reihe von Knoten(nodes). Jeder Knoten hat eine bestimmte Anzahl von Feldern, denen ein Name und ein Datentyp zugeordnet ist.

Beispiel 6.1
Eine Menge von dreidimensionalen Punkten wird durch den Knoten **Coordinate** *repräsentiert. Er ist wie folgt definiert:*

```
Coordinate
{
  exposedField MFVec3f point []
}
```

In diesem Fall hat der Knoten nur ein Datenfeld (`exposedField`) mit dem Namen point. Der Datentyp ist mit `MFVec3f` festgelegt und bedeutet in diesem Fall, dass es sich um eine Liste (`MF`=MultiField) von dreidimensionalen Vektoren (`Vec3`) handelt, deren Einträge als (ISO C) Gleitkommazahlen vorliegen (`f`=float). Die Definition von zwei Punkten erfolgt zum Beispiel durch:

```
...
Coordinate
{
  point [ 1.0 2.3 4.5,
          4.3 2.3 4.4 ]
}
...
```

Die folgende Liste zeigt die im Rahmen der Diplomarbeit benutzen Datentypen und Knoten.

Datentypen
`SFBool, SFInt32, MFInt32, SFFloat, MFFloat,`
`SFVec3f, MFVec3f, SFNode, MFNode, SFRotation`

Vordefinierte Knoten[12]
`Coordinate, Transform, Sphere, Cylinder,`
`Joint, RepresentationArtifical`

Jeder vordefinierte Knoten wird als eigene Klasse implementiert, die jeweils von einer Basisklasse für Knoten abgeleitet werden. Um die Daten aus der als Text vorliegenden VRML-Datei einzulesen und die entsprechend benötigten Instanzen der Klassen zu erzeugen, wird ein auf lex (Lexical Analyzer Generator) und yacc[13] [20] basierender Scanner und Parser implementiert, der die vom W3C veröffentlichte Grammatik zum VRML97-Standard [35] verwendet.

[12] Die beiden Knoten Joint und RepresentationArtificial sind nicht im VRML-Standard definiert und wurden den Anforderungen entsprechend eingeführt. Die verwendeten Namen sind an bereits vorhandene Klassen angelehnt.

[13] Die beiden Programme lex und yacc (bzw. deren GNU Varianten flex und bison) erzeugen ein C Programm anhand einer gegebenen Grammatikspezifikation. Dieses fasst syntaktisch zusammengehörende Elemente der vorgegebenen Spache zusammen und übergibt die Daten an benutzerdefinierte Funktionen.

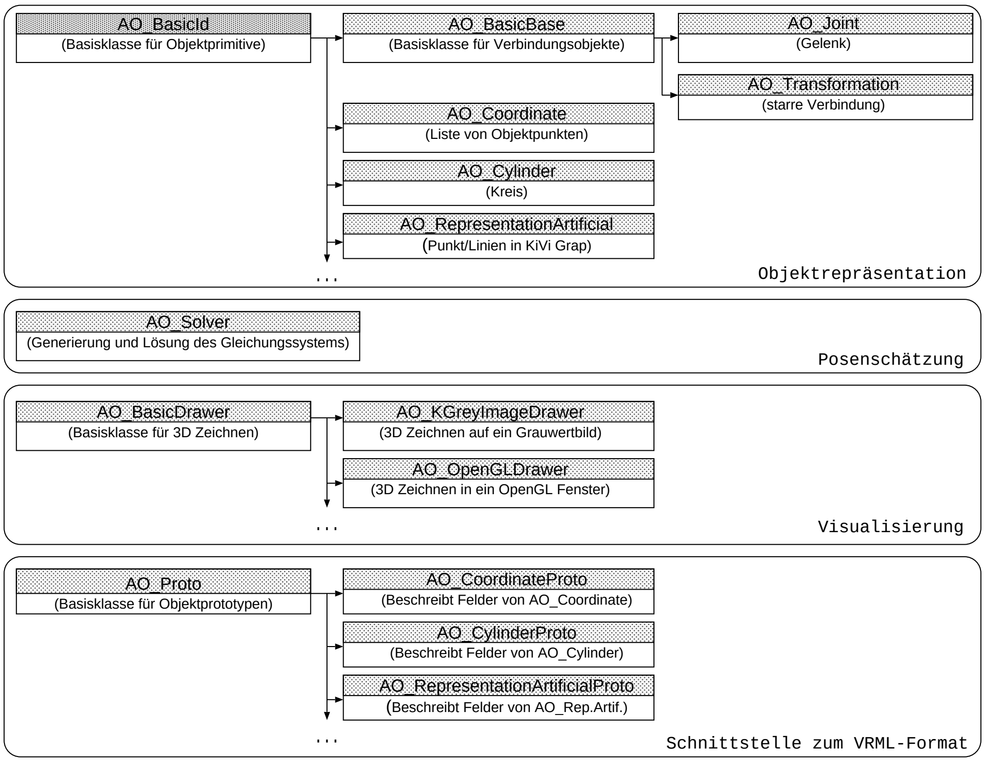


Abbildung 24: Das Diagramm zeigt den Strukturbaum der Klassenhierarchie

6.3. Die Klassenstruktur der Softwareimplementation

Die Implementation gliedert sich in vier Bereiche:

1) Klassen zur Objektrepräsentation

2) Klasse zur Berechnung der Pose

3) Klassen zur Visualisierung des Objektmodells

4) Funktionen zum Einlesen und Speichern der Modelldaten

Der Aufbau der Klassenhierarchie veranschaulicht die Abbildung 24. Die Basisklassen `AO_BasicId` und `AO_BasicBase` stellen dabei die Grundfunktionalitäten zur Verwaltung der Baumstruktur zur Verfügung. Klassen die direkt von `AO_BasicId` abgeleitet werden stellen dabei Blätter des Baumes dar. Klassen wie `AO_Joint` (Gelenke) und `AO_Transform` (Basistransformation und starre Verknüpfungen) repräsentieren jedoch Knoten des Baumes, sie sind von der Basisklasse `AO_BasicBase` abgeleitet, die die Funktionalität eines Knotens implementiert.

Zur Visualisierung des Modells dient die Klasse `AO_BasicDrawer`. Sie definiert eine Schnittstelle zum Zeichnen von dreidimensionalen Punkten und Linien. Die beiden Klassen `AO_KGreyImageDrawer` und `AO_OpenGLDrawer` implementieren diese

Schnittstelle und leiten die Ausgabe auf ein Bitmap (`KGreyImage`) oder an die Zeichenfläche eines OpenGL - Fensters weiter. Es ist damit möglich, durch einfaches Austauschen der von
`AO_BasicDrawer` abgeleiteten Klassen, die Ausgabe auf ein anderes Medium umzuleiten.

6.4. Erstellen und Lösen des Gleichungssystems

Die im Kapitel 4 entwickelten Gleichungen werden in einer Methode der `AO_Solver` Klasse generiert. Dabei werden die Korrespondenzen zwischen dem Modell und der Szene durch eine Matrix vorgegeben. Ihre Spalten und Zeilen entsprechen jeweils einem Trackingpunkt des Models bzw. einem Merkmal in der Szene. Ein Eins-Eintrag steht für eine Korrespondenz, eine Null für keine Korrespondenz.

Nun wird eine Systemmatrix generiert, deren Spalten mit den sechs Basisparameteren und der Anzahl der im Modell vorhandenen Gelenke übereinstimmt. Liegt eine Korrespondenz vor, so werden zunächst die Einträge der Systemmatrix berechnet, die zur Ermittlung der Basistransformation nötig sind. Anschließend wird untersucht von welchen Gelenken der Modellpunkt abhängig ist. Die entsprechenden Spalten der Systemmatrix können anschließend mit den im Kapitel 4 berechneten Koeffizienten initialisiert werden. Alle nicht initialisierten Einträge werden mit Null vorbelegt.

Nachdem auch die Einträge des Vektors auf der rechten Seite des Gleichungssystems entsprechend den Zwangsbedingungen belegt worden sind, kann mit der nächsten Korrespondenz fortgefahren und das Gleichungssystem weiter aufgebaut werden.

Am Ende wird dann das Householder Verfahren [32] eingesetzt um das lineare Ausgleichsproblem zu lösen. Die ermittelten Parameter werden zunächst zwischengespeichert und können durch eine entsprechende Methode der `AO_Solver` Klasse in das Modell übertragen werden. Durch mehrfaches Lösen und entsprechendes Anpassen der Modellkonfiguration erfolgt schließlich die iterative Annäherung an die vorgegebene Pose.

Bemerkung 6.1 *Bei der Exponentialbeschreibung erfolgt die Parametrisierung der Gelenkachsen im Bezug zu einem Ursprungskoordinatensystem. Dies hat zur Folge, dass nach jedem Schritt der Iteration die Parameter der Gelenkachsen neu berechnet werden müssen, da sich ihre Lage durch den zuvor durchgeführten Iterationsschritt verändert hat.*

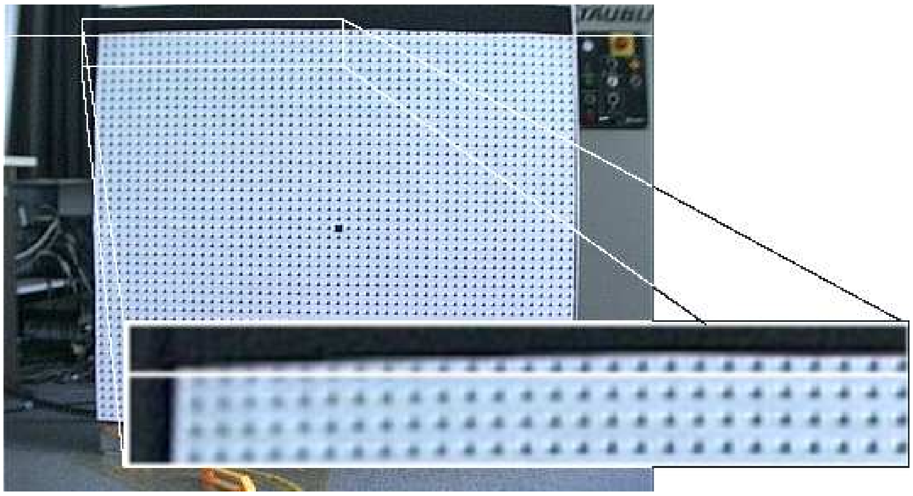


Abbildung 25: Bildverzerrungen durch Weitwinkelobjektiv

7. Experimente

7.1. Untersuchung der Kamera

Bei den Versuchen wird eine SONY Hyper HAD Kamera [30] benutzt und mit einer Auflösung von 384 × 288 Bildpunkten betrieben. Da die Experimente im Robotiklabor durchgeführt werden, ist ein 12 mm Weitwinkelobjektiv erforderlich, um die Szene mit dem Labormanipulator [31] vollständig aufnehmen zu können. Dies hat zur Folge, dass es im Bild zu starken Verzerrungen kommt, die nicht durch lineare Transformationen kompensiert werden können. Die Abbildung 25 zeigt die Aufnahme eines 2 cm-Punktrasters aus einer Entfernung von 2 m. Im vergrößerten Ausschnitt sind die Abweichungen von bis zu zwei Pixeln deutlich sichtbar. Rechnet man die Pixelabweichung in Weltkoordinaten um, so entspricht das einer Abweichung von 1 cm in der Szene (bei einem Abstand von 2 m von der Kamera). Betrachtet man hingegen die Abstände des Punktrasters, so erhält man bei gleichem Abstand eine Auflösung von

$$\frac{2 \text{ cm}}{5 \text{ Pixel}} = 0,4 \frac{\text{cm}}{\text{Pixel}}.$$

Die Abweichungen, die durch Linsenverzerrungen entstehen, betragen somit mehr als das Doppelte der eigentlichen Kameraauflösung.

Bei den nachfolgenden Versuchen hatte die Kamera einen Abstand von ca. 3 m zum Objekt. Bei dieser Entfernung beträgt die Auflösung ca.

$$\frac{2 \text{ cm}}{3,7 \text{ Pixel}} = 0,54 \frac{\text{cm}}{\text{Pixel}}.$$

Die Linsenverzerrung verursacht an den Rändern eine Abweichung von bis zu zwei Pixeln (1,8 cm in Weltkoordinaten). Rechnet man diese Ungenauigkeit auf ein

Gelenkglied mit einer Länge von 50 cm um (dies entspricht in etwa der Länge eines Gelenkgliedes des in den Versuchen verwendeten Manipulators), so ergibt sich eine Winkelabweichung von ±2,29 Grad. Da die Linsenverzerrung bei den Versuchen vernachlässigt wird, entstehen in den Laborversuchen entsprechende Abweichungen. Die Experimente werden deshalb zunächst in einer Simulation durchgeführt, bei der entsprechende Störungen ausgeschlossen sind.

7.2. Untersuchung des Konvergenzverhaltens

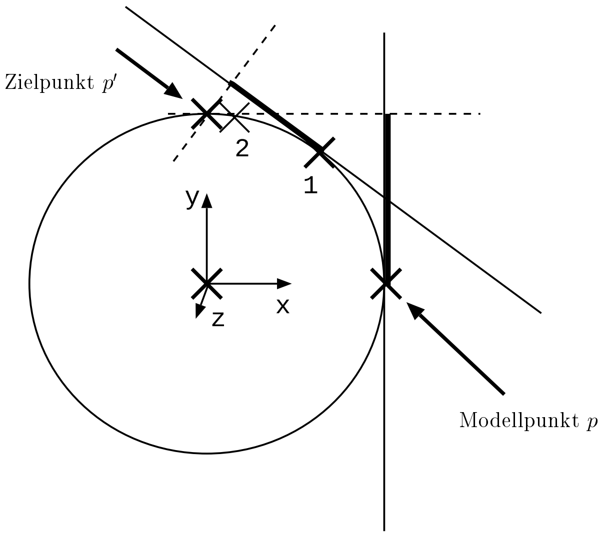


Abbildung 26: Untersuchung des Verfahrens durch vorgegebene Drehwinkel

Bei der Entwicklung der Gleichungssysteme wird eine Taylor - Approximation durchgeführt, durch die die kinematischen Abhängigkeiten lokal linearisiert werden. Die Bestimmung der Bewegungsparameter erfolgt erst durch eine iterative Annäherung an die vorgegebene Pose. Die Genauigkeit der Ergebnisse hängt somit von der Anzahl der Iterationen und der Größe der durchgeführten Bewegung ab. Im folgenden Versuch wird dieser Zusammenhang genauer untersucht.

Bei der Untersuchung wird von einem dreidimensionalen Punkt $p = [1 \ 0 \ 0]^T$ ausgegangen, der um die z-Achse des Koordinatensystems (s. Abb. 26) rotiert. Die Bewegung lässt sich in diesem Fall durch die Elemente aus $so(3)$ beschreiben und die Poseschätzung reduziert sich auf die Berechnung des Rotationswinkels. Diese Vereinfachung ermöglicht es, die Poseschätzung mit einer einzigen Punktkorrespondenz durchzuführen. Der Versuch bleibt dadurch unabhängig von der Geometrie des Modells.

Um die Konvergenz des Verfahrens zu untersuchen wird der Punkt p um den Winkel α rotiert:

$$p' = [\cos(\alpha) \quad \sin(\alpha) \quad 0]^T. \tag{7.100}$$

Ziel ist es nun, durch die Beziehung

$$p' = e^{\hat{\omega}\theta} p \approx_{Taylor} (I_{3\times 3} + \hat{\omega}\theta)p, \tag{7.101}$$

den Rotationswinkel $\alpha = \theta$ iterativ aus der Position von p und p' zu bestimmen. In Abbildung 27 sind die Ergebnisse der durchgeführten Untersuchung in einem Diagramm zusammengefasst. Jede der Kurven stellt den vorgegebenen und ermittelten

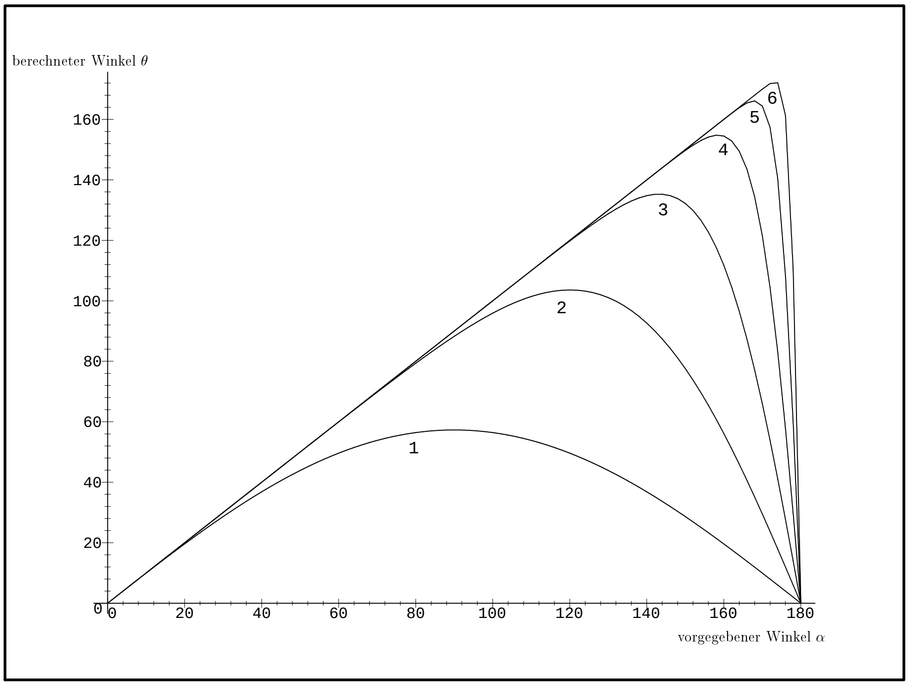


Abbildung 27: Untersuchung des Konvergenzverhaltens bei vorgegebenem Drehwinkel; Die Kurven geben den berechneten Winkel θ bei vorgegebenem Winkel α wieder. Die Zahlen entsprechen den durchgeführten Iterationen.

Winkel nach einer festen Anzahl von Iterationen gegenüber. Das Diagramm zeigt deutlich, wie sich die Kurven bei einer Erhöhung der Iterationsschritte an die optimale Funktion (entspricht der Identität) annähern. Bei großen Rotationen lässt die Konvergenzgeschwindigkeit nach. Drehungen von 180 Grad können nicht ermittelt werden. Dieses Verhalten wird verständlich, wenn man sich die Tangentialmenge im Punkt p betrachtet, auf der die Optimierung durchgeführt wird. Bei einer Drehung um 180 Grad liegt die Tangentialmenge senkrecht zum Differenzvektor von p und p'. Der minimale Abstand der Tangente zum Punkt p' ist deshalb schon im Punkt p selbst erreicht und das entspricht einer Winkeländerung von $\theta = 0$.

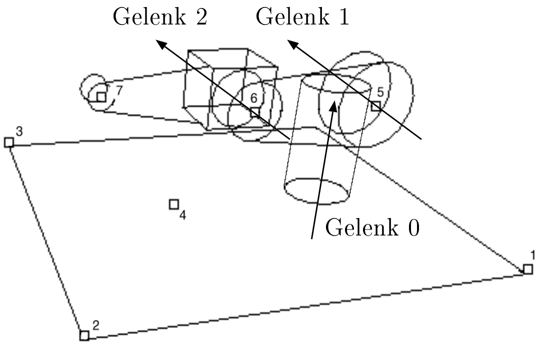
7.3. Poseschätzung mit der Punkt zu Punkt Bedingung

In den beiden folgenden Experimenten wird die Poseschätzung mit der in Abschnitt 4.1 entwickelten Punkt zu Punkt Bedingung untersucht. Bei der Untersuchung des

Konvergenzverhaltens (Abschnitt 7.2) konnte die Poseschätzung schon anhand einer einzigen Punktkorrespondenz durchgeführt werden, da sich das Experiment auf die Bestimmung des Rotationswinkels beschränkte. In den folgenden Experimenten werden auch die Parameter der Basistransformation bestimmt. Die Lösung setzt sich deshalb aus mindestens sechs Parametern zusammen. Da eine Punktkorrespondenz aber maximal drei lineare Gleichungen liefert, sind zur Durchführung einer Poseschätzung mehrere Modellpunkte nötigt. Eine von der Objektgeometrie unabhängige Untersuchung ist somit nicht möglich.

Als Objekt wird der im Labor vorhandene Manipulator [31] verwendet. Er besitzt sechs Freiheitsgerade und lässt sich mit einer Winkelgenauigkeit von 0.00072 bis 0.00220 Grad (Abhängig von dem jeweiligen Gelenk) ansteuern. Die Fehlerabweichungen können dadurch im wesentlichen auf das Verfahren zurückgeführt werden.

Zur Durchführung der Experimente wird zunächst eine VRML - Beschreibung des Laborroboters erzeugt. Die zur Posenschätzung verwendete Geometrie besteht aus 7 Punkten, die durch 3 Gelenke miteinander verbunden sind. Die genaue Anordnung der Punkte und Gelenke ist in der Abbildung 28 dargestellt.



Punkt	Koordinaten
1	(0,1500,32)
2	(1500,1500,32)
3	(1500,0,32)
4	(1000,750,32)
5	(400,1041,435)
6	(850,1041,435)
7	(1300,825,435)

Gelenk	$\xi = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$
0	(750,-400,0,0,0,1)
1	(-435,0,400,0,1,0)
2	(434,0,-850,0,1,0)

Abbildung 28: Modellgeometrie des Labormanipulators; die zur Poseschätzung relevanten Punkte sind durch kleine Quadrate markiert.

7.3.1. Experiment mit simulierten Daten

Das erste Experiment soll das Verfahren an rein künstlich erzeugten Daten untersuchen. Die dreidimensionalen Punktkoordinaten, aus denen die neue Pose bestimmt wird, wurden in diesem Fall direkt aus dem Modell berechnet. Störungen, wie die zuvor erwähnte Linsenverzerrung oder andere Ungenauigkeiten bei der Merkmalsextraktion, können dadurch ausgeschlossen werden.

Gestartet wird das Experiment in der initialen Position (das entspricht der in Abbildung 28 dargestellten Konfiguration). Die einzelnen Gelenke werden jeweils

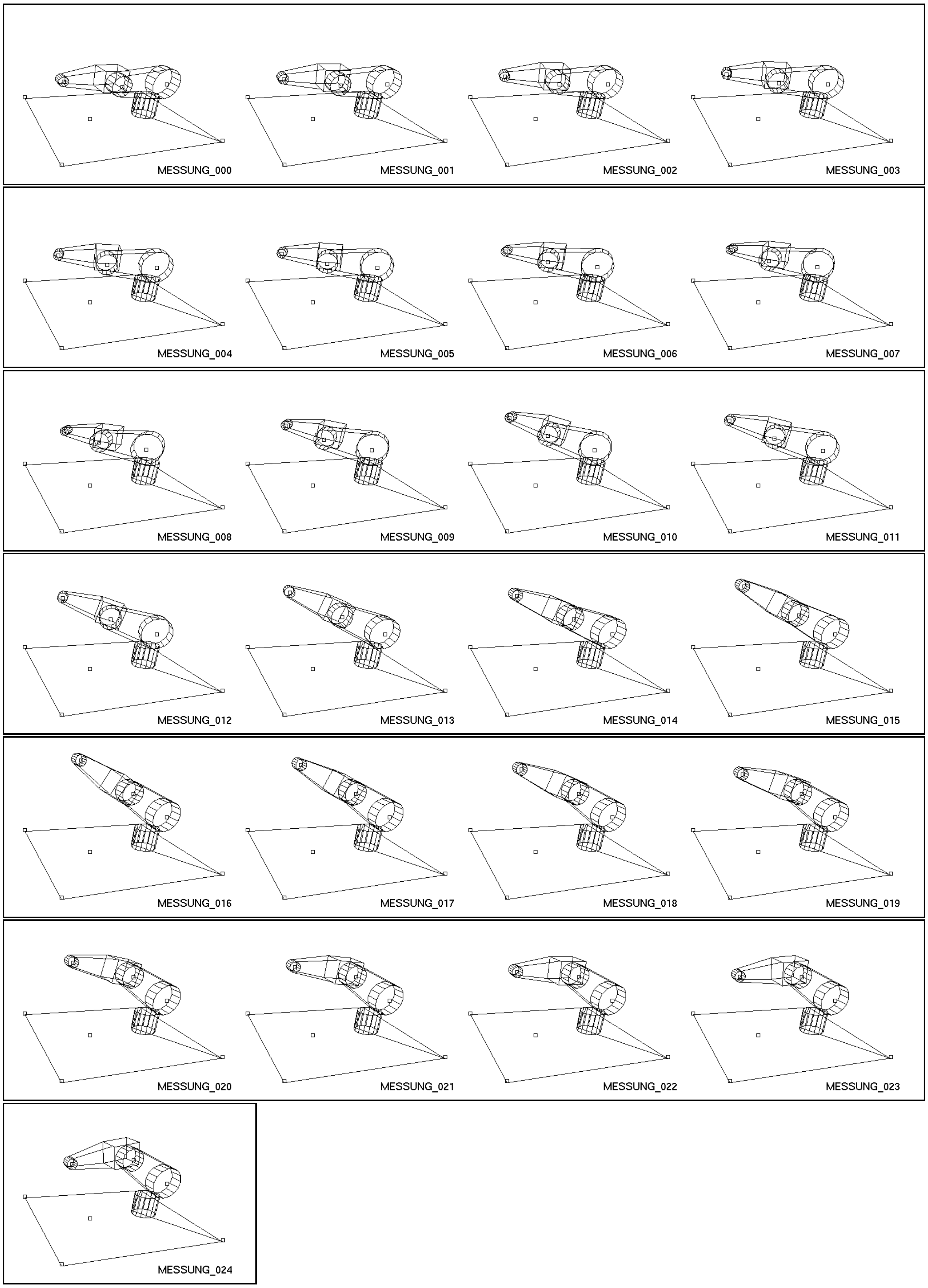


Abbildung 29: Visualisierung der vorgegebenen Bewegung (s. Tabellen 5, 7, 8 und 9)

um 5-10 Grad bewegt, so dass sich der in Abbildung 29 dargestellte Bewegungsablauf ergibt. Nach jeder Bewegung werden die Posenparameter mit den in Abschnitt 4.1 entwickelten Gleichungen bestimmt. Die Ergebnisse der Versuchsreihe sind in Tabelle 5 zusammengefasst. In der Tabelle sind die vorgegebenen und die nach einer Iteration ermittelten Winkel gegenübergestellt. Da sich die Gelenkwinkel zwischen dem Modell und der vorgegebenen Pose in dem Versuch nicht mehr als 10 Grad unterscheiden, sind bereits ab der zweiten Iteration die Abweichung kleiner 0.001 Grad.

Messung	Gelenk 0		Gelenk 1		Gelenk 2	
	vorgegeben	berechnet nach 1. Iter.	vorgegeben	berechnet nach 1. Iter.	vorgegeben	berechnet nach 1. Iter.
0	0	0,00	0	0,00	0	0,00
1	-5	-4,99	0	0,00	0	-0,02
2	-10	-9,99	0	0,00	0	-0,02
3	-15	-14,99	0	-0,00	0	-0,02
4	-20	-19,98	0	-0,00	0	-0,02
5	-25	-24,98	0	-0,01	0	-0,02
6	-30	-29,98	0	-0,01	0	-0,01
7	-35	-34,98	0	-0,01	0	-0,01
8	-40	-39,98	0	-0,01	0	-0,01
9	-35	-34,98	-5	-4,99	0	-0,03
10	-35	-34,97	-10	-9,99	0	-0,01
11	-25	-24,96	-10	-10,15	0	-0,03
12	-20	-20,02	-15	-15,02	0	-0,05
13	-10	-10,15	-20	-20,19	0	-0,11
14	0	0,04	-20	-20,21	0	-0,05
15	0	0,04	-25	-24,97	0	-0,01
16	0	0,04	-25	-24,97	0	-0,01
17	0	0,03	-30	-29,96	0	-0,00
18	0	0,00	-30	-29,99	-5	-5,00
19	0	-0,00	-30	-30,01	-10	-10,03
20	0	-0,00	-30	-30,02	-15	-15,05
21	0	-0,00	-30	-30,04	-20	-20,07
22	0	-0,01	-30	-30,05	-25	-25,09
23	0	-0,01	-30	-30,06	-30	-30,11
24	0	-0,01	-30	-30,07	-35	-35,12
25	0	-0,01	-30	-30,08	-40	-40,13

Tabelle 5: Messergebnisse zur Punkt zu Punkt Bedingung mit simulierten Daten; Die Tabelle zeigt die nach einer Iteration ermittelten Gelenkwinkel. Auf die Angabe der Ergebnisse weiterer Iterationen wird verzichtet, da die Abweichungen hier kleiner als 0,001 Grad sind.

Zur Betrachtung des Konvergenzverhaltens wird das Experiment mit einer größeren Winkeldifferenz zwischen dem Modell und der vorgegebenen Pose durchgeführt. In diesem Fall unterscheiden sich die Gelenkwinkel um jeweils 45 Grad von der initialen Pose. Während des Versuchs werden nach jedem Iterationsschritt die ermittelten Gelenkwinkel festgehalten, so dass in der Tabelle 6 die Annährung an die vorgegebene Pose verfolgt werden kann.

Iteration	Gelenk 0	Gelenk 1	Gelenk 2
	berechnet	berechnet	berechnet
0	0,00	0,00	0,00
1	21,10	-40,06	10,73
2	36,68	-47,59	36,85
3	44,85	-45,07	45,06
4	45,00	-45,00	45,00

Tabelle 6: Messergebnisse zur Punkt zu Punkt Bedingung mit simulierten Daten; Die Tabelle zeigt die Ergebnisse einer Iterationfolge von der initialen Position (alle Winkel 0,0 Grad) zur Konfiguration mit den Gelenkwinkeln (45,0; -45,0; 45,0) Grad.

7.3.2. Experiment auf realen Bilddaten

Um das Verfahren mit realen Bilddaten durchzuführen, müssen zunächst dreidimensionale Punktkorrespondenzen gewonnen werden. In diesem Fall werden die Daten von zwei Videokameras benutzt, um die dreidimensionalen Informationen zu rekonstuieren (s. Abb. 30). Vor der Durchführung wurde der Laborroboter mit blauen Farbmarken präpariert, um die automatische Detektion von Objektpunkten zu vereinfachen (s. Anhang B.1). Die Kamerakalibrierung erfolgt unter Berücksichtigung der in Abschnitt 6 eingeführten Beziehungen und liefert für jede Kamera eine entsprechende Projektionsmatrix. Radiale Linesenverzerrungen bleiben bei diesem Verfahren unberücksichtigt.

Die dreidimensionalen Punkte des Objektmodells werden nun in die beiden Kamerabilder projiziert. Um die projizierten Punkte werden Suchbereiche im Bild der Kamera definiert, in denen die Farbmarkierungen mit dem im Anhang B.1 beschriebenen Verfahren extrahiert werden. Die neue Position der Markierungspunkte wird durch den Schwerpunkt der im Suchbereich segmentierten Fläche bestimmt. Nachdem die Punkte in den Kamerabildern lokalisiert sind, erfolgt die Bestimmung der dreidimensionalen Punktkoordinaten. Dazu werden zu zwei korrespondierenden Bildpunkten (jeweils einer aus der linken und rechten Kamera) die Sehgeraden ermittelt. Die dreidimensionalen Punktkoordinaten werden dann nach dem Prinzip des führenden Auges [11] berechnet. Die rekonstruierten dreidimensionalen Punktkoordinaten entsprechen also dem Punkt auf dem Sehstrahl der *führenden Kamera*, der den geringsten Abstand zum Sehstrahl der zweiten Kamera besitzt (s. Abbildung 30).

Mit Hilfe der gewonnenen Informationen wird nun die eigentliche Poseschätzung durchgeführt. Diese erfolgt iterativ durch mehrfaches Lösen des in Abschnitt 4.1 entwickelten Gleichungssystems und der entsprechenden Anpassung des Objektmodells. Die Ergebnisse des Versuchs sind in der Tabelle 7 dargestellt. Die dort aufgeführten berechneten Werte ergeben sich nach 20 Iterationsschritten und stellen den Grenzwert der Iteration dar.

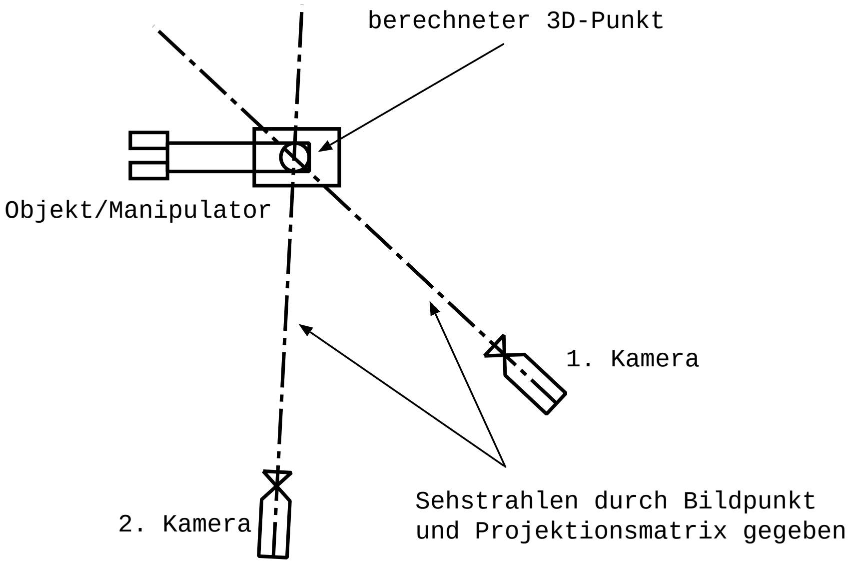


Abbildung 30: Versuchsaufbau zur Untersuchung der Punkt zu Punkt Bedingung auf realen Bilddaten; der dreidimensionale Punkt ergibt sich durch Minimierung des Abstandes auf dem Sehstrahl der *führenden* Kamera zum Sehstrahl der zweiten Kamera.

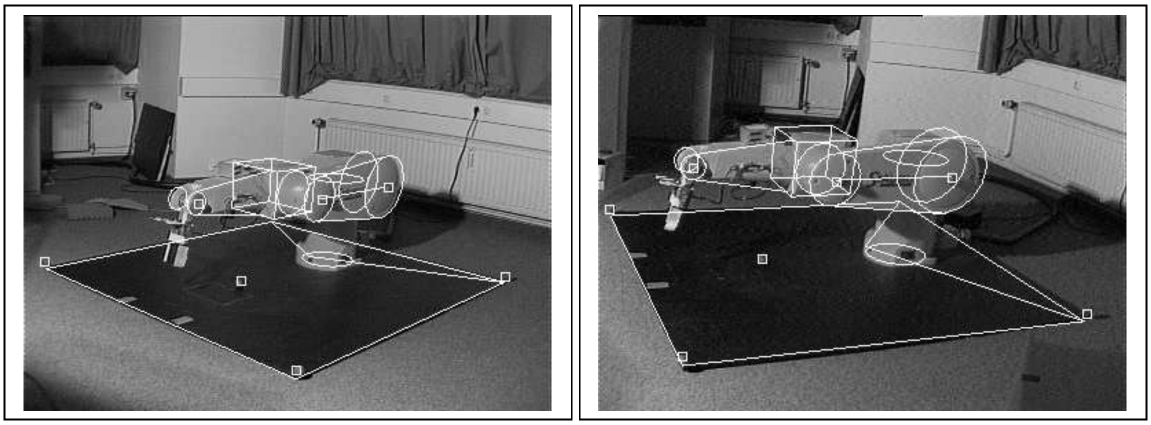


Abbildung 31: Kamerabilder der linken und rechten Kamera mit projiziertem Objektmodell

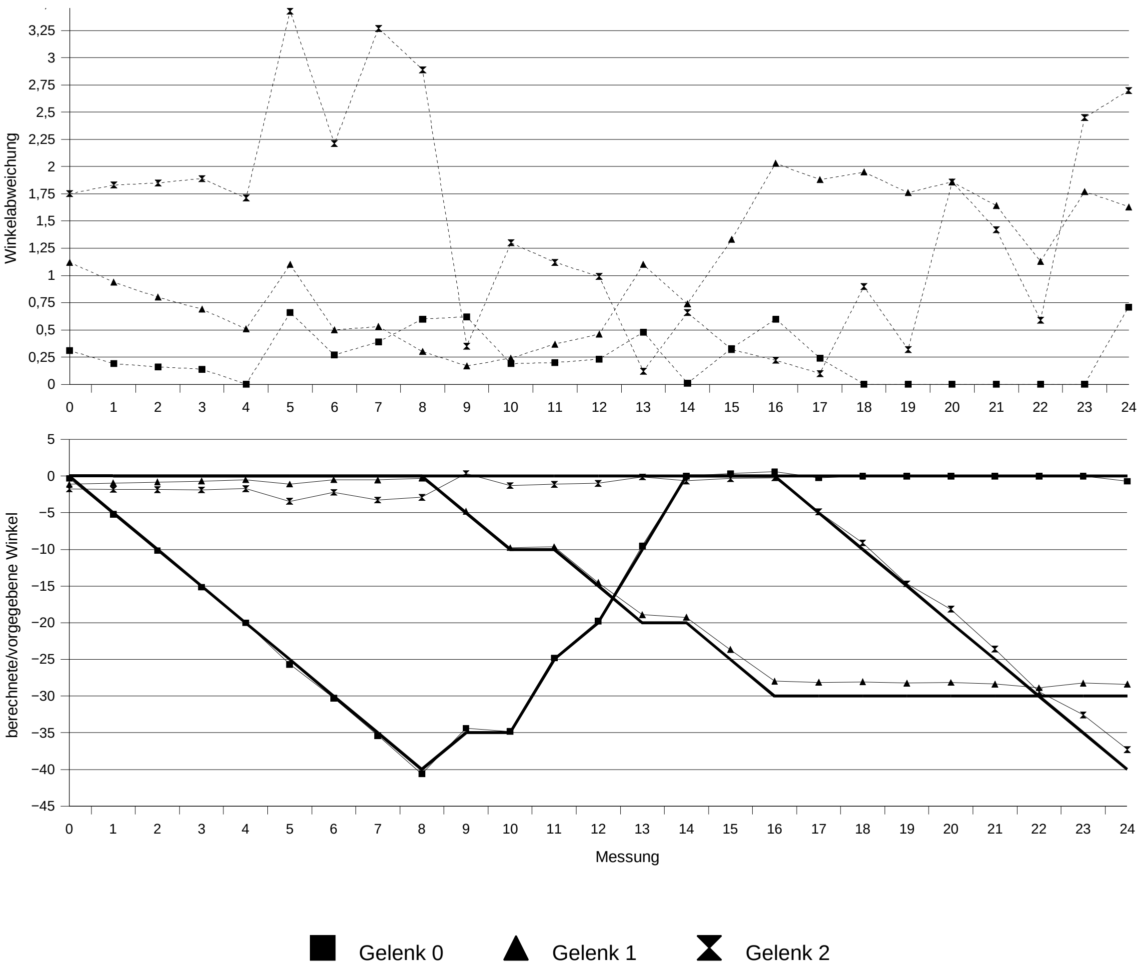


Abbildung 32: Fehlerdiagramm zu Tabelle 7. Die gestrichelten Linien geben die Winkelabweichungen, die dicken Linien die vorgegebenen Winkel und die dünnen Linien die berechneten Winkel wieder. Im dem Diagramm ist deutlich zu erkennen, dass die Winkelabweichung größer ist, wenn sich Teile des Manipulators von der Kamera entfernen (Messung 5-8) oder am Rand des Kamerabildes befinden (Messung 15-24).

Messung	Gelenk 0		Gelenk 1		Gelenk 2	
	vorgegeben	berechnet nach 20. Iter.	vorgegeben	berechnet nach 20. Iter.	vorgegeben	berechnet nach 20. Iter.
0	0	-0,31	0	-1,12	0	-1,75
1	-5	-5,19	0	-0,94	0	-1,83
2	-10	-10,16	0	-0,80	0	-1,85
3	-15	-15,14	0	-0,69	0	-1,89
4	-20	-20,00	0	-0,51	0	-1,71
5	-25	-25,66	0	-1,12	0	-3,43
6	-30	-30,27	0	-0,50	0	-2,21
7	-35	-35,39	0	-0,53	0	-3,27
8	-40	-40,60	0	-0,30	0	-2,89
9	-35	-34,38	-5	-4,83	0	0,35
10	-35	-34,81	-10	-9,76	0	-1,30
11	-25	-24,89	-10	-9,63	0	-1,12
12	-20	-19,77	-15	-14,54	0	-0,99
13	-10	-9,52	-20	-18,90	0	-0,12
14	0	-0,01	-20	-19,26	0	-0,66
15	0	0,33	-25	-23,67	0	-0,32
16	0	0,60	-30	-27,97	0	-0,22
17	0	-0,24	-30	-28,12	-5	-4,90
18	0	-0,00	-30	-28,05	-10	-9,10
19	0	-0,00	-30	-28,24	-15	-14,68
20	0	0,00	-30	-28,14	-20	-18,14
21	0	0,00	-30	-28,36	-25	-23,58
22	0	0,00	-30	-28,87	-30	-29,41
23	0	0,00	-30	-28,23	-35	-32,55
24	0	-0,71	-30	-28,37	-40	-37,30

Tabelle 7: Messergebnisse des Laborversuchs zur Punkt zu Punkt Bedingung

Im Vergleich zu den in der Simulation erreichten Messergebnissen weichen die berechneten Winkel im Laborversuch von den vorgegebenen Winkeln ab. Die maximalen Abweichungen liegen dabei zwischen 0,71 Grad am Gelenk 0 und 3,43 Grad am Gelenk 2. Dies entspricht in etwa der Grössenordung, die bei der Kameraunteruntersuchung (s. Abschnitt 7.1) ermittelt wurde. Dass die maximale Abweichung am Gelenk 2 etwas höher ausfällt, liegt an der Verteilung der Markierungspunkte. Am letzten Glied des Manipulators befindet sich nur ein Markierungspunkt, so dass auftretende Fehler durch das dritte Gelenk kompensiert werden können ohne an einem anderen Punkt den Fehler zu vergrößern. Deutlich sichtbar wird dieses Verhalten auch im Fehlerdiagramm (s. Abb. 32).

7.4. Poseschätzung mit der Punkt zu Sehgerade Bedingung

Im folgenden Experiment soll die Poseschätzung mit der Punkt zu Sehgerade Bedingung erfolgen. Auch in diesem Fall werden zwei Versuche durchgeführt. Zunächst wird das Verfahren wieder ohne Störungen des Trackingverfahrens untersucht. Danach demonstriert ein Laborversuch mit realen Kameradaten den praktischen Einsatz.

Bemerkung 7.1 *Die Versuche dieses Abschnitts demonstrieren die Möglichkeit einer monokularen Poseschätzung. In den Versuchen wird die dreidimensionale Konfiguration des Modells aus den zweidimensionalen Daten einer einzelnen Kamera rekonstuiert. Dies ist nur durch Ausnutzung der bekannten Objektgeometrie möglich.*

7.4.1. Versuch mit simulierten Daten

Die zur Poseschätzung verwendeten Daten werden, wie im simulierten Versuch der Punkt zu Punkt Bedingung, durch die von einem Referenzmodell vorgegebenen Punktkoordinaten geliefert. Das Referenzmodell nimmt dabei wieder nacheinander die in Abbildung 29 dargestellen Konfigurationen ein.

In diesem Experiment werden jedoch nicht die dreidimensionalen Punktkoordinaten benutzt, sondern ihre durch eine Projektionsmatix bestimmten Projektionsgeraden. In dem Versuch wird dabei die gleiche Projektionsmatrix wie im nachfolgenden Laborversuch 7.4.2 verwendet. Das hier durchgeführte Experiment entspricht damit einer Poseschätzung mit einer einzelnen idealisierten Kamera (keine Linsenverzerrung).

Mit den so gewonnenen Geradenparametern wird das Gleichungssystem nach dem in Abschnitt 4.2 entwickelten Schema generiert und die freien Parameter bestimmt. In der Tabelle 8 sind die Ergebnisse des Versuchs festgehalten.

Messung	Gelenk 0		Gelenk 1		Gelenk 2	
	vorgegeben	berechnet nach 1. Iter.	vorgegeben	berechnet nach 1. Iter.	vorgegeben	berechnet nach 1.Iter.
0	0	-0,00	0	0,00	0	0,00
1	-5	-4,77	0	-0,13	0	-0,06
2	-10	-9,76	0	-0,14	0	-0,09
3	-15	-14,81	0	-0,12	0	-0,11
4	-20	-19,90	0	-0,09	0	-0,12
5	-25	-25,01	0	-0,04	0	-0,11
6	-30	-30,11	0	0,00	0	-0,09
7	-35	-35,18	0	0,03	0	-0,06
8	-40	-40,21	0	0,05	0	-0,04
9	-35	-35,53	-5	-4,83	0	0,01
10	-35	-35,24	-10	-9,92	0	-0,00
11	-25	-25,86	-10	-10,01	0	-0,38
12	-20	-20,22	-15	-15,07	0	-0,21
13	-10	-9,88	-20	-20,44	0	-0,37
14	0	0,59	-20	-20,40	0	-0,04
15	0	0,24	-25	-24,99	0	0,05
16	0	0,28	-30	-29,97	0	0,07
17	0	0,08	-30	-29,99	-5	-4,98
18	0	0,07	-30	-30,01	-10	-9,99
19	0	0,07	-30	-30,02	-15	-15,00
20	0	0,07	-30	-30,03	-20	-20,01
21	0	0,06	-30	-30,04	-25	-25,02
22	0	0,06	-30	-30,05	-30	-30,04
23	0	0,05	-30	-30,06	-35	-35,05
24	0	0,04	-30	-30,07	-40	-40,07

Tabelle 8: Messergebnisse zur Punkt zu Sehgerade Bedingung mit simulierten Daten; Die Tabelle zeigt die nach einer Iteration ermittelten Gelenkwinkel. Auf die Angabe der Ergebnisse mit mehr Iterationen wurde verzichtet, da auch bei diesem Versuch die nach der 2. Iterationen ermittelten Abweichungen kleiner als 0,001 Grad sind.

7.4.2. Versuch auf realen Bilddaten

Dieser Versuch untersucht das Verfahren in einer realen Umgebung. Wie in Abschnitt 7.3.2 wird eine Farbsegmentierung durchgeführt, um die am Labormanipulator angebrachten Farbmarkierungen zu detektieren. Die Zuordnung der Modellpunkte zu den segmentierten Markierungen erfolgt wieder durch die Festlegung eines Suchfensters um die ins Bild projizierten Modellpunkte. Die Bestimmung der zweidimensionalen Position der Punkte erfolgt erneut durch eine Schwerpunktberechnung. Der Ablauf der Merkmalsextraktion ist in der Abbildung 33 zu sehen. In den Bildern auf der linken Seite sind die segmentierten Farbbereiche mit ihren Schwerpunkten eingezeichnet. Die Quadrate zeigen den Suchbereich an, in denen nach den Markierungen gesucht wird.

Aus den zweidimensionalen Punktkoordinaten und der Projektionsmatrix werden die Parameter der entsprechenden Projektionsgeraden berechnet und dann direkt zur Generierung der Gleichungen nach Abschnitt 4.2 genutzt.

Die Durchführung des Experiments erfolgt wieder durch abwechselndes Einstellen und Berechnen der in Abbildung 29 dargestellten Gelenkpositionen. In der Tabelle 9 sind die realen und nach 20 Iterationen ermittelten Werte der Versuchsreihe gegenübergestellt.

Messung	Gelenk 0		Gelenk 1		Gelenk 2	
	vorgegeben	berechnet nach 20 Iter.	vorgegeben	berechnet nach 20 Iter.	vorgegeben	berechnet nach 20 Iter.
0	0	0,83	0	1,30	0	0,92
1	-5	-4,33	0	1,36	0	1,20
2	-10	-9,07	0	2,28	0	2,99
3	-15	-14,11	0	0,77	0	-0,17
4	-20	-18,79	0	0,84	0	0,33
5	-25	-23,93	0	0,76	0	-0,25
6	-30	-28,59	0	0,91	0	0,04
7	-35	-33,69	0	0,83	0	-0,40
8	-40	-38,93	0	1,12	0	0,20
9	-35	-34,22	- 5	-3,50	0	0,41
10	-35	-34,37	-10	-8,46	0	-0,13
11	-25	-24,55	-10	-8,55	0	-0,09
12	-20	-19,42	-15	-13,48	0	0,25
13	-10	-9,40	-20	-18,36	0	-0,13
14	0	0,85	-20	-18,48	0	-0,10
15	0	0,95	-25	-23,08	0	-0,02
16	0	1,52	-30	-27,60	0	-0,14
17	0	1,05	-30	-27,45	-5	-4,48
18	0	0,81	-30	-27,88	-10	-9,71
19	0	0,02	-30	-27,49	-15	-13,65
20	0	0,87	-30	-27,67	-20	-18,48
21	0	0,69	-30	-27,56	-25	-23,33
22	0	0,90	-30	-27,47	-30	-27,55
23	0	0,02	-30	-27,62	-35	-32,62
24	0	0,70	-30	-27,82	-40	-37,61

Tabelle 9: Messergebnisse zur Punkt zu Sehgerade Bedingung (3 Gelenke)

Der Vergleich von Simulation und Laborexperiment lässt wieder eine Lokalisation der Fehlerquelle zu. Die Abweichungen des Laborexperiments liegen auch hier im Bereich der Winkelabweichungen, die bei der Kamerauntersuchung ermittelt

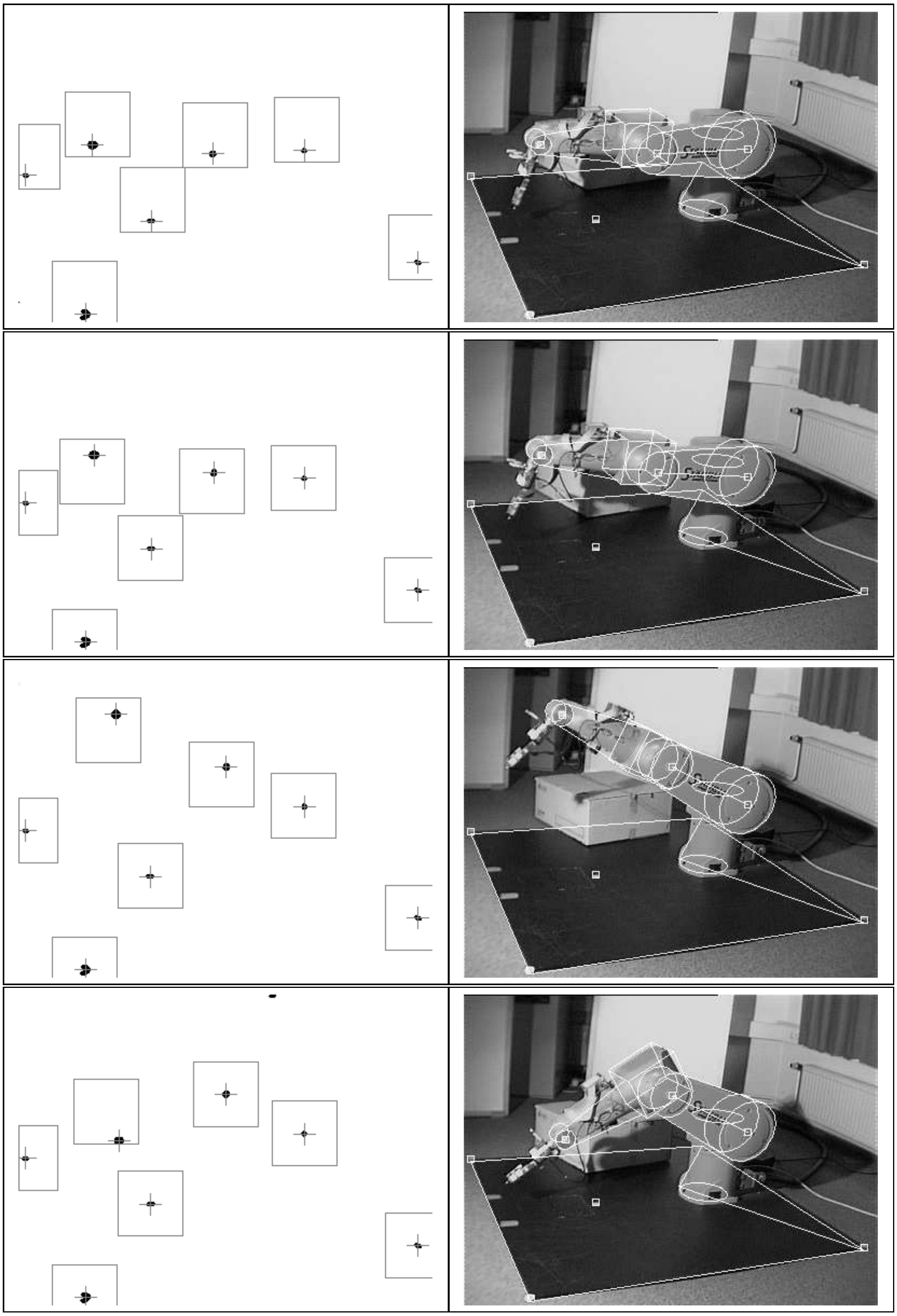


Abbildung 33: Aufnahme des Laborroboters mit projiziertem Objektmodell; links: Suchfenster mit detektierten Farbmarken; rechts: Bildaufnahme mit projiziertem Objektmodell

wurden. In der Simulation konnten hingegen schon nach wenigen Iterationen keine Fehler mehr festgestellt werden. Die festgestellten Abweichungen sind damit im Wesentlichen auf die Vernachlässigung der Linsenverzerrung zurückgeführt.

7.5. Roboternavigation mit der Punkt zu Sehebene Bedingung

Die Punkt zu Sehebene Bedingung setzt voraus, dass man im Kamerabild Geraden detektieren kann, aus denen dann die benötigten Ebenen berechnet werden können. Die zur Geradendetektion verwendete Hough-Transformation (s. Anhang B.2) funktioniert jedoch nur auf Bildern mit deutlich ausgeprägten Geradenstrukturen. Die zuvor verwendete Szene mit dem Labormanipulator ist in diesem Fall nicht geeignet.

Das folgende Experiment unterscheidet sich deshalb von den vorhergehenden. Der Schwerpunkt liegt hier nicht auf der Untersuchung des Konvergenzverhaltens und der Betrachtung der Fehlerabweichung. Der Versuch soll vielmehr demonstrieren, dass sich die vorgestellten Methoden auch zu Roboternavigation eignen.

Zur Durchführung des Experiments wird zunächst ein Drahtgitter-Modell einer Wand des Robotiklabors erstellt. In der Wand befindet sich eine Tür deren Geometrie und Gelenkachse ebenfalls im Modell festgehalten ist. Als nächstes wird eine

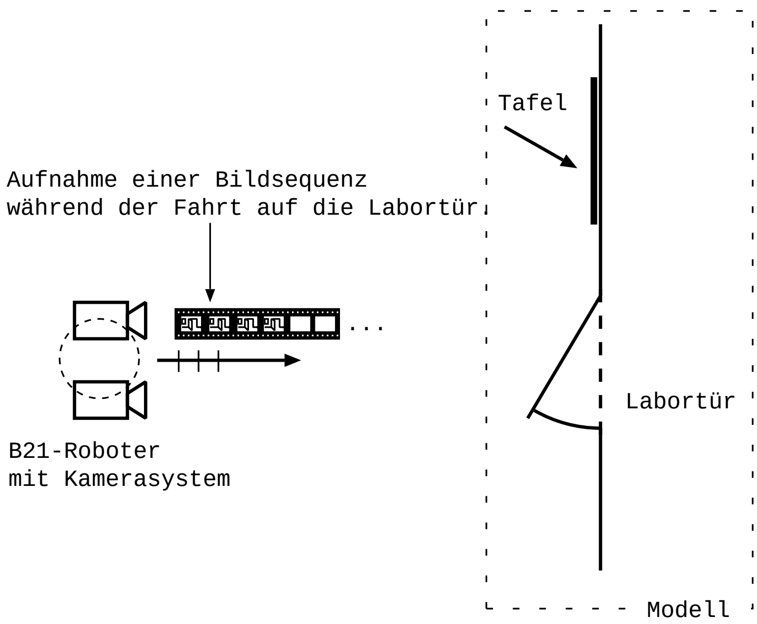


Abbildung 34: Versuchsanordnung beim Experiment zur Punkt zu Sehebene Bedingung

Kamera des B21-Robotors [24] kalibriert und eine Bildsequenz aufgenommen. Der Roboter bewegt sich dabei auf die Labortür zu. Die Abbildung 34 veranschaulicht den Versuch anhand einer Skizze.

Aufgabe ist es, Position und Lage des Modells an die realen Bilddaten anzupassen und den Öffnungswinkel der Labortür zu ermitteln. Aus den berechneten Transformationsparametern kann dann die aktuelle Position des Roboters bestimmt werden. Anhand des Öffnungswinkels lässt sich entscheiden, ob eine Fahrt durch die Tür möglich ist.

Zunächst werden die Geradeninformationen aus den Bilddaten benötigt. Sie werden mit der in [1] implementierten und erweiterten Hough-Transformation ermittelt.

Die detektierten Geraden müssen nun Punkten/Geraden des Modells zugeordnet werden. Dieses Korrespondenzproblem wird durch einen Algorithmus gelöst, der von Daniel Grest implementiert wurde. Eine detailierte Beschreibung des Algorithmus kann in [9, 25] nachgelesen werden.

Der Algorithmus wählt jeweils einige Geraden aus und ordnet sie den Modellpunkten zu. Um die Suche nach der richtigen Korrespondenz zu beschleunigen, werden Annahmen über die durchgeführte Bewegung gemacht und damit Distanz- und Winkeleinschränkungen festgelegt, die den Suchraum einschränken.

Durch diese Heuristik existieren nur noch wenige mögliche Zuordnungen. Mit diesen wird dann die Poseschätzung durchgeführt. Dabei werden die in Abschnitt 4.3 oder die auf planare Bewegungen eingeschränkten Gleichungen aus Abschnitt 4.5 verwendet. Ist eine richtige Zuordnung gewählt worden, so können die Abstände zwischen den Modellpunkten und den aus Hough-Geraden resultierenden Ebenen minimiert werden. Bei einer falschen Zuordnung können die Abstände im allgemeinen nicht erfolgreich minimiert werden, da die vorgegebene Objektgeometrie die Anpassung an die falschen Korrespondenzen verhindert. Die Summe der Abweichungen von transformierten Modellpunkten und vorgegebenen Ebenen eignet sich somit als Fehlermaß und wird vom Algorithmus verwendet, um die richtigen Korrespondenzen von den falschen zu unterscheiden.

Bemerkung 7.2 *In einigen Fällen ist das gewählte Fehlermaß nicht ausreichend. Dies wurde auch bei den Versuchen deutlich. Das Verfahren wählte dann Korrespondenzen, bei denen das Modell zwar auf die extrahierten Geraden projiziert wird, die ermittelte Pose jedoch nicht mit der real vorliegenden Szene übereinstimmte (siehe auch Abschnitt 7.6).*

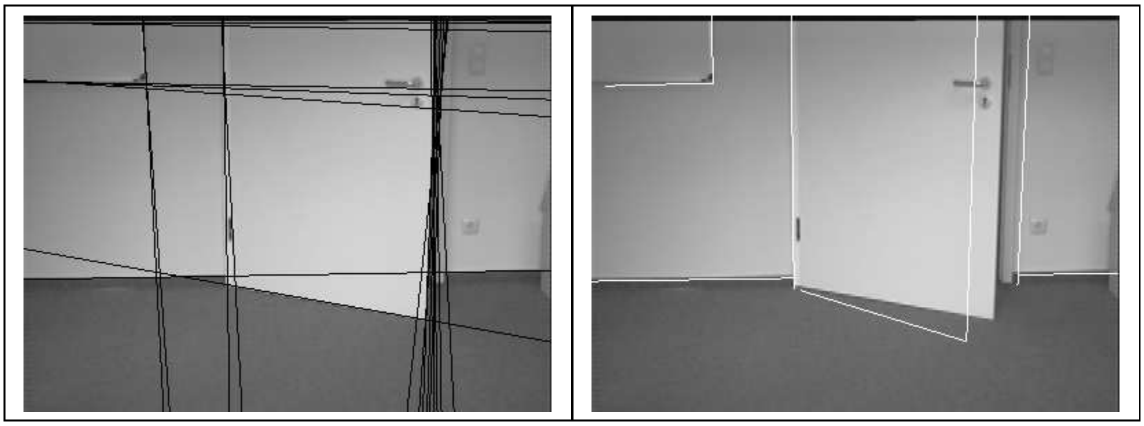


Abbildung 35: Grundlage der Punkt zu Sehebene Beziehung; links: extrahierte Hough-Geraden; rechts: geometrisches Modell der Laborumgebung

Die Abbildungen 35 und 36 zeigen, wie das Verfahren aus den Hough-Geraden und dem Modell (Abbildung 35 links und rechts) die entspechende Pose bestimmt (Abbildung 36 links). In dem rechten Bild von Abbildung 36 wird deutlich, dass auch der Öffnungswinkel der Tür berechnet und angepasst wird.

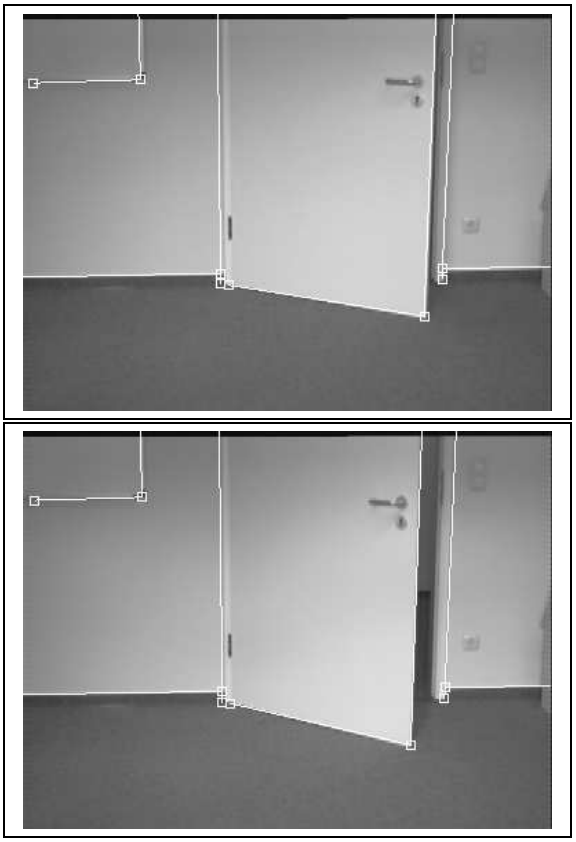


Abbildung 36: Zwei Beispiele in denen die richtigen Korrespondenzen gefunden und die Pose erfolgreich ermittelt wurde.

7.6. Ergebnisse

Die Versuche zeigen, dass mit den implementierten Verfahren gute Bewegungsschätzungen möglich sind. Das Experiment zum Konvergenzverhalten macht zwar deutlich, dass es Situationen gibt, in denen das Verfahren nicht konvergiert. Diese treten in der Praxis jedoch kaum auf. Insbesondere bei der Analyse von Bildfolgen sind die Bewegungen meistens klein genug, um die Zielkonfiguration zu erreichen. Das in den Bilddaten vorhandene Rauschen verhindert, dass sich das System längere Zeit in einem stationären Zustand befindet. Eine Rotation von exakt 180 Grad wird man in einer natürlichen Umgebung selten vorfinden.

Vergleicht man die Experimente auf den künstlich erzeugten Daten mit den Versuchen auf realen Bilddaten, so liegen die festgestellten Abweichungen zwischen 0 und 3 Grad. Diese Fehler sind im wesentlichen auf die Linsenverzerrung zurückzuführen, denn bei der Untersuchung der Kamera konnten bereits Abweichungen in dieser Größenordnung begründet vorhergesagt werden. Dass sich die Poseschätzung auch ohne Berücksichtigung der Linsenverzerrung auf realen Kamerabilden durchführen lässt macht deutlich, dass das Verfahren auch mit stark verzerrten Daten akzeptable Ergebnisse liefert.

Der Versuch zur Punkt zu Sehebene Bedingung zeigt die universelle Einsatzmöglichkeit des Verfahrens. Durch die Möglichkeit die Geometrie der Modelle in einer VRML-Datei beschreiben zu können ist das hier entwickelte System schnell an andere Situationen angepasst. Die Gleichungen zur Lösung der inversen Kinematik können durch das Verfahren automatisch aus dem Modell generiert werden, denn die entwickelten Gleichungen setzen keine spezielle Objektgeometrie voraus.

Probleme traten bei den Versuchen im Wesentlichen bei der Merkmalsextraktion und der Korrespondenzfindung auf. Bei der Detektion der Farbmarkierungen werden zum Beispiel Suchbereiche festgelegt, deren Positionen durch die projizierten Modellpunkte bestimmt werden. Liegen die Punkte im Kamerabild sehr nah zusammen, so können die einzelnen Punkte nicht mehr separiert werden. In diesem Fall liefert die Schwerpunktberechnung einen Mittelwert aus den beiden Punktkoordinaten zurück und die Ergebnisse der Poseschätzung werden verfälscht.

Im Versuch zur Roboternavigation wird das Korrespondenzproblem noch wesentlich deutlicher. Bei dem Experiment mit der Labortür wird zum Beispiel manchmal eine Konfiguration ermittelt, bei der die Tür einen Öffnungswinkel von fast 90 Grad hat. In dieser Stellung liegt das Türbatt in Deckung mit dem linken Türpfosten. Die gewählten Korrespondenzen entsprechen einer Konfiguration, die zwar aus Sicht der genutzten Informationen richtig ist, mit der real vorliegenden Situation jedoch nicht übereinstimmt.

Diese Ergebnisse zeigen, das sich das auftretende Korrespondenzproblem nicht in allen Fällen durch das Ausnutzen der Hough-Geraden lösen lässt. Die Versuche machen aber deutlich, dass die Implementation zur Selbstlokalisation eines Roboters eingesetzt werden kann. Die Berücksichtigung von Farb- und Texturinformationen, die auch in dem Modell gespeichert werden könnten, sollte das vorgestellte Verfahren deutlich stabiler machen.

Der Schwerpunkt dieser Arbeit liegt jedoch nicht in der Lösung des Korrespondenzproblems oder der Merkmalsextraktion. Im Mittelpunkt steht vielmehr die mathematische Modellierung von kinematischen Ketten, deren Kombination mit der projektiven Geometrie, und die Implementation der Verfahren in die objektorientierte Klassenbibliothek KiVi Grap.

Die verwendeten Vorverarbeitungsschritte sind daher einfach gehalten und die Anforderungen an die Szene entsprechend hoch. Durch die objektorientierte Struktur ist die Implementation zur Verwaltung der Objektgeometrie und die eigentliche Berechnung der Parameter unabhängig von der Vorverarbeitung und kann den Anforderungen entsprechend angepasst werden.

7.6.1. Geschwindigkeit

Bei den Experimenten konnte festgestellt werden, dass die Geschwindigkeit hauptsächlich von der Merkmalsextraktion und der Korrespondenzfindung beeinflusst wird. 10 Iterationen der eigentlichen Poseschätzung können auf einem AMD-K6-2 Prozessor mit 450Mhz mit ca. 10 Hz durchgeführt werden. Die Messungen fanden dabei ohne spezielle Compileroptimierungen[14] statt, so dass durch Optimierungen bei der Implementation und Übersetzung weitere Geschwindigkeitssteigerungen möglich sind.

In der Laborumgebung kann die Bewegungen des Manipulator bereits in Echtzeit verfolgt werden.

[14] Die Messung erfolgte mit einem durch den C++ Compiler GCC Version 3.0.1 [34] übersetzten Programm.

8. Diskussion und Ausblick

In diesem Abschnitt werden noch einmal die Ergebnisse zusammengefasst und im Kontext anderer Arbeiten betrachtet. In einem Ausblick werden dann Möglichkeiten zur Erweiterung der Verfahren angegeben, die Gegenstand weiterer Untersuchungen sein könnten.

8.1. Diskussion

Die Aufgabe dieser Arbeit war es, ein System zur Repräsentation und Verfolgung kinematischer Ketten zu entwickeln. Außerdem sollten die entwickelten Algorithmen implementiert und in die Klassenbibliothek KiVi-Grap [17] integriert werden.

Das Ergebnis dieser Arbeit ist eine universell einsetzbare Implementation, die es ermöglicht Orientierung, Lage und interne Konfiguration geometrisch bekannter Objekte zu ermitteln. Die Anpassungsfähigkeit wird dabei durch die Modellierung verschiedener Zwangsbedingungen im projektiven Raum und der Beschreibung von Objektgeometrien im VRML-Format erreicht.

Im Kapitel über mathematische Grundlagen wurden verschiedene Beschreibungsformen für starre Bewegungen vorgestellt und anhand von Beispielen Vor- und Nachteile der entsprechenden Repräsentationen demonstriert. Es stellte sich dabei heraus, dass zur Berechnung der inversen Kinematik Beschreibungen von Vorteil sind, die Rotationen durch Angabe von Rotationsachse und Winkel parametrisieren und dadurch Singularitäten vermeiden.

Für die Modellierung der kinematischen Ketten ist schließlich die Exponentialdarstellung gewählt worden[15]. Die Exponentialdarstellung stellt die Bewegungsgruppe $SE(3)$ durch die Repräsentanten ihrer Lie-Algebra $se(3)$ dar. Dadurch existiert einerseits eine Parametrisierung von Rotationen durch die Angabe von Rotationsachse und Winkel, andererseits besteht eine direkte Beziehung zu den Tangentialmengen der Bewegungsgruppe. Durch die Taylorapproximation kann dann sogar erreicht werden, dass sich die von der reellen Exponentialfunktion gewohnte Rechenregel $\exp(a)\exp(b) = \exp(a+b)$ anwenden lässt. Die Berechnung von kinematischen Ketten wird dadurch wesentlich vereinfacht, denn die antikommutative Hintereinanderausführung von Transformationen aus $SE(3)$ wird durch die Approximation in eine kommutative Addition der Elemente aus $se(3)$ überführt.

Die im projektiven Raum formulierten Zwangsbedingungen von Punkt zu Punkt, Punkt zu Gerade und Punkt zu Sehebene ermöglichen es verschiedene Informationen über die Szene auszunutzen, um die Poseschätzung durchzuführen.

Anhand der durchgeführten Experimente wird deutlich, wie sich die verschiedenen Zwangsbedingungen ausnutzen lassen. Im Experiment zur Punkt zu Punkt Beziehung wurde die Poseschätzung mit Hilfe eines Stereosystems durchgeführt. Der Versuch mit der Punkt zu Sehgerade Bedingung demonstriert die Möglichkeit einer

[15] In [26] wird gezeigt, dass sich die hier vorgestellten Methoden sehr kompakt in einer Clifford Algebra (*conformal geometric algebra* [21]) beschreiben lassen.

monokularen Posenschätzung. Der Einsatz des Verfahren bei der Roboternavigation nutzt schließlich die Möglichkeiten der Punkt zu Sehebenen Bedingung aus. Die aus den Geraden in der Bildebene resultierenden Sehebenen konnten durch die Beziehung direkt zur Bestimmung der Pose genutzt werden.

Im Vergleich zu Verfahren die nach dem *Teile und Herrsche* - Prinzip ein Objekt in seine starren Teilkomponenten zerlegen und dann deren Pose getrennt bestimmen [40], werden hier die Beziehungen zwischen den einzelnen Komponenten ausgenutzt. In den Experimenten kann die Poseschätzung dadurch schon mit nur einem Punkt in den Gliedern des Manipulators durchgeführt werden. Ohne Berücksichtigung der Abhängigkeiten wären dazu mindestens drei Punkte nötig.

Die Modellierung der kinematischen Ketten basiert im Wesentlichen auf der Arbeit von Bregler und Malik [4]. Sie benutzen ebenfalls die Exponentialdarstellung und führen zur Vereinfachung eine Taylorapproximation durch. In ihrer Arbeit beschränken sie sich jedoch auf die orthographische Projektion und legen sich auf den optischen Fluss zur Verfolgung der Bildmerkmale fest. Durch die Vereinfachung der projektiven Abbildung vernachlässigen sie die Grössenveränderungen innerhalb des beobachteten Objektes. Bewegungen, die zu grösseren Tiefenveränderungen innerhalb des Objektes führen, sind dadurch mit einem entsprechenden Fehler behaftet.

Der optische Fluss ermöglicht es ihnen die Poseschätzung ohne spezielle Farbmarkierungen durchzuführen. Verfahren, die auf den optischen Fluss basieren, haben aber den Nachteil, dass die Bewegungen zwischen den einzelnen Bildern nicht zu groß seien dürfen. Durch die diskrete Bildaufnahme wird sonst die Annahme eines kontinuierlichen Grauwertflusses verletzt. In der Diplomarbeit von S. Willemsen [41] wurde bereits ein Mehrauflösungsverfahren vorgestellt, mit dem auch schnelle Gelenkbewegungen mit dem Optischen Fluß verfolgt werden können. In ihrer Arbeit fehlt jedoch eine dreidimensionale Modellierung der kinematischen Abhängigkeiten und die Einbeziehung der projektiven Geometrie.

In dieser Arbeit wurde darauf verzichtet, ein bestimmtes Verfahren zur Extraktion der Bildinformationen in die Zwangsbedingungen zu integrieren. Bei der Entwicklung wurde vielmehr darauf geachtet, dass es sich mit möglichst vielen Methoden der Bildverarbeitung kombinieren lässt.

Viel Wert wurde auf die Einführung der mathematischen Grundlagen gelegt. Weitere Untersuchungen können dadurch auf die hier entwickelte Basis aufsetzen, ohne dass der Bezug zu den Problemen und der Vorgehensweise verloren geht.

Im Vergleich zu anderen Arbeiten (wie z.B. [4, 12, 25, 41]), in denen die Versuche nur anhand realer Bildsequenzen untersucht werden, wird das Verfahren in dieser Arbeit in künstlichen und realen Umgebungen untersucht. Dadurch können die ermittelten Fehler genauer analysiert und lokalisiert werden. Außerdem werden durch den Labormanipulator sehr genaue Posen vorgegeben, so dass der Vergleich von Soll- und Ist-Werten besonders genau ist.

Titel	Model Based Pose Estimation of Articulated and Constrained Objects	Mehrauflösungsverfahren zur Detektion und Verfolgung von schnellen Gelenkbewegungen in Sport–Bildfolgen	Hierarchical 3D Pose Estimation for Articulated Human Body Models from a Sequence of Volume Data	Video Motion Capture	Poseschätzung kinematischer Ketten
Autor	Y. Hel–Or, M. Werman	S.Willemsen	S. Weik, C. E. Liedtke	C. Bregler, J. Malik	O. Granert, B. Rosenhahn
Merkmals–extraktion	keine Angaben (Punktkorrespondenzen werden vorausgesetzt)	Optischer Fluß / Pyramidenverarbeitung	Farbsegmentierung	Optischer Fluß	verschiedene Verfahren Experimente: Farbsegmentierung / Hough–Transformation
Projektion	Stereoverfahren; keine genauen Angaben zur Modellierung	keine (Bewegung in der Bildebene)	perspektivische Projektion	orthographische Projektion	perspektivische Projektion
Modell	bisher nur Versuche mit planaren 3D–Modell	Gebiete in der Bildebene / 2D–Modell	3D – Modell	3D – Modell	3D – Modell
Lösungs–verfahren	Kalman–Filter	Kalman–Filter	hierarchisches Verfahren	Approximation / lineare Regression	Approximation / lineare Regression
Probleme	keine automatische Merkmalsextraktion; Versuche nur mit planaren 3D–Modell	Verdeckungen nur kurzzeitig möglich (2D–Modell); Optischer Fluß => Fehlerakkumulation bei längeren Bildsequenzen	explizite Formulierung der kinematischen Abhängigkeiten => mehrere Kameras nötig	Optischer Fluß => Fehlerakkumulation bei längeren Bildsequenzen; orthographische Projektion => Objekte mit großer Tiefe werden verzerrt	Farbsegmentierung und Hough–Transformation nur in bestimmten Umgebungen einsetzbar;

Abbildung 37: Vergleich verschiedener Verfahren zur Poseschätzung

8.2. Ausblick

Die im Rahmen dieser Arbeit durchgeführten Experimente demonstrieren nur einen Teil der Einsatzmöglichkeiten. In der Implementation ist bereits die Basis für Anwendungen geschaffen worden, in denen die verschiedenen Zwangsbedingungen gleichzeitig genutzt werden können. Ebenso können gleichzeitig mehrere kinematische Ketten verfolgt werden. Wie in Abschnitt 6 erwähnt wurde, ist diese Möglichkeit bei der Implementierung zwar berücksichtigt worden, ein entsprechendes Experiment wurde aber im Rahmen dieser Arbeit nicht mehr durchgeführt. Denkbar wäre hier zum Beispiel die Schätzung der Pose einer Hand oder der Gliedmaßen einer Person. Die Schwierigkeiten bei diesen Anwendungen sollten jetzt im Wesentlichen bei der Merkmalsextraktion und der Lösung des Korrespondenzproblemes liegen, da die Basis zur Berücksichtigung der kinematischen Beziehungen bereits in dieser Arbeit gelegt wurde.

Einige der hier erwähnten Anwendungsmöglichkeiten für das Poseschätzungsverfahren befinden sich bereits in der Entwicklung. Im Fortgeschrittenen Praktikum (Computer Vision II Praktikum) des Wintersemesters 2001 wird bereits an einem Projekt gearbeitet, bei dem der Laborroboter anhand der Pose eines menschlichen Armes geführt werden soll.

In der Diplomarbeit von Thomas Jäger [16] wird das Poseschätzungsverfahren bereits zur Generierung einer dreidimensionalen Objektrepräsentation eingesetzt. Diese Anwendung macht noch einmal die vielseitigen Einsatzmöglichkeiten der Implementation deutlich.

A. Zeichenerklärung

$\hat{\omega}$	schiefsymmetrische Matrix zur Beschreibung des Kreuzproduktes (Seite 10)
$\hat{\xi}$	Matrix zur Twist-Repräsentation (Seite 13)
$\overline{p}$	homogene Darstellung eines dreidimensionalen Vektors (Seite 12)
$GL(n)$	Menge der invertierbaren $n \times n$ Matrizen (Seite 5)
$O(n)$	Menge der orthogonalen $n \times n$ Matrizen (Seite 6)
$SO(n)$	die spezielle orthogonale Gruppe der $n \times n$ Matrizen (Seite 7)
$SE(n)$	die spezielle euklidische Gruppe der $n \times n$ Matrizen (Seite 11)
$so(n)$	Tangentialmenge der speziellen orthogonalen Gruppe SO(3) (Seite 11)
$se(n)$	Tangentialmenge der speziellen Euklidischen Gruppe SE(3) (Seite 13)
$\mathbb{H}$	Menge der Quaternionen (Seite 8)
$\mathbb{H}_1$	Menge der Einheitsquaternionen (Seite 9)
$\hat{\mathbb{H}}$	Menge der Dualquaternionen (Seite 14)

B. Merkmalsextraktion auf den Bilddaten

Dieser Abschnitt gibt eine kurze Übersicht über die in den Versuchen verwendeten Verfahren zur Merkmalsextraktion. Da die Merkmalsextraktion nicht Schwerpunkt dieser Arbeit ist, wird hier auf eine detailierte Darstellung der Verfahren verzichtet und nur ein grober Überblick über die wesentlichen Schritte der Merkmalsextraktion gegeben. In dieser Arbeit werden zwei unterschiedliche Verfahren verwendet:

1) Farbsegmentierung im HSI-Raum zur Punktdetektion

2) Hough-Transformation zur Geradendeterktion

B.1. Detektion der Farbmarkierungen

Ein Farbmodell ist die Spezifikation eines dreidimensionalen Farbkoordinatensystems. Ein Beispiel dafür ist das RGB (Red, Green, Blue) Farbmodell. Dabei werden die Farben in einem Würfel des dreidimensionalen kartesischen Koordinatensystems dargestellt. Das RGB-Farbmodell wird von vielen Hardwarekomponenten unterstützt und auch die verwendeten Laborkameras liefern die Bilder im RGB-Format.

Im Gegensatz zum technischen RGB-Farbmodell ist das HSI (Hue, Saturation, Intensity) - Modell ein dem Menschen angepasstes Farbmodell. Dieses Modell eignet sich besonders zur Extraktion der Farbmarkierungen, da in diesem Fall eine Koordinate (Hue) speziell für den Farbton vorhanden ist. Die Komponenten Sättigung und Intensität, die sehr von der Beleuchtung beeinflusst werden, können so bei einer Segmentierung entsprechend bewertet werden. Zur Durchführung des Versu-

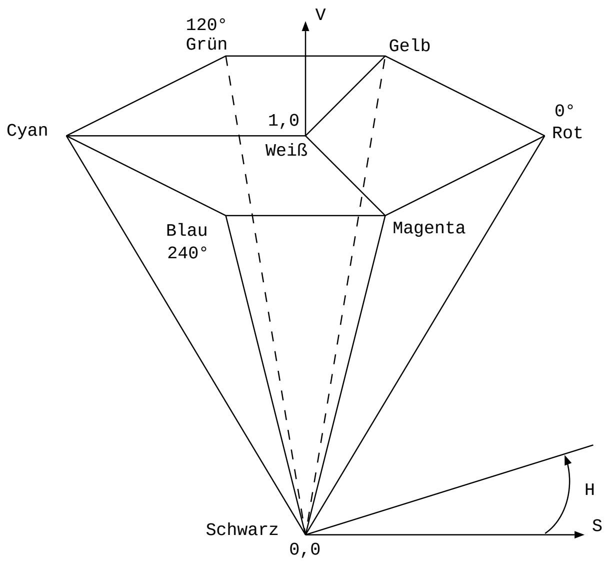


Abbildung 38: Das sechseckige HSV-Farbmodell

ches wurden daher zunächst die von der Videokamera gelieferten Bilder durch ein

Modul der KIVI-Klassenbibliothek in das HSI-Modell konvertiert und anschliessend segmentiert. Es stellte sich heraus, dass diese Module nicht für eine Echtzeitumgebung geeignet sind, da die internen Berechnungen mit Gleitkommaarithmetik durchgeführt wurden. Um die Geschwindigkeit zu erhöhen, wird nun nicht mehr von dem kreisförmigen HSI, sondern mit dem sechseckigen HSV-Modell (Hue, Saturation, Value) [8] gearbeitet (s. Abb. 38). Die Berechnung des Winkels (H-Komponente) lässt sich somit ohne trigonometrische Funktionen durchführen. Desweiteren wurde auf eine Skalierung der Farbwerte auf ein Intervall zwischen 0 und 1 verzichtet und die Winkel direkt im gewünschten Ausgabeformat (0 bis 255) berechnet. Dadurch ist die Berechnung ohne Gleitkommaarithmetik durchführbar. Der in B.1 ausgeführte Programmabschnitt zeigt die Umrechnung in vereinfachter Form.

Programm B.1 Schnelle RGB nach HSV Konvertierung

```
/////////////////////////////////////////////////////
// Convert from the RGB to the HSV color model.
// Input : r,g,b (values 0..255)
// Output: H,S,V (values 0..255)
////////////////////////////////////////////////////
void RGB2HSV_fast(unsigned char r,
                  unsigned char g,
                  unsigned char b,
                  int &H,int &S,int &V)
{
  int min;

  // V = max and Value
  V=MAX(r,g,b);       // calculate maximum from r,g,b
  min=MIN(r,g,b);     // calculate minimum from r,g,b

  // calculate Saturation
  S=(V!=0.0) ? ((int)(V-min)*255 / V) :0;

  // calculate Hue
  if(S == 0)
    H = 0;    // normaly UNDEFINED
  else
  {
    int delta = V - min;
    if( r == V)
      H=((g-b)*42)/delta;
    else
      if( g == V)
        H=85+((b-r)*42)/delta;
      else
        if( b == V)
          H=170+((r-g)*42)/delta;
  }
  if( H<0 )
    H+=256;
}
```

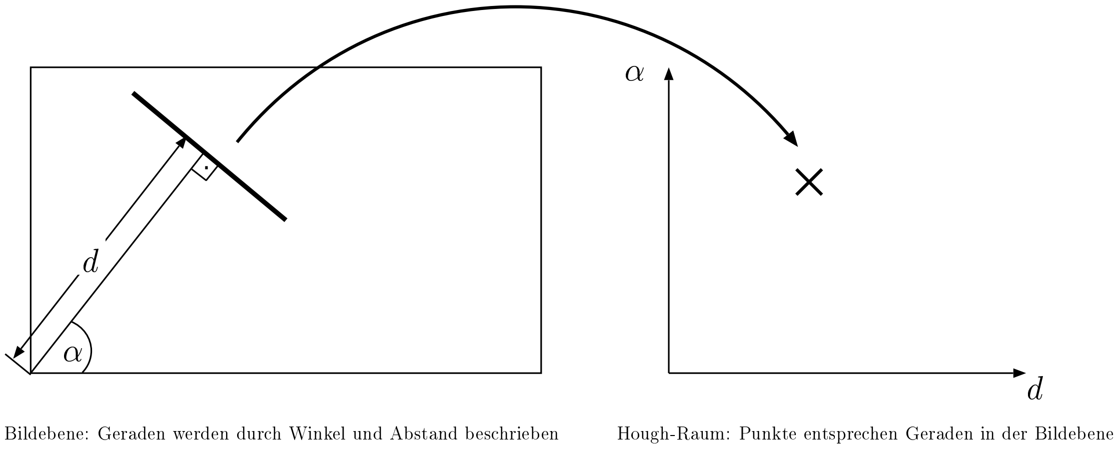


Abbildung 39: Darstellung der Hough-Transformation: Geraden werden durch die Berechnung von Winkel und Abstand in den Hough-Raum transformiert

B.2. Geradenextraktion mit Hough-Transformation

Die Hough-Transformation wurde 1962 von P.V.C. Hough [15] zum Patent angemeldet und ermöglicht es Strukturen aus Bildern zu extrahieren, die durch eine implizite Funktion beschreibbar sind. Insbesondere kann sie zur Extraktion von Geraden aus digitalen Bilddaten genutzt werden. Geraden können zum Beispiel durch einen Winkel α und den minimalen Abstand d der Geraden zum Ursprung beschrieben werden:

$$x \cos \alpha + y \sin \alpha - d = 0. \tag{2.102}$$

Diese Darstellung ist mit der Plücker-Koordinaten Darstellung von Geraden im dreidimensionalen Raum vergleichbar. Auch hier werden Geraden durch Punkte eines anderen abstrakteren Raumes repräsentiert.

Man verwendet diese Parametrisierung, um Sigularitäten zu vermeiden, wie sie z.B. bei der Parametrisierung einer Geraden durch Punkt und Steigung auftreten. Die Beschreibung einer senkrechten Linie würde hier nämlich zu einer *unendlichen* Steigung führen.

In dieser Arbeit wird eine Variante der Hough-Transformation benutzt, bei der nicht nur der Gradientenbetrag, sondern auch die Gradientenrichtung berücksichtigt wird. Man nennt dies auch gewichtete Hough-Transformation. Grundlage der gewichteteten Hough-Transformation ist das Gradientenbetragsbild und das Gradientenrichtungsbild. Das Gradientenbetragsbild wird dabei pixelweise abgearbeitet und bei der Überschreitung eines bestimmten Schwellwertes wird die Gerade bestimmt, die senkrecht zur Gradientenrichtung liegt und durch den aktuellen Pixel verläuft. Ein sogenanntes Akkumulatorfeld[16] hält dann die gefundenen Geradenpa-

[16] Der Akkumulator ist ein zweidimensionales Array mit ganzzahligen Werten. Er kann als Diskretisierung des Hough-Raumes angesehen werden, wobei die Zeilen und Spalten jeweils einem der beiden Geradenparameter (Winkel und Abstand) entsprechen. Jede Zelle des Akkumulators entspricht einer bestimmten Geraden in der Bildebene.

rameter fest, indem der Feldeintrag mit dem entsprechenden Winkel und Abstandswert um eine Einheit erhöht wird. Durch die Akkumulation der Geradeninformation über das gesammte Bild können lokale Störungen des Gradientenbildes ausgeglichen werden und eine stabilere Merkmalsextraktion durchgeführt werden.
Die großen Vorteile der Hough-Transformation bringen jedoch auch folgende Probleme mit sich:

1) Verlust von lokaler Information (Anfangs- und Endpunkt einer Linie gehen verloren)

2) Alias-Effekte können Gradientenvektor beeinflussen

3) optimale Maximumsbestimmung im Akkumulator (Hough-Raum) nicht eindeutig lösbar

4) Schwellwertfestlegung für den Gradientenbetrag schwierig

5) Schwellwert im Hough-Raum nicht a priori bekannt (d.h. es ist nicht bekannt wie viele Geraden sich im Bild befinden)

Einige dieser Probleme wurden in der Arbeit von Markus Ackermann genauer untersucht und in einigen Punkten auch deutliche Verbesserungen erzielt. Eine detailierte Beschreibung der benutzten Verfahren zur Geradenextraktion ist in seiner Diplomarbeit unter [1] zu finden.

C. Berechnung von Exponentialausdrücken

In diesem Abschnitt wird auf die Berechnungsmöglichkeiten von Exponentialausdrücken eingegangen. Die Definition der Exponentialfunktion basiert auf dem Grenzwert einer unendliche Reihe und eignet sich deshalb nicht zur Berechnung. In diesem Abschnitt wird deshalb eine Methode zur schnellen Berechnung der auftretenden Ausdrücke vorgestellt.

Bei der Beschreibung von Rotationen mit Hilfe der Exponentialfunktion treten im Argument der Exponentialfunktion meistens Produkte eines Skalars mit einer schiefsymmetrischen Einheitsmatrix auf. Gesucht ist also eine Möglichkeit, Ausdrücke der Form $\exp(\hat{\omega}\theta)$ mit $|\omega| = 1$ schnell und effektiv zu berechnen. Der als Rodrigues' Formel bekannte Ansatz löst dieses Problem durch Umwandlung der Exponentialfunktion in einen trigonometrischen Term:

$$\exp(\hat{\omega}\theta) = \mathbf{I} + \hat{\omega}\sin(\theta) + \hat{\omega}^2(1 - \cos(\theta)) \tag{3.103}$$

Im Folgenden wird gezeigt, wie man die Berechnung des Exponenten eines Twist, d.h. die Umwandlung von $\hat{\xi} \in se(3)$ in eine Transformationsmatrix aus $SE(3)$, auf die Rodrigues' Formel zurückführt. Geht man von dem Twist

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$$

aus, so berechnet sich die zugehörige Transformationsmatrix durch die beiden folgenden Fälle:

i. $\omega = 0$: In diesem Fall ist $\xi^2 = \xi^3 = \xi^4 = \cdots = 0$ und der Exponentialausdruck hat die einfache Form

$$\exp(\hat{\xi}\theta) = \mathbf{I} + \hat{\xi}\theta = \begin{bmatrix} \mathbf{I} & v\theta \\ 0 & 1 \end{bmatrix}. \tag{3.104}$$

ii. $\omega \neq 0$: Sei o.B.d.A. $||\omega|| = 1$[17]. Definiert man nun eine Transformation

$$g = \begin{bmatrix} \mathbf{I} & \omega \times v \\ 0 & 1 \end{bmatrix},$$

so lässt sich die Rotation um eine beliebige Achse im Raum in eine Rotation um den Ursprung mit anschließender Translation umrechnen. Man ersetzt dazu den Twist $\hat{\xi}$ wie folgt:

$$\begin{aligned} \hat{\xi}' &= g^{-1}\hat{\xi}g \\ &= \begin{bmatrix} \mathbf{I} & -\omega \times v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \omega \times v \\ 0 & 1 \end{bmatrix}, \\ &= \begin{bmatrix} \hat{\omega} & \omega\omega^T v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & \omega h \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{3.105}$$

[17] lässt sich durch Skalierung mit dem Faktor θ erreichen

wobei $h := \omega^T v$ ist.

Benutzt man nun die Gleichung

$$e^{\hat{\xi}\theta} = e^{g\hat{\xi}'\theta g^{-1}} = g e^{\hat{\xi}'\theta} g^{-1} \tag{3.106}$$

und die einfache Darstellung

$$e^{\hat{\xi}'\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & h\omega\theta \\ 0 & 1 \end{bmatrix},$$

so erhält man eine Beschreibung für $e^{\hat{\xi}\theta}$, die auf die Berechnung einer Drehung in $so(3)$ zurückgeführt worden ist:

$$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (\mathbf{I} - e^{\hat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}. \tag{3.107}$$

Literatur

[1] Ackermann, M. : *Objektrepräsentationen im Wahrnehmungs-Handlungs-Zyklus*, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Diplomarbeit, 2000

[2] Artin, M. : *Algebra.* Birkhäuser, 1993

[3] Blaschke, W. : *Mathematische Monographien 4, Kinematik und Quaternionen.* Deutscher Verlag der Wissenschaften, 1960

[4] Bregler, C. ; Malik, J. : Tracking people with twists and exponential maps. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , 1998, S. 8–15

[5] Ernst, J. : *Kalibrierung der internen Parameter einer aktiven Kamera*, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Diplomarbeit, 1996

[6] Faugeras, O. : *Three-Dimensional Computer Vision, A Geometric Viewpoint.* Cambridge : Mit Press, 1993

[7] Fischer, G. : *Lineare Algebra.* 10. Auflage. Vieweg Verlagsgesellschaft, 1995

[8] Foley, F. ; van Dam, A. ; Feiner, S. ; Hughes, J. : *Computer Graphics, Principle and Practice, 2nd ed. in C.* Addison-Wesley Publishing Company, 1996

[9] Grest, D. : *Ein System zur visuellen landmarkenbasierten Roboternavigation*, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Diplomarbeit, 2001

[10] v. Grudzinski, O. : *Lie-Gruppen I*, Mathematisches Seminar, Christian-Albrechts-Universität Kiel, Skriptum, 2000/2001

[11] Hansen, M. ; Diesner, G. ; Schmidt, H. ; Sommer, G. : Integration von Sensor und Aktor in einem binokularen Sehsystem. In: *In Tagungsband Fachtagung Sensoren-Aktoren-Buskommunikation* , 1995, S. 27–34

[12] Hel-Or, Y. ; Werman, M. : Model Based Pose Estimation of Articulated and Constrained Objects. In: *European Conf. on Computer Vision*, 1994, S. 262–273

[13] Hilgert, J. ; Neeb, K.-H. : *Lie-Gruppen und Lie-Algebren.* Vieweg Verlagsgesellschaft, 1991

[14] Horaud, R. ; Phong, T. ; Tao, P. : Object pose from 2d to 3d point and line correspondences. In: *International Journal of Computer Vision, Vol. 15* , 1995, S. 225–243

[15] Hough, P. : *Methods and means for recognizing complex patterns.* U.S. Patent 3, 069,654, 1962

[16] Jäger, T. : *Interaktion verschiedener visueller Modalitäten zur stabilen Extraktion von Objektrepräsentationen*, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Diplomarbeit, 2002

[17] KiViGraP: *Homepage of the Kieler Vision and Grasping Project.* http://www.ks.informatik.uni-kiel.de/∼ kivi/kivi.html

[18] Klette, R. ; Schlüns, K. ; Koschan, A. : *Three-Dimensional Data from Images.* Springer-Verlag Singapore, 1998

[19] Kloss, J. ; Rockwell, R. ; Szabo, K. ; Duchrow, M. : *VRML 97 - Der neue Standard für interaktive 3D-Welten im World Wide Web.* Addison-Wesley Publishing Company, 1997

[20] Levine, J. ; Mason, T. ; Brown, D. : *lex & yacc, 2nd Edition.* O'Reilly, 1992

[21] Li, H. : Generalized Homogeneous Coordinates for Computational Geometry. In: *Geometric Computing with Clifford Algebra.* , 2001, S. 27–52

[22] Manocha, D. : Solving Polynomial Systems Using Matrix Computations. In: Dikshit, H. P. (Hrsg.); Miccelli, C. A. (Hrsg.): *Advances in Computational Mathematics*, World Scientific Publishing Co., Inc., 1994, S. 99–129

[23] Murray, R. M.; Li, Z. ; Sastry, S. S.: *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994

[24] Real World Interface, Inc.: *Homepage der Firma iRobot.* http://www.irobot.com/rwi/

[25] Rosenhahn, B. ; Krüger, N. ; Rabsch, T. ; Sommer, G. : Tracking with a novel pose estimation algorithm. In: *International Workshop on Robot Vision; Auckland, New-Zealand, LNCS 1998* , 2001, S. 9–19

[26] Rosenhahn, B. ; Granert, O. ; Sommer, G. : Monocular pose estimation of kinematic chains. In: *Applied Geometric Algebras for Computer Science and Engineering* , 2001, S. 371–382

[27] Rosenhahn, B. ; Zhang, Y. ; Sommer, G. : Pose estimation in the language of kinematics. In: *Second International Workshop, Algebraic Frames for the Perception-Action Cycle, AFPAC 2000, LNCS 1888* , 2000, S. 284–293

[28] Schimmel, O. : *Dreidimensionale Szenenmodellierung durch monokulare Exploration mit einem mobilen Roboter*, Lehrstuhl Rechnerarchitektur, Universität Tübingen, Diplomarbeit, 1999

[29] SHEVLIN, F. : Analysis of orientation problems using Plücker lines. In: *International Conference on Pattern Recognition, Brisbane* , 1998, S. 685–689

[30] SONY: *Homepage der Firma SONY Deutschland.*
http://www.pro.sony-europe.com/isp/

[31] STÄUBLI S.A. : *Roboter RX 90 - CS7 Bedienungsanleitung.* Frankfurt Niedereschbach: Stäubli-Unimation, 1993

[32] STOER, J. : *Numerische Mathematik 1 und 2. Springer Lehrbuch.* Berlin : Springer Verlag, 1972-93

[33] T-ONLINE AG: *Homepage der T-Online International AG.*
http://www.t-online.de

[34] THE GCC TEAM: *GNU Compiler Collection Homepage.*
http://www.gnu.org/software/gcc/gcc.html

[35] THE VRML CONSORTIUM INC.: *ISO/IEC 14772-1:1997, Virtual Reality Modeling Language, (VRML97).*
http://www.vrml.org/Specifications/VRML97

[36] THOUIS, J. : *Introduction to Plücker Coordinates.*
http://www.flipcode.com/tutorials/tut_pluecker.shtml

[37] TSAI, R. Y.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: *IEEE Conference Computer Vision and Pattern Recognition, Miami Beach,FL* , 1986, S. 364–374

[38] UDE, A. : Filtering in a unit quaternion space for model-based object tracking. In: *Robotics and Autonomous Systems, Vol. 28, No.2-3* , 1999, S. 163–172

[39] WALKER, M. W.; SHAO, L. : Estimating 3-d location parameters using dual number quaternions. In: *Computer Vision, Graphics and Image Processing: Image Understanding, Vol. 54:3* , 1991, S. 358–367

[40] WEIK, S. ; LIEDTKE, C. E.: Hierarchical 3D Pose Estimation for Articulated Human Body Models from a Sequence of Volume Data. In: KLETTE, R. (Hrsg.); PELEG, S. (Hrsg.) ; SOMMER, G. (Hrsg.): *Robot Vision, LNCS 1998*, Springer-Verlag Berlin Heidelberg, 2001, S. 27–34

[41] WILLEMSEN, S. : *Mehrauflösungsverfahren zur Detektion und Verfolgung von schnellen Gelenkbewegungen in Sport-Bildfolgen*, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Diplomarbeit, 1997

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen der Arbeit beigetragen haben.

Als erstes möchte ich mich dabei bei *Herrn Prof. Dr. Gerald Sommer* für die interessante Themenstellung bedanken, ohne die diese Arbeit nicht hätte entstehen können.

Ganz besonderen Dank gilt meinem Betreuer *Bodo Rosenhahn*, der mich während der gesammten Zeit unterstützt hat und stets offen für Diskussionen war.

Mein Dank gilt auch *Dr. Norbert Krüger* mit dem ich viele fruchtbare Diskussionen führen konnte und der auch den Grundstein für das KiViGrap Projekt gelegt hat, so dass die hier entwickelten Verfahren auch in anderen Arbeiten genutzt werden können.

Bei *Daniel Grest* und *Kord Ehmcke* möchte ich mich für die sehr kooperative Zusammenarbeit bedanken. Sie haben mich auf viele Probleme aufmerksam gemacht und mir bei der Integration der Verfahren in die KiViGrap Umgebung immer hilfreich zur Seite gestanden.

Bedanken möchte ich mich auch bei meinem Vater *Rolf Granert* und *Melanie Wons* die die Arbeit Korrekturgelesen haben.

Hiermit versichere ich, dass ich die Arbeit selbstständig verfaßt und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.

Kiel, den 5. März 2002