

Bildbasierte Bewegungsschätzung eines unbemannten Helikopters

Diplomarbeit zur Erlangung des Grades eines

**Diplom-Informatikers
im Studiengang Computervisualistik**

vorgelegt von

Christian Isleib

Betreuer: Dipl.-Inf. Franz Andert, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Flugsystemtechnik, Abt. Unbemannte Luftfahrzeuge

Erstgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik

Zweitgutachter: Dipl.-Inf. Johannes Pellenz, Institut für Computervisualistik, Fachbereich Informatik

Koblenz, im Oktober 2008

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Arbeitsgruppe für Studien- und Diplomarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja nein

Koblenz, den 31. Oktober 2008

Unterschrift

Zusammenfassung

Das Wissen über die genaue Position und Lage eines unbemannten Luftfahrzeugs spielt während der Durchführung einer autonomen Mission eine dominante Rolle. Unbemannte Luftfahrzeuge sind daher mit einer Vielzahl an Sensoren ausgestattet. Jeder der Sensoren leistet einen Beitrag zu diesem Ziel, wobei ein Sensor entweder eine *absolute* oder eine *relative* Angabe über den derzeitigen Aufenthaltsort oder die Fluglage ermöglicht. Alle Sensoren werden zu einer Gesamtlösung verknüpft, die *Navigationslösung* genannt wird. Das am häufigsten eingesetzte – und auch meistens einzige – Verfahren zur absoluten Positionsbestimmung ist die Satellitennavigation. Diese ist abhängig von einer direkten Sichtlinie der mitgeführten Empfangsantenne zu den Satelliten. Falls es zu einer Unterbrechung dieser Sichtlinie kommt, ist eine genaue, absolute Positionsangabe nicht mehr möglich. Die Navigationslösung hat somit nur noch Sensoren zur Verfügung, die eine relative Positions- beziehungsweise Lageangabe ermöglichen. Hierzu gehören das mitgeführte Magnetometer und das Inertialmesssystem. Beide unterliegen dem Phänomen der Drift. Dieses bedeutet, dass die Genauigkeit der Positions- und Lageangabe bei einem Ausfall der Satellitennavigation mit fortschreitender Zeit zunehmend unzuverlässiger wird. Um diese Drift in einem bestimmten Rahmen zu kompensieren, kann ein Bildsensor verwendet werden. Dieser ermöglicht eine *bildbasierte Bewegungsschätzung* und stellt somit einen zusätzlichen Sensor zur Messung von *relativen* Lage- und Positionsänderungen dar. Ziel dieser Arbeit ist es, ein Verfahren zur bildbasierten Bewegungsschätzung für einen unbemannten Helikopter zu entwickeln und zu evaluieren.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Unbemannte Luftfahrzeuge	7
1.2	Die ARTIS-Plattform	8
1.3	Motivation	9
1.4	Kapitel-Übersicht	10
2	Grundlagen	13
2.1	Sensorik des Versuchsträgers	13
2.2	Stereo-Vision	15
2.3	Merkmalsdetektion und Merkmalsverfolgung	21
2.4	Stand der Wissenschaft und eigener Ansatz	23
3	Bildbasierte Bewegungsschätzung	25
3.1	Merkmalsausbeute	25
3.2	Filterung der Merkmale	28
3.2.1	Mahalanobis-Distanz	28
3.2.2	Smooth Local 3D	29
3.2.3	RANSAC	32

3.3	Schätzung der Eigenbewegung	34
3.4	Berechnung der Trajektorie	38
4	Implementierung	41
4.1	Das DIP-Framework	41
4.2	Beschreibung der Implementierung	43
4.3	Beschreibung der grafischen Schnittstelle	45
5	Experimente und Ergebnisse	49
5.1	Erstellung von synthetischen Testdaten	49
5.1.1	Die Simulationsumgebung	51
5.1.2	Simulation der Stereokamera	51
5.1.3	Erstellung der Testdaten	54
5.2	Versuchsansätze	57
5.2.1	Synthetische Kamerafahrten	57
5.2.2	Reale Kamerafahrten	58
5.3	Durchführung der Schätzung der Eigenbewegung	59
5.4	Ergebnisse der synthetischen Kamerafahrten	62
5.4.1	Quadrat	63
5.4.2	Schleife	64
5.4.3	Hummelflug	66
5.4.4	Dynamisch	67
5.5	Ergebnisse der realen Kamerafahrten	68
6	Evaluation der Ergebnisse	71
6.1	Evaluation der synthetischen Kamerafahrten	71

6.2	Evaluation der realen Kamerafahrten	72
6.3	Gesamtevaluation der Verfahren	75
6.4	Referenzmessung der Tiefenwerte der Stereokamera	77
7	Zusammenfassung und Ausblick	81
8	Anhang	85
8.1	Spezifikation der Stereokamera	87
8.2	Auswertungsdaten	88
8.2.1	Referenzmessung der Tiefenwerte der Stereokamera	88
8.2.2	Tiefenwert-Testtabelle der Simulation	90
	Literaturverzeichnis	91

Kapitel 1

Einleitung

Diese Diplomarbeit wurde am Institut für Flugsystemtechnik in der Abteilung für Unbemannte Luftfahrzeuge (UL¹) des Deutschen Zentrums für Luft und Raumfahrt (DLR) in Braunschweig verfasst.

1.1 Unbemannte Luftfahrzeuge

Unbemannte Luftfahrzeuge, oft auch *Drohnen* genannt, spielen eine immer größer werdende Rolle in sowohl militärischer als auch ziviler Hinsicht. Dadurch, dass UL keinen Piloten an Bord mitführen, können Missionen durchgeführt werden, die vorher nicht oder nur schwer möglich gewesen wären. Die gewonnenen Vorteile sind zum einen, dass der Pilot nicht mehr durch einen riskanten Einsatz gefährdet wird und dass zum anderen die Einsatzdauer signifikant erhöht werden kann, da ein UL keine *Ermüdung* kennt – allein die Technik gibt die limitierenden Faktoren vor. UL sind dabei nicht auf ein Flugmuster beschränkt. Neben normalen Flächenflugzeugen werden auch alle anderen Arten von Luftfahrzeugen wie etwa Drehflügler, Luftschiffe etc. je nach Verwendungszweck eingesetzt.

Unbemannt bedeutet jedoch nicht, dass diese Luftfahrzeuge auch unbedingt *autonom* agieren – der Grad der Automatisierung kann sehr unterschiedlich ausfallen. Es wird hierbei

¹Das englische Pendant zu UL ist UAV, *Unmanned Aerial Vehicle*.

zwischen den halb-autonomen und den voll-autonomen UL unterschieden. Halb-autonom bedeutet, dass ein UL zum Großteil der Flugphase durch einen Piloten über eine Bodenstation gesteuert werden muss. Komplexere Systeme, wie etwa der *Global Hawk*, erlauben einen höheren Grad der Automatisierung, so dass Flugmuster diesen Typs voll-autonom starten, eine Mission fliegen und wieder landen können.

Der Preis für die Autonomie eines UL liegt im Einsatz einer Vielzahl an Sensorik, die den sicheren Flug und die präzise Navigation ermöglicht. Zu den am häufigsten genutzten Sensoren zählen dabei die Satellitennavigation², Radionavigation, Trägheitsnavigation³, sowie das Magnetometer, Radar⁴, LIDAR⁵ und auch sichtbasierte Sensoren.

1.2 Die ARTIS-Plattform

Das Institut für Flugsystemtechnik des DLR in Braunschweig hat in 2003 mit der Entwicklung eines Versuchsträgers zur Erforschung autonomierelevanter Themen im Bereich der UL begonnen [DBT03]. Wichtig ist, dass dieses Projekt nicht das Ziel hat, ein *spezielles* UL mit einem *bestimmten* Einsatzzweck zu entwickeln, sondern im Allgemeinen intelligente Systeme und Funktionen für UL evaluieren soll. Der Versuchsträger wird ARTIS (**A**utonomous **R**otorcraft **T**estbed for **I**ntelligent **S**ystems) genannt und besteht heute aus einer Familie von drei Modell-Helikoptern: miniARTIS, midiARTIS und maxiARTIS.

Flugversuche werden am häufigsten mit den beiden größeren Modellen geflogen, da die Nutzlast eine wichtige Rolle für die Einsätze spielt. Je größer die mögliche Nutzlast ist, desto mehr zusätzliche Avionik⁶ kann pro Flugversuch eingesetzt werden. Der maxiARTIS (siehe Abbildung 1.1) ist in 2006 zum ersten Mal eingesetzt worden und basiert im Gegensatz zum mini- und midiARTIS nicht mehr auf einem handelsüblichen Modellbausatz. Er ist eine Spezialanfertigung, die über genügend Leistung für die Zuladung experimentel-

²englisch: GPS, **G**lobal **P**ositioning **S**ystem.

³englisch: INS, **I**nternal **N**avigation **S**ystem.

⁴englisch: **R**adio **D**etection and **R**anging.

⁵englisch: **L**ight **D**etection and **R**anging.

⁶Avionik: *die*, zu **A**viatik und **E**lektronik, die Gesamtheit elektronischer Luftfahrtgeräte sowie die Wissenschaft und Technik dieser Geräte und Systeme. Meyers Lexikon.



Abbildung 1.1: Der Versuchsträger maxiARTIS während eines Flugversuches.

ler Nutzlast verfügt. Das maximale Abfluggewicht⁷ liegt derzeit (aufgrund von Beschränkungen der Luftverkehr-Zulassungs-Ordnung) bei 25 kg. Tabelle 1.1 fasst die wichtigsten Eckdaten des Helikopters zusammen.

Rotordurchmesser	3 m
Antrieb	Wellenleistungsturbine, 4,5 kW
Leergewicht	15 kg
Gewicht Avionik	4 kg
Zuladung	max. 6 kg
MTOW	25 kg

Tabelle 1.1: Technische Spezifikationen des maxiARTIS-Versuchsträgers.

1.3 Motivation

Ziel dieser Arbeit ist die bildbasierte Bewegungsschätzung⁸ eines unbemannten Helikopters zur Unterstützung der Navigation. Damit ein UL zuverlässig autonom betrieben werden kann, ist unter anderem das Wissen über dessen genaue Position und Fluglage wichtig.

⁷MTOW, englisch: **Maximum Takeoff Weight**.

⁸Häufig wird auch der Begriff der *visuellen Odometrie* verwendet.

Hierzu sind die ARTIS-Versuchsträger mit Sensoren ausgestattet, die jeweils einen unterschiedlichen Beitrag zur Positions- und Lagebestimmung leisten. Hierbei ist es wichtig, zwischen Sensoren zu unterscheiden, die eine *relative* beziehungsweise eine *absolute* Positionsbestimmung ermöglichen. Zur absoluten Positionsbestimmung sind die Versuchsträger mit einem GPS ausgestattet, welches das Wissen über die eigene, absolute Position liefern kann. Im Gegensatz dazu liefert der IMU⁹-Sensor durch Integration der Beschleunigungswerte und Drehraten eine nur relative Positionsinformation. Die relative Positionsbestimmung unterliegt der *Drift*. Daraus folgt, dass der Sensor mit zunehmender Flugdauer immer ungenauere Positionsangaben liefern wird, es sei denn, er wird durch ein GPS gestützt.

Für den Fall, dass die Positionsbestimmung via GPS versagt – dieses kann zum Beispiel immer dann auftreten, wenn nicht ausreichend Satelliten zur Positionsbestimmung vorhanden sind oder gar keine direkte Sicht zu den Satelliten möglich ist – so ist nur noch eine relative Positionsbestimmung für den Zeitraum des Ausfalls möglich. Je länger dieser Ausfall dauert, desto unsicherer wird das Wissen über die eigene Position.

Neben der IMU kann auch ein Bildsensor die relativen Positions- und Lageänderungen messen, indem die Bilddaten mittels Methoden der Bildverarbeitung und des Rechnersehens ausgewertet werden. Die Versuchsträger verfügen über ein Stereokamerasystem, welches die benötigten Bilddaten liefern kann.

Die Herausforderung besteht darin, diese Bilddaten so auszuwerten, dass man die Eigenbewegung des Versuchsträgers rein aus den Bilddaten schätzen kann. So hätte man für zukünftige Szenarien zwei Sensoren, die ähnliche Informationen liefern und zusammen eine bessere relative Positions- und Lagebestimmung schätzen können, um einen GPS-Ausfall zu kompensieren.

1.4 Kapitel-Übersicht

Diese Arbeit gliedert sich wie folgt: In **Kapitel 2** werden zum einen die Grundlagen wie etwa eine erweiterte Einführung in die Sensorik des Versuchsträgers, der Stereo-Vision und

⁹englisch: Inertial Measurement Unit. Wird zur Trägheitsnavigation verwendet.

der Merkmalsverfolgung besprochen. Zum anderen wird der Stand der Wissenschaft dargelegt. **Kapitel 3** behandelt das Konzept des in dieser Arbeit vorgestellten Verfahrens zur bildbasierten Bewegungsschätzung und liefert die nötigen mathematischen Hintergründe. Auf die Implementierung des Verfahrens in die Zielumgebung wird in **Kapitel 4** eingegangen. In **Kapitel 5** werden die Experimente und Ergebnisse präsentiert, gefolgt von der Evaluation in **Kapitel 6**. Zuletzt wird in **Kapitel 7** eine Zusammenfassung dieser Arbeit gegeben, abgeschlossen mit einem Ausblick in die Zukunft.

Kapitel 2

Grundlagen

Das folgende Kapitel erläutert anfangs die Sensorik des eingesetzten Versuchsträgers und den daraus folgenden Konsequenzen für die Navigation. Danach wird genauer auf das Stereosystem des Versuchsträgers eingegangen, da dieser Sensor den Ausgangspunkt für die bildbasierte Bewegungsschätzung darstellt. Anschließend wird auf die Grundlagen der Merkmalsfindung und der Merkmalsverfolgung eingegangen. Der letzte Abschnitt behandelt den Stand der Technik und leitet zum eigenen Ansatz zur bildbasierten Bewegungsschätzung über.

2.1 Sensorik des Versuchsträgers

Der Versuchsträger ist mit einer Vielzahl an Sensoren ausgestattet, die nach zwei Hauptmerkmalen klassifiziert werden können:

- aktiv / passiv
- relativ / absolut

Das Hauptmerkmal *aktiv/passiv* ist hauptsächlich in der militärischen Anwendung interessant. Eine passive Sensorik erhöht die Schwierigkeit der Ortung des UL durch die Gegenpartei, da keine elektromagnetischen Emissionen durch die Sensorik ausgestrahlt wird.

Eine Zielmarkierung oder Entfernungsmessung via Laser wäre leichter zu detektieren als ein rein auf Bilddaten basierende Aufklärung. Da der Schwerpunkt dieser Arbeit in der Navigation liegt, wird nicht weiter auf diesen Punkt eingegangen. Es ist lediglich zu erwähnen, dass der einzige aktive Sensor des ARTIS-Versuchsträgers das ausschließlich zur Landung eingesetzte Sonar ist.

In der Navigation ist die Klassifizierung eines Sensors in die relative beziehungsweise absolute Positionsbestimmung wichtig. Absolute Positionsbestimmung bedeutet, dass zu jedem Zeitpunkt t der Sensor seine absolute Position bestimmen kann, ohne auf vorhergehende Daten zurückgreifen zu müssen. Der Fehler in der absoluten Positionsbestimmung bleibt hauptsächlich in einem gewissen, durch das Verfahren beeinflussten Rahmen. Neben der Funkortung ist die Bestimmung der absoluten Position durch ein GPS das am häufigsten eingesetzte Verfahren in der Selbstlokalisierung von Flugzeugen.

Die relative Positionsbestimmung kann unter dem Begriff der Koppelnavigation (*engl. dead reckoning*) geführt werden. In der Luftfahrt wird bei Anwendung der Sichtflugregeln häufig dieses Verfahren eingesetzt, um die eigene Position auf einer Karte zu markieren und zu verfolgen. Durch den vom Kompass abgelesenen Kurs und die derzeitige Geschwindigkeit kann man seine aktuelle und zukünftige Position *schätzen*, aber eben *nicht exakt* angeben. Eine exakte Angabe ist immer nur dann möglich, wenn man seine Position zu einem bekannten Objekt referenzieren kann. Dieses bedeutet zwangsläufig: Je länger man ohne einen Bezugspunkt, also *blind* fliegt, desto größer wird die Unsicherheit über die eigene Position. Ein relativer Fehler wird sich demnach immer akkumulieren. Wenn eine bekannte Position erreicht wird, kann dieser Fehler jedoch wieder zurückgesetzt werden und die Akkumulation der relativen Fehler beginnt erneut. Abbildung 2.1 zeigt die im maxiARTIS verwendete Avionik.

Der Versuchsträger führt momentan zur absoluten Lokalisierung nur ein GPS-System mit sich. Alle anderen Sensoren, wie etwa das Inertialmesssystem, das Magnetometer oder die Stereokamera, können nur zur relativen Positionsbestimmung genutzt werden. Da ein GPS zum Beispiel eine gute laterale beziehungsweise longitudinale Ortsbestimmung ermöglicht, aber in der Höhenangabe ungenauere Werte liefert, kombiniert man alle Sensoren in einen Navigationsfilter. Diese Kombination der Sensorik läuft unter dem Begriff der Sensor-Fusion [WL90] und hat zum Ziel, das Gesamtergebnis der Positions- und La-

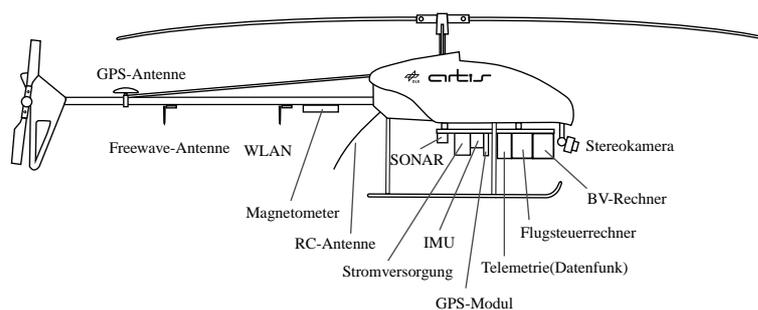


Abbildung 2.1: Seitenansicht maxiARTIS mit Avionik.

geangabe zu verbessern. Jeder Sensor hat seine eigenen Stärken und Schwächen, einige Sensoren wirken sogar komplementär und können sich somit gut ergänzen. Die Modellierung solcher Abhängigkeiten nennt man Systemidentifikation. Gerade Intertialmesssysteme und sichtbasierte Sensoren können sich sinnvoll ergänzen (vgl. [CLD07]), warum auch viele einen solchen kombinierten Ansatz für die visuelle Odometrie wählen. Für die optimale Verrechnung der Eingangsdaten des Navigationsfilters sorgt ein Kalman Filter [TBF05]. Abbildung 2.2¹ zeigt exemplarisch die Auswirkung der Filterung an den Navigationsdaten eines Flugversuches, besonders gut erkennbar im Intervall $t[30; 35]$.

2.2 Stereo-Vision

Kameras können als Bild- und Entfernungssensoren verwendet werden. Als einfacher Bildsensor liefert eine Kamera je nach Typ Helligkeits- und/oder Farbwerte der aufgenommenen Umwelt. Als Entfernungssensor können Kameras dazu benutzt werden, dreidimensionale Strukturen aus Bildern zu rekonstruieren. Diese Rekonstruktion der Tiefe einer aufgenommenen Umwelt ist schon mit einer einzelnen Kamera möglich. Zwei Fotos aus einer leicht unterschiedlichen Perspektive reichen aus, um die Entfernungen zu den Objekten zu berechnen (siehe [HZ04]). Dieses bedeutet, dass durch die Änderung der Perspektive Zeit zwischen den einzelnen Aufnahmen vergangen ist. Mit einer Stereokamera hat man die Möglichkeit, zeitgleich zwei Fotos aus einer leicht unterschiedlichen Perspektive

¹Die dort gezeigten Flughöhen sind nach der deutschen Luftfahrtnorm [Nor70] negativ, gemessen vom Schwerpunkt des Flugobjektes.

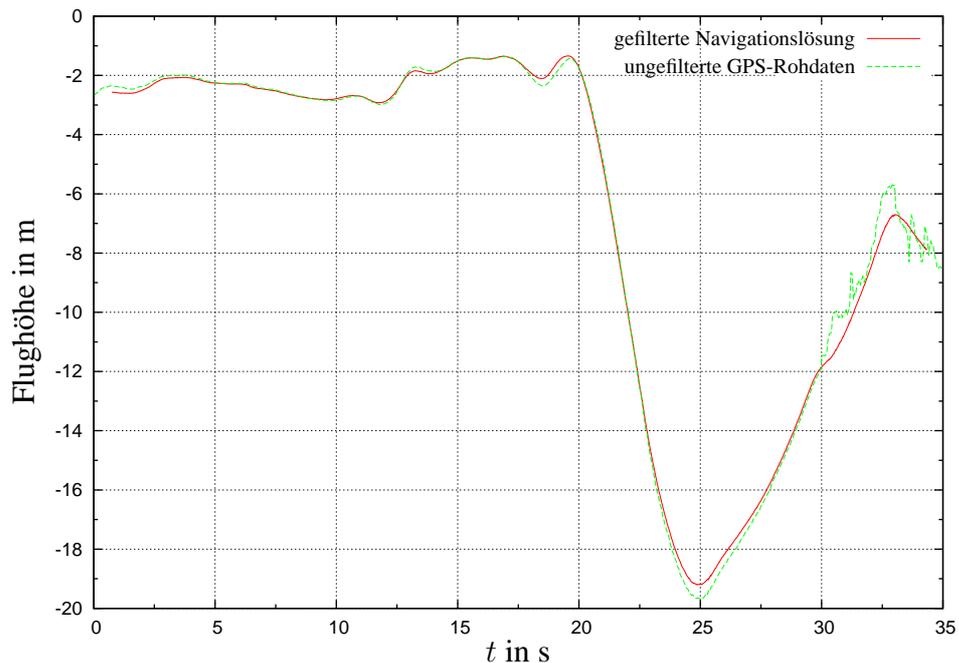


Abbildung 2.2: Abweichung zwischen GPS und gefilterter Navigationslösung.

aufzunehmen. Die aufgenommenen Bilder können mit den Gesetzen der Epipolargeometrie [HZ04] zu sogenannten *Tiefenbildern* verarbeitet werden. Hierzu sind Berechnungen notwendig, die normalerweise durch einen an die Stereokamera angeschlossenen Rechner durchgeführt werden.

Die in dieser Arbeit verwendete Stereokamera benötigt jedoch *keinen* zusätzlichen Rechner zur Berechnung der Tiefenbilder. Sie ist eine sogenannte STOC²-Stereokamera von Videre Design [Vid07] und führt die benötigten Berechnungen auf einem in die Stereokamera integrierten FPGA³-Chip in Echtzeit aus. Dieser Vorteil hilft, wertvolle Rechenzeit auf dem an die Stereokamera angeschlossenen Bildverarbeitungsrechner zu sparen. Abbildung 2.3 zeigt die am maxiARTIS montierte Stereokamera.

Als Ausgabe liefert die Stereokamera – mit einer maximalen Rate von 30 Hz bei einer Auflösung von $640p_x \times 480p_y$ Pixeln – jeweils ein Graustufenbild und ein Tiefenbild (sie-

²englisch: **Stereo on Chip**.

³englisch: **Field Programmable Gate Array**.



Abbildung 2.3: Stereokamera des maxiARTIS-Versuchsträgers.

he Abbildung 2.4). Die Graustufenbilder enthalten die Helligkeitsinformationen der aufgenommenen Umwelt und werden deshalb auch *Intensitätsbilder* genannt. Die Tiefenbilder werden auch als *Tiefenkarten* bezeichnet und enthalten die Entfernungen zu den Objekten der aufgenommenen Umwelt. Die Entfernungsinformationen werden in Helligkeitswerte umgewandelt. Nahe Objekte werden in einer Tiefenkarte *heller*, weiter entfernte Objekte *dunkler* dargestellt.

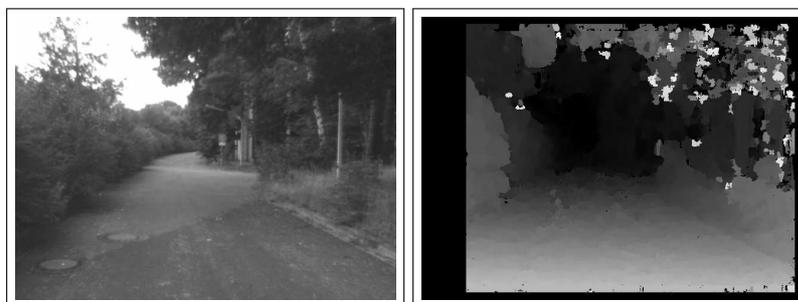


Abbildung 2.4: Beispiel für ein Intensitäts- und ein Tiefenbild. Nahe Objekte werden *heller*, weiter entfernte Objekte *dunkler* dargestellt.

Wie schon Eingangs erwähnt, werden die Tiefenwerte über die Epipolargeometrie berechnet. Da die verwendete Stereokamera zwei parallel ausgerichtete Einzelkameras besitzt,

spricht man vom *idealen* Stereomodell (vgl. [SK08], Seite 523). Abbildung 2.5 verdeutlicht die Eigenschaften dieses Modells.

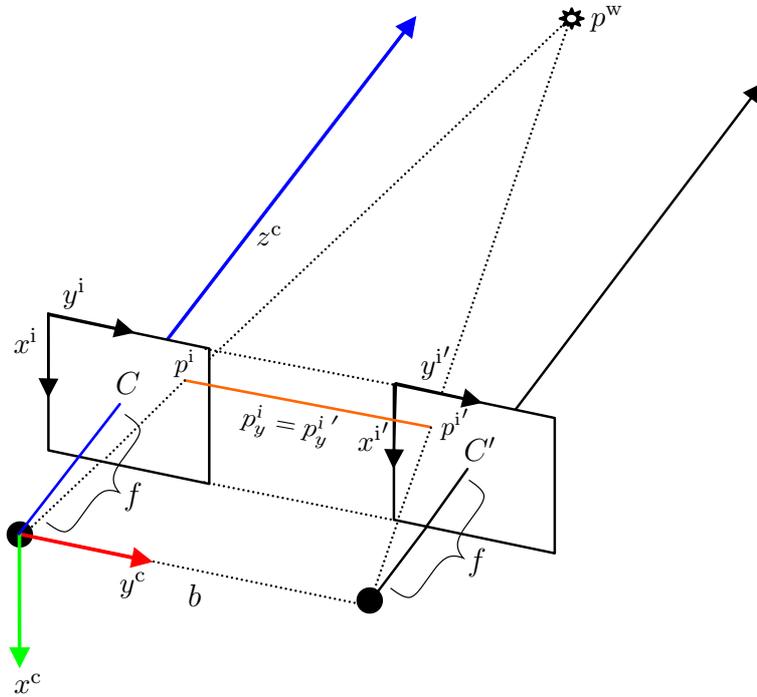


Abbildung 2.5: Ideal-Stereo: Epipolargeometrie mit parallelen Kameras.

In dieser Abbildung sieht man die zwei parallel angeordneten Kameras, die einen Weltpunkt $p^w = (p_x^w, p_y^w, p_z^w)^T$ sehen können. Da die Kameras um die Basislinie b voneinander entfernt sind, ergibt sich durch die Projektion ein Unterschied im Darstellungsort in der Bildebene der jeweiligen Kamera. In der linken Kamera wird p^w auf den Punkt p^i projiziert, in der rechten Kamera auf Punkt $p^{i'}$. Die y -Koordinate ist in beiden Fällen gleich, nur in der x -Koordinate gibt es einen Unterschied. Dieser Unterschied wird *Disparität* d genannt und in das Tiefenbild als Grauwert kodiert. Die Rückrechnung der ursprünglichen Raumkoordinate p^w aus dem Tiefenbild erhält man mit

$$p_z^w = \frac{b \cdot f}{d}, \quad p_x^w = p_z^w \cdot \frac{(p_x^i - C_x)}{f}, \quad p_y^w = p_z^w \cdot \frac{(p_y^i - C_y)}{f}, \quad (2.1)$$

wobei C_x, C_y der Hauptpunkt des Tiefenbildes ist.

Mit der Kalibrierung der Stereokamera wird sichergestellt, dass alle notwendigen Eigenschaften für das ideale Stereomodell erfüllt werden. Hierzu zählt zum Beispiel die Rektifizierung, die die Korrespondenzsuche insofern vereinfacht, als dass Korrespondenzen immer die gleiche y -Komponente aufweisen. Die eingesetzte Stereokamera hat eine Brennweite f von 4,1 mm. Der horizontale Öffnungswinkel beläuft sich auf ca. 49° , der vertikale beträgt ca. 38° . Die Basislinie b zwischen beiden Kameras liegt bei ca. 30 cm.⁴

Aufgrund der mit der Entfernung abnehmenden Disparität und der daraus resultierenden Ungenauigkeit in der Tiefenwertberechnung ergeben sich Einschränkungen in der Verwertbarkeit der Entfernungsmessungen. Die Berechnung der Entfernungsauflösung ΔZ erfolgt durch

$$\Delta Z = \frac{Z^2}{b \cdot f} \cdot \Delta d, \quad (2.2)$$

wobei Z der Tiefe entspricht, b der Basislinie, f der Brennweite der Linsen und Δd der kleinsten noch messbaren Pixelgenauigkeit.⁵ Wie man sehen kann, spielt die Basislinie eine große Rolle in der Tiefenaufklärung. Je größer b ist, desto flacher wird der Verlauf der Kurve und somit verbessert sich die Tiefenmessbarkeit von weit entfernten Objekten. Ein weiterer, einflussreicher Faktor wird durch die Pixelgenauigkeit bestimmt. Der Hersteller gibt die auch Subpixelgenauigkeit genannte Größe mit $\frac{1}{16}$ an, wobei dieser Wert äußerst optimistisch gewählt ist. Nach den Erkenntnissen nach Williamson [Wil98] wäre eine Genauigkeit von $\frac{1}{4}$ angemessen, wobei Szeliski [SS02] sogar nur von $\frac{1}{2}$ ausgeht. Abbildung 2.6 zeigt die Entfernungsauflösung für die Subpixeldisparitäten von $\frac{1}{16}$ und $\frac{1}{4}$.

Bedingt durch die Basislinie und der maximalen Disparität des Korrespondenzalgorithmus' von 64 Pixeln, liegt der durch die Stereokamera theoretisch messbare Entfernungsbereich bei $[3,28 \text{ m}, \infty]$. Berechnet wird die untere Schranke mit Gleichung 2.3:

$$D_{\min \text{ Stereo}} = \frac{b \cdot f}{d_{\max}} \quad (2.3)$$

Realistischerweise liegt die Obergrenze für die Entfernungsmessung aufgrund der qua-

⁴Eine komplette Spezifikation der Stereokamera ist im Anhang unter 8.1 zu finden.

⁵Hinweis: p_z^w wurde durch Z substituiert.

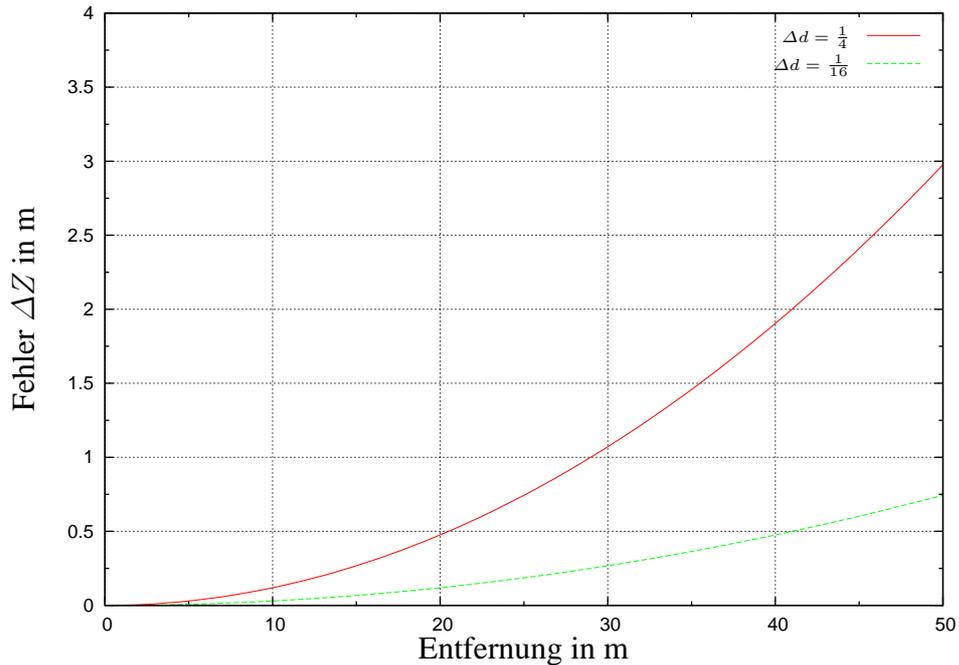


Abbildung 2.6: Tiefenauflösung der Stereokamera bei $\Delta d = \frac{1}{4}$ und $\Delta d = \frac{1}{16}$.

dratisch zunehmenden Tiefenungenauigkeit weit niedriger. Das Maximum könnte bei 50 m angesetzt werden, da hier im schlimmsten Fall eine Tiefenabweichung von 3 m erreicht wird. Eventuell sollte die Schranke noch tiefer angesetzt werden. Hierdurch ergibt sich, dass der Einsatzbereich des Versuchsträgers durch den Messbereich der Stereokamera eingeschränkt wird. Dieses wirkt sich zum Beispiel direkt auf die realisierbare Flughöhe aus. Der Helikopter muss folglich in einer Höhe operieren, die den Messbereich weder unter- noch überschreitet. Die beste Lösung dieses Problems bestünde darin, immer an der Untergrenze zum Messbereich zu fliegen, da hier die Tiefenauflösung der Stereokamera die zuverlässigsten Messwerte liefern kann. Sollte sich der Helikopter außerhalb dieses Raumes befinden, so würde das im Umkehrschluss zu einem Versagen dieses Verfahrens zur bildbasierten Bewegungsschätzung führen. Als Kompensation müsste dann ein anderes Verfahren eingesetzt werden. Vorstellbar wäre ein Mono-Kamera-Ansatz über den optischen Fluss. Das Ergebnis wäre ein hybrides Verfahren, welches beide Ansätze vereint.

2.3 Merkmalsdetektion und Merkmalsverfolgung

Der in dieser Arbeit vorgestellte Ansatz zur bildbasierten Bewegungsschätzung setzt zur Gewinnung von Punktmengen und deren Korrespondenzen den Einsatz einer Merkmalsverfolgung voraus. Um ein Merkmal in einer Bildsequenz verfolgen zu können, bedarf es generell zweier Schritte. Der erste Schritt besteht darin, geeignete Merkmale im Bild zu ermitteln. Sobald diese gefunden sind, besteht der zweite Schritt darin, diese Merkmale über den Verlauf einer Bildsequenz wiederzufinden und zu verfolgen. Abbildung 2.7 zeigt hierzu ein Beispiel. Für beide Schritte gibt es verschiedenste Verfahren, die eingesetzt werden können. Ein wichtiger Aspekt für den aktiven Einsatz in einem UL ist die Laufzeit, damit *während* des Fluges die Bildsequenzen ausgewertet werden können und den darauf aufbauenden Algorithmen auch genügend Rechenleistung für die Echtzeitverarbeitung zur Verfügung steht.



Abbildung 2.7: Beispielsequenz zur Demonstration der Merkmalsverfolgung. Zwei Merkmale wurden zur Verdeutlichung durch eine Umrandung exemplarisch hervorgehoben.

Ein sehr häufiges, in echtzeitrelevanten Kontexten eingesetztes Verfahren zur Merkmalsdetektion und -verfolgung basiert auf den Arbeiten von Tomasi et al. ([ST94] und [TK91]). Die Merkmalsdetektion in *good features to track* [ST94] ist ähnlich zum Harris Corner Detector [HS88]. In beiden Fällen ist ein gutes Merkmal von den Eigenwerten abhängig. Durch die Eigenwertzerlegung können die Eigenwerte einer quadratischen Matrix G extrahiert werden. G kann in diesem Falle ein Merkmalsfenster von 15×15 Pixeln sein. Die Eigenschaften der Eigenwerte sind wie folgt:

- λ_1, λ_2 klein : Das Merkmalsfenster enthält eine homogene Fläche.
- λ_1 klein, λ_2 groß : Das Merkmalsfenster enthält ein unidirektionales Muster.

- λ_1, λ_2 groß : Das Merkmalsfenster enthält eine Ecke oder “salt-and-pepper“ Textur.

Ein Merkmalsfenster wird dann akzeptiert, wenn $\min(\lambda_1, \lambda_2) > \lambda$ ist. λ ist dabei ein Schwellwert, der nach Tomasi et al. [TK91] zufriedenstellende Ergebnisse mit $\lambda = \frac{1}{2}(\lambda_{\min} + \lambda_{\max})$ liefert. λ_{\min} entspricht dem Eigenwert einer Fläche mit einer nahezu homogenen Helligkeit und λ_{\max} entspricht dem Mittel der Eigenwerte von einer Auswahl von Kanten und stark texturierten Regionen.

Bei der Merkmalsverfolgung gehen Tomasi et al. davon aus, dass nur eine kleine Bewegung zwischen zwei Bildern stattgefunden hat. Das *aktuelle* Bild $I(t + \tau)$ resultiert somit durch die Verschiebung des *alten* Bildes $I(t)$. Es ist, als ob man sich vorher die aktuelle Szene von einer nur leicht unterschiedlichen Perspektive aus betrachtet hat und die beiden Bilder zwangsläufig eine hohe Ähnlichkeit aufweisen müssen. Gleichung 2.4 drückt diesen Zusammenhang aus

$$I(x, y, t + \tau) = I(x - \xi, y - \eta, t), \quad (2.4)$$

wobei (x, y) für die Pixelpositionen (räumliche Koordinaten), (ξ, η) für den Verschiebungsvektor und t für die Zeit stehen. Gesucht wird die 2D-Verschiebung \mathbf{d} des Merkmalsfensters. Dieses Problem kann nach einer Reihe von Zwischenschritten⁶ mit Hilfe der Newton-Raphson-Iteration gelöst werden, da die Ursprungsgleichung 2.4 auf

$$\mathbf{G}\mathbf{d} = \mathbf{e} \quad (2.5)$$

reduziert werden kann. Die Koeffizientenmatrix \mathbf{G} ist eine symmetrische 2×2 Matrix. Sie enthält die für die Merkmalsextraktion wichtigen Eigenwerte. Der zweidimensionale Vektor \mathbf{e} enthält die Intensitätsunterschiede der korrespondierenden Merkmalsfenster und \mathbf{d} entspricht dem gesuchten Verschiebungsvektor.

Die Konvergenz dieses Verfahrens ist schnell und erreicht zumeist schon nach fünf Iterationen eine subpixelgenaues Ergebnis. Bezogen auf den Einsatz dieses Verfahrens zur

⁶Die genaue Ausführung dieser Zwischenschritte ist in [TK91] nachzulesen.

visuellen Odometrie darf die Fluggeschwindigkeit des Versuchsträgers nicht zu groß werden, auch wenn die Bildsequenzen mit 30 Hz geliefert werden können, da sonst die Bewegungsunterschiede ebenfalls zunehmen und die Merkmalsverfolgung Gefahr läuft, ein Merkmal zu verlieren. Man könnte den Suchbereich für die Verschiebung erhöhen, nur wirkt sich dieses wiederum negativ auf die Laufzeit aus. Eine Methode zur weiteren Beschleunigung der Merkmalsverfolgung stellt der Pyramidenansatz von Bouguet [Bou02] dar.

2.4 Stand der Wissenschaft und eigener Ansatz

Die Schätzung der Eigenbewegung aus Bilddaten ist ein momentan sehr aktuelles Thema im Bereich der autonomen Navigation. Es wurden Verfahren entwickelt, die basierend auf Mono-Kameras, Stereokameras und sogar Multikamera-Konfigurationen – d.h. mit mehr als zwei Linsen – Lösungsansätze bieten. Der nun folgende Überblick über den Stand der Wissenschaft beschränkt sich dabei auf die auf Stereokameras basierenden Ansätze, da der UL-Abteilung bereits eine spezielle Stereokamera zur Verfügung steht.

Der Artikel von Sünderhauf et al. [SP07] gibt einen guten Einstieg in drei verschiedene Verfahren, die zur bildbasierten Bewegungsschätzung mit Stereokameras eingesetzt werden. Der erste Ansatz behandelt eine Lösung über die Singulärwertzerlegung [Ume91]. Sie setzt 3D-Punktmengen und deren Korrespondenzen voraus und berechnet die Rotation und Translation über deren Registrierung. Der zweite Ansatz wird zwar hauptsächlich im Mono-Bereich angewandt, kann aber auch für einen Stereoansatz benutzt werden. Da zwei Kameras zur Verfügung stehen, wird einfach auch eine doppelte Schätzung der *essential Matrix* [Hor90] durchgeführt, deren Skalierungsfaktor und Uneindeutigkeiten über einen einzelnen, rekonstruierten 3D-Punkt gelöst werden können. Der letzte Lösungsansatz fällt in den Bereich der Statistik und besteht in einem Ansatz über die *Maximum Likelihood* Methode, die die Rotation und Translation unter Einbeziehung der Tiefenunsicherheit berechnet [CMM06].

Oft ist der Grundansatz zur Schätzung der Eigenbewegung in vielen Verfahren gleich: Man wählt eine Anzahl an Merkmalen aus, verfolgt diese, filtert die Ausreißer heraus

und berechnet dann zwischen den pro Zeitabschnitt übriggebliebenen Punktmengen die relative Positions- und Lageänderung, die dann akkumuliert den Pfad der Eigenbewegung und den Lageänderungsverlauf ergeben. Zu jedem dieser Einzelschritte existieren eine Vielzahl von Ansätzen, so dass es verschiedenste Möglichkeiten gibt, ein eigenes System zur bildbasierten Bewegungsschätzung zu entwickeln. Die vorliegende Arbeit entwickelt zwar ein Gesamtkonzept, jedoch liegt der Schwerpunkt in der Filterung und Registrierung der Punktmengen. Dahingehend werden die nun folgenden, ausgewählten Publikationen beleuchtet.

In [ZON⁺06] wird von Naroditsky et al. die Filterung der Punktkorrespondenzen über den *Least Median of Squares* durchgeführt. Zunächst werden anhand einer Menge von Hypothesen jeweils die Rotation und die Translation berechnet. Es wird dann die Hypothese ausgewählt, die den kleinsten Wert nach dem *Least Median of Squares* enthält. Die resultierende Menge an Korrespondenzen wird dann dazu verwendet, eine letzte Verfeinerung der Schätzung mittels einer nicht weiter benannten nichtlinearen Optimierungsmethode durchzuführen. Ähnlich dazu, wurde in einem früheren Artikel von Naroditsky et al. [NNB04] Punktmengen durch eine Erweiterung des RANSAC gefiltert [FB81].

Ein anderer Ansatz zur visuellen Odometrie wurde von Nesnas et al. [NBM⁺04] in Erwägung gezogen. Die Gruppe hat in 2004 versucht, die Methode des ICP⁷ zur Verbesserung der Merkmalsverfolgung einzusetzen. Dieser Ansatz wurde jedoch verworfen, da die Ergebnisse unzufriedenstellend waren. Die Methode des ICP wurde aber von Milella et al. [MS06] in 2006 aufgegriffen und zur Schätzung der Eigenbewegung eingesetzt. Die durch den ICP iterativ berechneten Korrespondenzen werden in Milella et al. [MS06] durch die Methode von Zhang [Zha94] gefiltert. Die Berechnung der Registrierung basiert auf der schon Anfangs genannten Methode der Singulärwertzerlegung.

Der Ansatz, der in dieser Diplomarbeit verfolgt wird, orientiert sich hauptsächlich an dem zuletzt genannten Artikel von Milella et al. Dabei werden ähnliche Verfahren zur Ermittlung der Korrespondenzen, aber unterschiedliche Verfahren bezüglich deren Filterung analysiert. Diese Verfahren werden als Schwerpunkt im folgenden Kapitel erläutert.

⁷englisch: Iterative Closest Point, vgl. [BM92].

Kapitel 3

Bildbasierte Bewegungsschätzung

Die einzelnen Schritte, die zur Bestimmung der Eigenbewegung aus Stereo-Bilddaten notwendig sind, werden in diesem Kapitel erklärt. Hauptansatz ist dabei, die Informationen aus einer Serie von Intensitäts- und Disparitätsbildern so zu kombinieren und zu filtern, dass geeignete Punktmengen zur Registrierung erzielt werden und damit schließlich die Trajektorie der Kamera rekonstruiert werden kann. Angefangen mit der vom Kamera-winkel abhängigen Merkmalsausbeute, werden danach im Schwerpunkt drei Verfahren zu Filterung der Punktmengen erläutert. Darauf folgt die Ausführung des Verfahrens zur Registrierung der gefilterten Punktmengen, welches letztendlich in der Schätzung der Eigenbewegung resultiert. Die Berechnung der resultierenden Trajektorie schließt das Kapitel ab.

3.1 Merkmalsausbeute

Da das vorgesehene Verfahren auf der Registrierung von Punktmengen beruht, müssen die aus der in Kapitel 2.3 erklärten Merkmalsverfolgung stammenden Merkmale einen zuverlässigen Tiefenwert aufweisen. Zuverlässig bedeutet, dass ein gemessener Tiefenwert in einem Entfernungsbereich liegt, der von einer geringen Tiefenungenauigkeit betroffen ist. Aufgrund der quadratischen Zunahme der Tiefenungenauigkeit ist ein Tiefenwert umso besser, je näher dieser zur Kamera liegt.

Angenommen, die Stereokamera sei genau in Flugrichtung ausgerichtet (siehe Abbildung 3.1), so wäre ein Großteil des Bildes durch den Himmel ausgefüllt. Dieser Bereich ist sowohl zum Zweck der Merkmalsverfolgung als auch der Triangulierung von Tiefendaten nutzlos. Man verschenkt für das Verfahren also wertvolle Bildfläche. Für den Fall, dass ein Gebäude, Baum oder ein anderes Hindernis in Flugrichtung liegt und die Flughöhe gering ist, so hätte man zwar ein messbares Objekt zur Verfügung, jedoch ist aufgrund der zumeist großen Entfernung die Tiefeninformation relativ unzuverlässig und somit für den Beitrag zu einer verwertbaren Punktmenge zur robusten Registrierung eher ungeeignet.

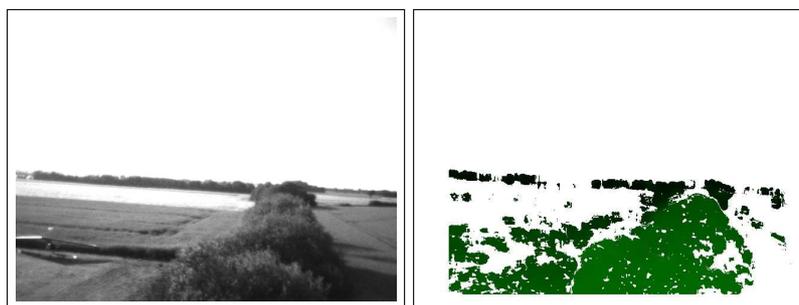


Abbildung 3.1: Intensitäts- und Tiefenbild, Stereokamera genau in Flugrichtung. Durch den hohen Anteil an Himmel wird wertvoller Bildanteil für den ICP-Ansatz verschenkt. Die Tiefenwerte des Tiefenbildes sind hier monochrom grün gefärbt. Diese Art der Darstellung der Tiefenwerte entspricht der Grundeinstellung im später erläuterten Framework des DLR.

Um eine höhere Merkmalsausbeute zu erreichen, könnte man die Stereokamera so ausrichten, dass diese schräg nach unten blickt. Kelly et al. [KS07] haben hierzu eine Untersuchung durchgeführt und einen zum Lot gemessenen Kamerawinkel von $66,5^\circ$ ermittelt. In dieser Arbeit wird der Winkel aus messtechnischen Gründen auf 70° gesetzt. Da der durchschnittliche Nickwinkel des Flugversuchsträgers während einem normalen Flug bei ca. 8° liegt (siehe Abbildung 3.2), ergibt sich für die Montierung der Stereokamera ein endgültiger Montierungswinkel von 62° . Der daraus resultierende Sichtbereich und die Anordnung während des Fluges ist in Abbildung 3.3 dargestellt.

Auch wenn die Kamera nun eine für das Verfahren günstigere Ausrichtung hat, so bedeutet das nicht, dass automatisch ein verfolgtes Intensitätsmerkmal auch einen Tiefenwert

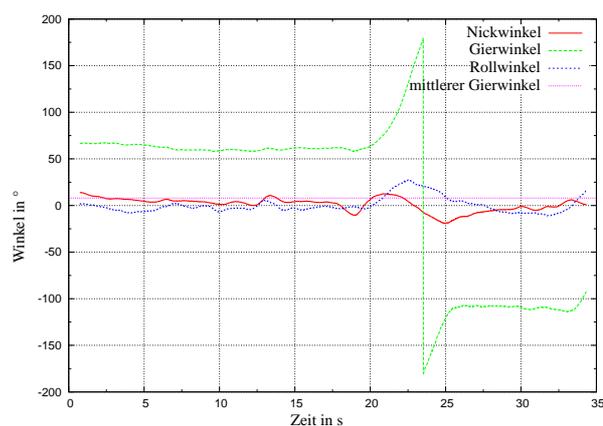


Abbildung 3.2: Fluglagedaten eines maxiARTIS-Flugversuches. Der Wechsel des Gierwinkels von 180° auf -180° wird durch den Wertebereich von $[-2\pi, 2\pi]$ vorgegeben und bedeutet keine Diskontinuität in der Drehung um die Gierachse.

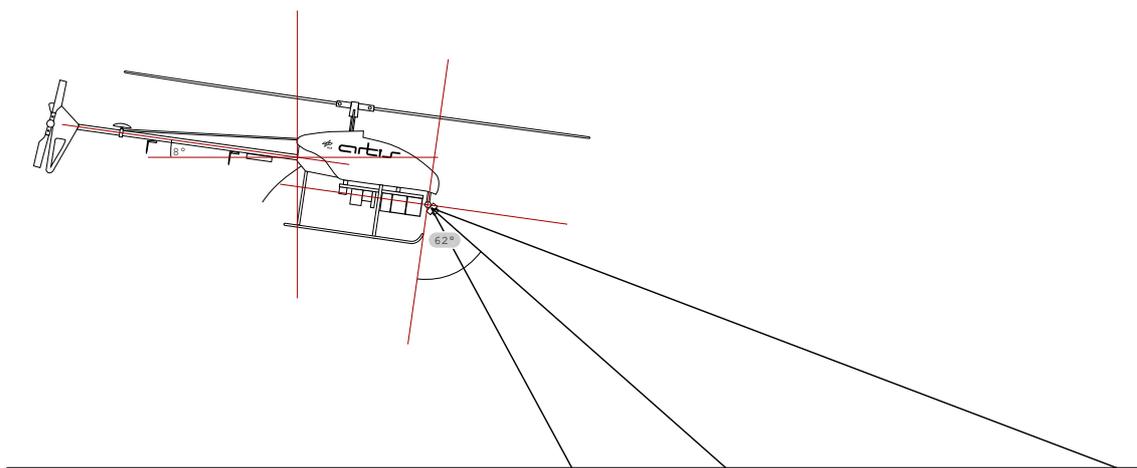


Abbildung 3.3: Fluglage des Versuchsträgers mit einem Kamerawinkel von 62° .

haben *muss*. Im Falle von über- oder unterbelichteten Regionen oder je nach Konfiguration der Stereokamera kann es sein, dass diese keinen Tiefenwert liefert, auch wenn die Merkmalsverfolgung immer noch in der Lage ist, Merkmale zu verfolgen oder zu generieren. Insofern sollte der Algorithmus für die Merkmalsverfolgung nur dort neue Merkmale generieren, wo auch ein gültiger Tiefenwert durch die Tiefenkarte zur Verfügung gestellt wird. Intensitätsmerkmale, die nicht dreidimensional rekonstruiert werden können, sind

für das in dieser Arbeit vorgestellte Verfahren nutzlos.

3.2 Filterung der Merkmale

Drei Verfahren zur Filterung werden nun näher ausgeführt. Alle haben zum Ziel, Ausreißer aus den Eingangsdaten zu entfernen um somit die Schätzung der Eigenbewegung so robust wie möglich zu machen. Das erste Verfahren nach der *Mahalanobis-Distanz* verwendet den iterativen Ansatz des ICP. Die beiden anderen Verfahren *Smooth Local 3D* und *RANSAC* nutzen die aus der Merkmalsverfolgung stammenden Korrespondenzen und brauchen dementsprechend nur eine Iteration zur Bestimmung der Eigenbewegung. Alle Verfahren werden nun im Einzelnen ausführlich beschrieben.

3.2.1 Mahalanobis-Distanz

Die Mahalanobis-Distanz ist ein Verfahren, welches erlaubt, die Zunahme in der Tiefenun- genauigkeit der Stereokamera zu berücksichtigen. Möglich wird dies durch eine Kovari- anzmatrix, die zwischen beiden Elementen einer Korrespondenz über deren Ähnlichkeit entscheidet. Das Distanzmaß ist mathematisch wie folgt definiert:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \quad (3.1)$$

Wenn man die Kovarianzmatrix Σ durch eine Einheitsmatrix ersetzt, sieht man, dass die Mahalanobis-Distanz eine erweiterte Form der euklidischen Distanz darstellt. Im Gegen- satz zur Mahalanobis-Distanz berechnet der euklidische Abstand die absolute, geometri- sche Entfernung zweier Punkte im Raum. Würde man diesen Abstand zur Filterung von Korrespondenzen benutzen, so würde man alle Punkte im Raum *gleich* behandeln. Da die Stereokamera eine mit der Tiefe zunehmende Unsicherheit in der Entfernungsschätzung aufweist, wäre der euklidische Abstand nicht die richtige Wahl.

Ein Beispiel soll dieses verdeutlichen: Nehmen wir an, wir haben zwischen den Elementen (\mathbf{x}, \mathbf{y}) einer Korrespondenz einen euklidischen Abstand von 50 cm. Sind (\mathbf{x}, \mathbf{y}) im Nahbe- reich der Kamera, also in dem Bereich, der von weniger Messungenauigkeit betroffen ist,

so wären diese 50 cm ein großer Wert und man könnte davon ausgehen, dass die Korrespondenz falsch ist. Sind (x, y) aber im Fernbereich, so würde man diese Korrespondenz nicht verwerfen, da auch die Messungenauigkeit größer ist. Mit Hilfe der Mahalanobis-Distanz lässt sich eine Tiefenabhängige Filterung realisieren.

Die vereinfachte Kovarianzmatrix Σ für diesen Fall bildet sich wie folgt

$$\Sigma = \begin{pmatrix} \frac{x_z + y_z}{f} & 0 & 0 \\ 0 & \frac{x_z + y_z}{f} & 0 \\ 0 & 0 & \frac{x_z + y_z}{f} \end{pmatrix} \quad (3.2)$$

und berücksichtigt dabei eine lineare Tiefenabhängigkeit, wobei die Unsicherheiten in x, y und z als unkorreliert angenommen wird. Ein Vergleich mit Formel 2.2 auf Seite 19 und dem daraus resultierenden Graphen bei Abbildung 2.6 zeigt, warum dieses ein denkbarer Ansatz ist: die quadratisch ansteigende Tiefenungenauigkeit wird durch die große Basislänge der Kamera gestreckt und somit vermindert die Funktion 2.2 in dem für das Verfahren zur visuellen Odometrie wichtigen Nahbereich der Stereokamera die Krümmung der Funktion der Tiefenauflösung (vgl. Abbildung 2.6). Zur Filterung wird ein Schwellwert definiert, der über die Eignung einer Korrespondenz entscheidet.

3.2.2 Smooth Local 3D

Der in dieser Arbeit verwendete Smooth Local 3D ist eine eigene Erweiterung des Smooth Local-Ansatzes von Goormann [Goo04] von 2D nach 3D. Grundgedanke ist die Analyse der Bewegungsvektoren der Korrespondenzen. Die *lokale* Ähnlichkeit in der Bewegung ist hier das Entscheidungsmerkmal, ob eine Korrespondenz als Ausreißer angesehen wird, oder nicht. Abbildung 3.4 zeigt das Prinzip des Verfahrens.

Auf der linken Seite sieht man ungefilterte Korrespondenzvektoren. Eindeutig sind eine Menge an Querschlägern erkennbar (die senkrecht verlaufenden Korrespondenzvektoren). Würde man diese ungefilterte Korrespondenzmenge direkt an den Registrierungsalgorithmus weitergeben, so würde das Ergebnis keine zufriedenstellende Lösung darstellen. Auf der rechten Seite der Abbildung sieht man das Resultat des Smooth Local 3D-Ansatzes:

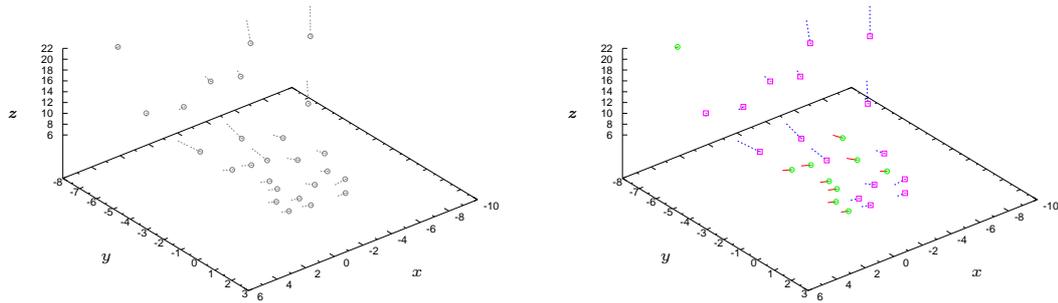


Abbildung 3.4: Ungefilterte Korrespondenzen (links) und das Resultat des Smooth Local 3D-Ansatzes (rechts). In der rechten Abbildung entsprechen die rechteckig gezeichneten Korrespondenzen den erkannten Ausreißern.

Die Korrespondenzvektoren mit einem rechteckigen Kopfteil wurden herausgefiltert.

Die mathematischen Grundlagen zum Smooth Local 3D-Ansatz werden nun ausgeführt. Gegeben sei eine Korrespondenzmenge R , die durch die Merkmalsverfolgung zweier Punkt-mengen A, B gebildet wird. A entspricht dem Ergebnis der Merkmalsverfolgung zum Zeitpunkt t_1 , B dem zum Zeitpunkt t_2 . Es existiert durch die Merkmalsverfolgung eine Zuordnung zwischen jedem $\mathbf{a} \in A$ und jedem $\mathbf{b} \in B$, die jeweils ein Paar $(\mathbf{a}_i, \mathbf{b}_i)$ erzeugt, mit $1 \leq i \leq |R|$. Zuerst berechnet man den mittleren Bewegungsunterschied zwischen einer Korrespondenz $(\mathbf{a}_j, \mathbf{b}_j)$ und den anderen mit

$$\Gamma_1 = \sum_{i=1}^{|R|} \frac{|(\mathbf{a}_j - \mathbf{b}_j) - (\mathbf{a}_i - \mathbf{b}_i)|}{|\mathbf{a}_j - \mathbf{a}_i|}. \quad (3.3)$$

Die Berechnung des mittleren Bewegungsunterschieds Γ_1 wird nun für alle anderen Korrespondenzen durchgeführt und ergibt so den mittleren Wert der Gleichung 3.3,

$$\Gamma_2 = \frac{1}{|R|} \sum_{j=1}^{|R|} \sum_{i=1}^{|R|} \frac{|(\mathbf{a}_j - \mathbf{b}_j) - (\mathbf{a}_i - \mathbf{b}_i)|}{|\mathbf{a}_j - \mathbf{a}_i|}, \quad (3.4)$$

also den *mittleren* mittleren Bewegungsunterschied Γ_2 . Die Filterung geschieht durch den

Vergleich der Ergebnisse von Γ_1 und Γ_2 . Ist der Bewegungsunterschied zwischen einer Korrespondenz und den restlichen Korrespondenzen über einem Schwellwert, wird diese aus der ursprünglichen Korrespondenzmenge R herausgefiltert ($R \rightarrow R'$) und die Rechnungen wiederholen sich, bis die Abbruchkriterien greifen:

1. $|R'|_{\min}$ wird unterschritten. R war demnach durch so viele Ausreißer *verunreinigt*, so dass wenigstens die Korrespondenzen weitergereicht werden sollen, die noch in der gefilterten Korrespondenzmenge enthalten sind.
2. Der Unterschied zwischen $\Gamma_{2(t-1)}$ und $\Gamma_{2(t)}$, also dem letzten und dem aktuellen *mittleren* mittleren Bewegungsunterschied ist zu klein. Dieses entspricht dem Konvergenzkriterium, das kaum noch eine Verbesserung der Filterung erreicht werden kann.
3. Der *mittlere* mittlere Bewegungsunterschied Γ_2 fällt zu gering aus. Ähnlich zum zweiten Kriterium lohnt sich hier auch keine weitere Iteration, da ebenfalls keine Verbesserung der Filterung erreicht werden kann.

Die nun folgende Auflistung verdeutlicht den Gesamtablauf des *Smooth Local 3D*-Verfahrens:

1. Nehme eine Korrespondenz und berechne deren Translationsvektor.
2. Berechne den Unterschied zwischen diesem Vektor und den restlichen Korrespondenzvektoren, gewichtet mit der jeweiligen Entfernung und akkumuliere diesen Unterschied zur gemittelten mittleren Bewegungsdistanz.
3. Akkumuliere diese gemittelte mittlere Bewegungsdistanz.
4. Wiederhole Schritte 1-3 solange, bis alle Vergleiche durchgeführt wurden.
5. Berechne die Ausreißer. Alle Korrespondenzen, die einen zu hohen Bewegungsunterschied zum gemittelten Wert aus Schritt 3 vorweisen, werden verworfen und die Korrespondenzmenge dementsprechend verkleinert.
6. Wiederhole Schritte 1-5 solange, bis die Akkumulation der mittleren Fehler zu gering wird, es zu wenig Restkorrespondenzen gibt oder die Differenz zwischen der letzten und der aktuellen gemittelten mittleren Bewegung ebenfalls zu klein wird.

3.2.3 RANSAC

Das RANSAC¹-Verfahren wurde von Fischler und Bolles [FB81] entwickelt und ist eine häufig eingesetzte Methode zur Optimierung von mit Ausreißern behafteten Daten. Das Verfahren ermittelt durch eine mehrfach wiederholte Modellschätzung deren beste Lösung. Das Ergebnis hängt davon ab, wie sehr die Daten mit Ausreißern durchsetzt sind und wie oft versucht wird, das Modell zu schätzen. Es ist nicht deterministisch und führt bei ausreichend vielen Versuchen zu einer guten Lösung. Diese Lösung wird Konsensmenge² genannt.

Im zweidimensionalen lässt sich der Ansatz zur ersten Erklärung gut nachvollziehen. Hierzu haben Fischler und Bolles ein Beispiel aufgeführt, welches die Problematik nahe bringt:

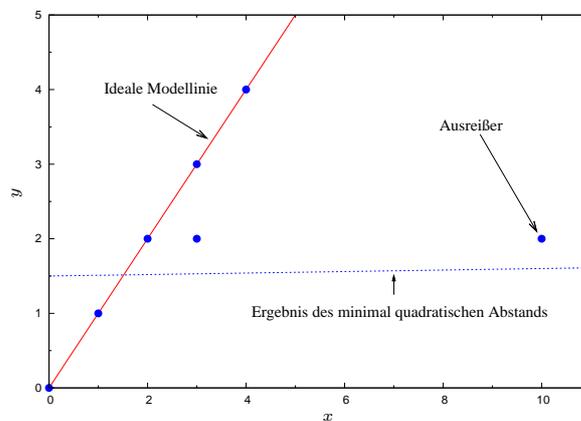


Abbildung 3.5: Versagen des Ansatzes der minimal quadratischen Fehler.

Entlang der idealen Modelllinie befinden sich vier Punkte. Ein weiterer Punkt liegt in der unmittelbaren Umgebung der Modelllinie und der letzte Punkt ist ein von der Modelllinie weiter entfernter Ausreißer. Würde man versuchen, die beste Gerade nach der Methode des minimal-quadratischen Fehlers zu finden, so würde man ein falsches Ergebnis erhalten. Nur ein einzelner, ungünstiger Messwert kann zum Ergebnis haben, dass die korrekte Lösung nicht gefunden werden kann.

¹englisch: *Random Sampling Consensus*.

²englisch: *Consensus set*.

Mit Hilfe des RANSAC-Verfahrens ist man in der Lage, die Modelllinie in dem oben gezeigten Beispiel zu finden. Zwei Punkte sind ausreichend zur Bestimmung der Geradengleichung $y = mx + b$. In diesem einfachen Beispiel könnte man alle $\binom{7}{2} = 21$ Modellkombinationen durchgehen, dabei die Abstände zu den restlichen Punkten berechnen, um die Lösung zu finden. Hat man jedoch ein Wissen über die vermutete Wahrscheinlichkeit des Auftretens der Ausreißer, so kann man mit weniger Versuchen auskommen. In dem Beispiel liegt die Wahrscheinlichkeit eines Ausreißers bei 15%, so dass die Anzahl der Versuche auf zwei bis drei verringert werden kann (vgl. [FB81], Seite 385).

Um mit einer Wahrscheinlichkeit P die richtige Lösung zu erhalten, muss man basierend auf der Wahrscheinlichkeit des Auftretens eines Ausreißers ϵ eine gewisse Anzahl an Versuchen m mit der für die Modellschätzung gegebener Mindestanzahl an Punkten p durchführen:

$$P = 1 - (1 - (1 - \epsilon)^p)^m \Leftrightarrow m = \frac{\ln(1 - P)}{\ln(1 - (1 - \epsilon)^p)} \quad (3.5)$$

Wendet man 3.5 auf den in dieser Diplomarbeit vorgestellten Ansatz an, so ist zunächst $p = 8$, da dieses die Mindestanzahl an Eingabewerten zur Registrierung ist. Der Wert für ϵ hängt davon ab, ob man synthetische oder reale Stereokameradaten hat, da zwangsläufig Unterschiede in der Wahrscheinlichkeit des Auftretens von Ausreißern zustande kommen werden. P richtet sich nach der gewünschten Sicherheit des Ergebnisses, was je nach ϵ eine große Anzahl an Versuchen bedeuten kann. Das RANSAC-Verfahren, angepasst auf die Schätzung der Eigenbewegung, wird wie folgt dargestellt:

1. Wähle per Zufall aus der Korrespondenzmenge acht Korrespondenzen aus.
2. Berechne \mathbf{R} , \mathbf{t} und nehme diese als ein Modell an.
3. Berechne, wie viele von den übrigen Korrespondenzen mit diesem Modell mit einem gewissen Fehler übereinstimmen. Diese Korrespondenzen werden der Modellmenge hinzugefügt und ergeben eine mögliche Konsensmenge.
4. Wenn die aktuelle Konsensmenge einen geringeren Registrierungsfehler *und* eine gleich große oder größere Mächtigkeit als die alte Konsensmenge hat, wird dieses Modell behalten, ansonsten verworfen.
5. Durchlaufe Schritte 1-4 mit der Anzahl der festgelegten Versuche.

3.3 Schätzung der Eigenbewegung

Die Schätzung der Bewegung der Stereokamera und somit auch die Bewegung des Helikopters zwischen zwei Zeitpunkten, lässt sich – wie Eingangs unter Abschnitt 2.4 erklärt – durch mehrere Methoden berechnen.

Das für diese Arbeit ausgewählte Verfahren nach Milella et al. [MS06] beruht auf dem *iterative closest point* Algorithmus (ICP) [BM92]. Es berechnet iterativ die Rotation und Translation zwischen zwei 3D-Punktmengen, die eine zuordenbare Schnittmenge besitzen. Dies bedeutet, dass die beiden Punktmengen nicht die gleiche Anzahl an Elementen haben müssen, jedoch ist es zwingend erforderlich – um auch die wirkliche Transformation zwischen den Punktmengen zu berechnen – eine ausreichend große korrespondierende Teilmenge zu erreichen.

Zu jedem Zeitpunkt t liefert die Stereokamera eine Tiefenkarte (Disparitätsbild) ihrer Umwelt. Aus dieser Tiefenkarte können die räumlichen Koordinaten rekonstruiert werden (siehe Kapitel 2.2) und ergeben eine 3D-Punktmenge. Zwei dieser Punktmengen dienen als Eingabe für den ICP-Algorithmus, der zwischen zwei Zeitpunkten t_1, t_2 die Bewegung berechnet, zerlegt in eine Rotation und eine Translation. Die Punktmenge zum Zeitpunkt t_1 wird als *Modellmenge* bezeichnet, wo hingegen die Punktmenge zum Zeitpunkt t_2 als *Datenmenge* bezeichnet wird. Die Datenmenge entspricht einer aktuellen Messung, welche mit der vorherigen Modellmenge registriert werden soll. Dieses bedeutet, dass die Datenmenge an die Modellmenge iterativ angenähert wird. Pro Iteration wird mittels der Singulärwertzerlegung (SVD) die zu diesem Schritt gehörende neue Teilrotation und Teiltranslation berechnet. Die Summe dieser Teilrotationen und Teiltranslationen ergibt zusammengerechnet die gesuchte Gesamtrotaion \mathbf{R} und Gesamttranslation \mathbf{t} . Der Algorithmus wird solange durchgeführt, bis ein Konvergenzkriterium erreicht wird. Ein Vergleich des Registrierungsfehlers über den euklidischen Abstand zwischen zwei Iterationen ist ein mögliches Konvergenzkriterium. Wenn die Differenz des Registrierungsfehlers zwischen zwei Iterationen unter einem Schwellwert liegt, wird der ICP abgebrochen.

Der mathematische Ansatz des ICP wird nun basierend auf der Ausführung von Nüchter et al. [NLHS05] und Konolige et al. [SK08]³, wobei deren Arbeiten wiederum auf Besl et

³siehe dort in Kapitel 22, Seite 536.

al. [BM92] und Arun et al. [AHB87] zurückzuführen sind, erklärt. Das Fehlermaß

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2 \quad (3.6)$$

drückt aus, dass der quadratische Abstand zwischen der Modellmenge M und der Datenmenge D durch die am besten geeignete Rotation \mathbf{R} und Translation \mathbf{t} minimiert werden soll. Dabei wird über alle Punkte aus M und D iteriert, wobei $\mathbf{m} \in M$ und $\mathbf{d} \in D$ ist. Der Faktor $w_{i,j}$ drückt dabei die Korrespondenz zwischen zwei Elementen \mathbf{m} und \mathbf{d} aus – 0 für eine falsche und 1 für eine richtige Zuordnung.

Pro Iterationsschritt des ICP müssen zwei Rechnungen durchgeführt werden. Zuerst müssen Korrespondenzen zwischen den beiden Punktmengen berechnet werden.⁴ Nach der Zuordnung der Punkte werden \mathbf{R} und \mathbf{t} über die Singulärwertzerlegung ermittelt und mit der Datenmenge D verrechnet ($\mathbf{R}\mathbf{d}_j + \mathbf{t}$). Da die Korrespondenzen via *Closest Point* berechnet werden, können sich in der nächsten Iteration⁵ des ICP neue Korrespondenzen ergeben, da man die Datenmenge *näher* an die Modellmenge transformiert hat. Der ICP läuft so lange, bis \mathbf{R} und \mathbf{t} konvergiert sind – die Modellmenge ist nun bestmöglich mit der Datenmenge registriert worden, weitere Iterationen des ICP würden keine Verbesserungen in der Berechnung von \mathbf{R} und \mathbf{t} bringen. Hierbei ist es wichtig darauf hinzuweisen, dass die beim letzten Schritt ermittelten Korrespondenzen als richtig angenommen werden. Hier liegt auch eines der Probleme des ICP: Falsche Punktkorrespondenzen ergeben auch falsche Werte für \mathbf{R} und \mathbf{t} . Dieses Problem der richtigen Zuordnung wurde in Kapitel 3.2 behandelt.

Zur Berechnung der jeweiligen Rotation und Translation, wird pro Iteration des ICP eine Abwandlung der Gleichung 3.6 benutzt.

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})\|^2 \quad (3.7)$$

Der *Closest Point*-Schritt ist der Grund für die verkürzte Schreibweise. Es wird dabei

⁴daher die letzten beiden Worte des Algorithmus' *Closest Point*.

⁵daher das erste Wort des Algorithmus' *Iterative*.

davon ausgegangen, dass die Punktkorrespondenzen bereits bekannt sind. Nur gültige Zuordnungen zur Schätzung von \mathbf{R} und \mathbf{t} werden verwendet, ausgedrückt durch $N = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j}$.⁶ Um die Gleichung 3.7 zu minimieren, wird ein Ansatz über die Singulärwertzerlegung verfolgt. Dieser ist laut Nüchter et al. [NLHS05] robust und gleichzeitig einfach zu implementieren. Der wichtigste Schritt besteht darin, die Rotation von der Translation zu entkoppeln, da die Rotationsmatrix \mathbf{R} zwingend orthogonal sein muss. Hierzu berechnet man zuerst den Schwerpunkt beider Korrespondenzmengen:

$$\mathbf{c}_m = \frac{1}{N} \sum_{i=1}^N \mathbf{m}_i, \quad \mathbf{c}_d = \frac{1}{N} \sum_{i=1}^N \mathbf{d}_i \quad (3.8)$$

Die zwei resultierenden Vektoren werden anschließend von ihrer jeweiligen Punktmenge abgezogen und ergeben durch die folgende Mengenkonstruktionen die Zwischenmengen M' und D' :

$$M' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\dots,N}, \quad D' = \{\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d\}_{1,\dots,N} \quad (3.9)$$

Setzt man nun 3.8 und 3.9 in Gleichung 3.7 ein, so erhält man folgendes Ergebnis:

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{\tilde{\mathbf{t}}}\|^2 \quad (3.10)$$

Durch die Verschiebung um den Schwerpunkt kann Term $\tilde{\mathbf{t}} = 0$ angenommen werden. Hieraus folgt, dass man die Minimierungsfunktion getrennt von der Translation nur durch eine Rotation ausdrücken kann:⁷

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 \quad (3.11)$$

Mit Hilfe der Singulärwertzerlegung lässt sich die optimale Rotation zwischen den Punkt-mengen M' und D' berechnen. Hierzu muss man zunächst eine 3x3 Korrelationsmatrix \mathbf{H}

⁶Mehrfachzuordnungen sind hierdurch möglich.

⁷Beweise und weitere Ausführungen sind ausführlich in [NLHS05] beschrieben.

bilden:

$$\mathbf{H} = \sum_{i=1}^N \mathbf{m}'_i{}^T \mathbf{d}'_i = \sum_{i=1}^N \begin{pmatrix} \mathbf{m}'_{ix} \mathbf{d}'_{ix} & \mathbf{m}'_{ix} \mathbf{d}'_{iy} & \mathbf{m}'_{ix} \mathbf{d}'_{iz} \\ \mathbf{m}'_{iy} \mathbf{d}'_{ix} & \mathbf{m}'_{iy} \mathbf{d}'_{iy} & \mathbf{m}'_{iy} \mathbf{d}'_{iz} \\ \mathbf{m}'_{iz} \mathbf{d}'_{ix} & \mathbf{m}'_{iz} \mathbf{d}'_{iy} & \mathbf{m}'_{iz} \mathbf{d}'_{iz} \end{pmatrix} \quad (3.12)$$

Diese Korrelationsmatrix \mathbf{H} wird per Singulärwertzerlegung in die drei Matrizen $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ zerlegt. Die Rotation ergibt sich durch:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T. \quad (3.13)$$

Da das Minimierungsproblem rein auf die Rotation beschränkt wurde, muss man nun als letzten Schritt die gewonnene Rotationsmatrix \mathbf{R} zur Berechnung der Translation \mathbf{t} verwenden:⁸

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d \quad (3.14)$$

Pro Iteration berechnet der ICP eine optimale Rotation \mathbf{R} und Translation \mathbf{t} für die jeweilig ermittelten Punktkorrespondenzen. Die Gesamtrotaion und -translation erhält man durch die Akkumulierung der Einzelrotationen und -translationen:

$$\mathbf{R} = \prod_{i=1}^n \mathbf{R}_i, \quad \mathbf{t} = \sum_{i=1}^n \mathbf{R}_i \cdot \mathbf{t}_i \quad (3.15)$$

Die Häufigkeit n der Produktbildung beziehungsweise der Summation hängt vom Konvergenzverhalten ab und entspricht der Anzahl der benötigten Iterationen. Die Rotation \mathbf{R}_i sorgt dafür, dass die Translation auf das richtige Koordinatensystem bezogen wird.

Die iterative Ausprägung des ICP ergibt sich rein aus den immer wiederkehrenden Neuberechnungen der unbekanntenen Korrespondenzen. Das Verfahren wird häufig dazu benutzt, LIDAR-Daten zu registrieren. Benutzt man ausschließlich diesen Sensor zur Raumanalyse, so hat man zunächst keine Möglichkeit, die Punktkorrespondenzen vor der Berechnung

⁸vgl. hierzu Gleichung 3.10.

des ICP zu erhalten. Ganz im Gegensatz dazu hätte man bei kombinierter Verwendung eines bildgebenden Verfahrens mit einer Tiefenkarte die Möglichkeit, die *mutmaßlichen* Korrespondenzen aus der Merkmalsverfolgung zu erhalten (siehe Kapitel 2.3) und den ICP auf eine einmalig durchgeführte SVD zu reduzieren. Für den iterativen Ansatz wird in dieser Arbeit die *Mahalanobis-Distanz* zur Filterung eingesetzt. Der andere Fall wird mit dem *Smooth Local 3D* und dem *RANSAC* untersucht.

3.4 Berechnung der Trajektorie

Zu jedem Zeitpunkt t wird durch den ICP – oder einer einmaligen SVD – eine relative Positions- und Lageänderung ermittelt. Die Akkumulation dieser Änderungen ergibt eine Trajektorie, die die Kamera zurückgelegt hat. Die Berechnung der aktuellen Position und Lage \mathbf{P}_t der Kamera ergibt sich durch die Matrixmultiplikation der bisherigen Akkumulation \mathbf{P}_{t-1} mit der aktuellen, relativen Positions- und Lageänderung \mathbf{V}_t .

$$\left(\begin{pmatrix} \mathbf{O}_t \\ 0 \ 0 \ 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_t \\ 1 \end{pmatrix} \right) = \mathbf{P}_t = \mathbf{P}_{t-1} \cdot \mathbf{V}_t, \quad \mathbf{V}_t = \left(\begin{pmatrix} \mathbf{R}_t \\ 0 \ 0 \ 0 \end{pmatrix} \begin{pmatrix} \mathbf{t}_t \\ 1 \end{pmatrix} \right). \quad (3.16)$$

Die so ermittelte aktuelle Position \mathbf{p}_t und Lage \mathbf{O}_t ist momentan noch auf das Kamerakoordinatensystem bezogen. Um diese Trajektorie im richtigen Bezugssystem zu erhalten, müssen zwei Koordinatentransformationen durchgeführt werden. Abbildung 3.6 veranschaulicht diese Schritte.

Der Punkt \mathbf{p}^c muss zuerst in Helikopterkoordinaten umgerechnet werden:

$$\mathbf{p}^h = \mathbf{M}^{c \rightarrow h} \cdot \mathbf{p}^c, \quad (3.17)$$

$\mathbf{M}^{c \rightarrow h}$ entspricht dabei einer Rotation von $(-\alpha_{cam} - \frac{\pi}{2})$ um die x -Achse, gefolgt von einer Rotation von $-\frac{\pi}{2}$ um die z -Achse. α_{cam} gibt an, mit welchem Winkel die Stereokamera

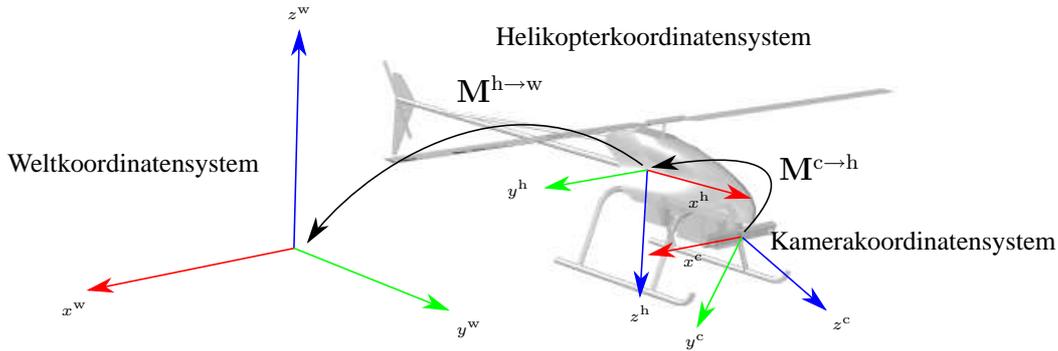


Abbildung 3.6: Die verwendeten Koordinatensysteme und deren Transformationen.

am Helikopter montiert ist. Dabei wird keine Verschiebung der Koordinatensystem angenommen und nur ein Freiheitsgrad in der Rotation betrachtet:

$$\mathbf{M}^{c \rightarrow h} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\alpha_{cam} - \frac{\pi}{2}) & \sin(-\alpha_{cam} - \frac{\pi}{2}) \\ 0 & -\sin(-\alpha_{cam} - \frac{\pi}{2}) & \cos(-\alpha_{cam} - \frac{\pi}{2}) \end{pmatrix} \quad (3.18)$$

Der letzte Schritt ist die Transformation der Helikopterkoordinaten in die Weltkoordinaten⁹, mittels einer Rotation von π um die x -Achse:

$$\mathbf{p}^w = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}}_{\mathbf{M}^{c \rightarrow w}} \cdot \mathbf{M}^{c \rightarrow h} \cdot \mathbf{p}^c \quad (3.19)$$

Bisher wurde nur auf die Transformation der Position eingegangen. Die Transformation der Lage ergibt sich analog zu 3.19:

$$\mathbf{O}^c = \mathbf{M}^{c \rightarrow w} \cdot \mathbf{O}^c \quad (3.20)$$

⁹Als Weltkoordinatensystem wird das Koordinatensystem von Blender angenommen, da in diesem System auch die Ground-Truth-Trajektorien definiert sind.

Kapitel 4

Implementierung

Dieses Kapitel geht auf die Umsetzung der im Konzept beschriebenen Verfahren ein. Angefangen mit der Einführung in die Zielumgebung, werden anschließend die wichtigsten Bestandteile der Implementierung aufgeführt und erklärt. Das Kapitel wird mit der Erklärung der grafischen Schnittstelle beendet.

4.1 Das DIP-Framework

Die Abteilung für unbemannte Luftfahrzeuge des DLR setzt im ARTIS-Projekt ein speziell entwickeltes Framework zur Bildverarbeitung ein. Dieses Framework, genannt DIP¹, ist in der Hochsprache C++ geschrieben und dient als Basis zur Integration von Bildverarbeitungsalgorithmen oder autonomierelevanter Verfahren. Die im Framework vorhandenen Funktionen und Filter werden über die grafische Benutzerschnittstelle *SPICE*², entwickelt von Goorman und Guth in 2004 [Goo04][Gut04], gesteuert.

Die Funktionen und Filter des DIP-Frameworks können über SPICE beliebig als Eingaben oder Ausgaben kombiniert werden, so dass bei der Neuentwicklung eines Prototypen die schon implementierte Funktionen und Filter effizient wiederbenutzt werden können. Zum

¹Digital Image Processing.

²Smart Program for Image Computing Experiments.

Beispiel ist die Infrastruktur für die Stereobildverarbeitung und die Merkmalsverfolgung schon vorhanden, so dass nur darauf aufgebaut werden muss. Teile des DIP-Frameworks nutzen zur Bildverarbeitung direkt die OpenCV³-Bibliotheken von Intel. Ausgaben, die einer 3D-Darstellung bedürfen, werden mit Hilfe der integrierten OSG-Bibliothek⁴ realisiert. Die Oberfläche der Benutzerschnittstelle SPICE basiert auf dem FLTK⁵ (siehe Abbildung 4.1).

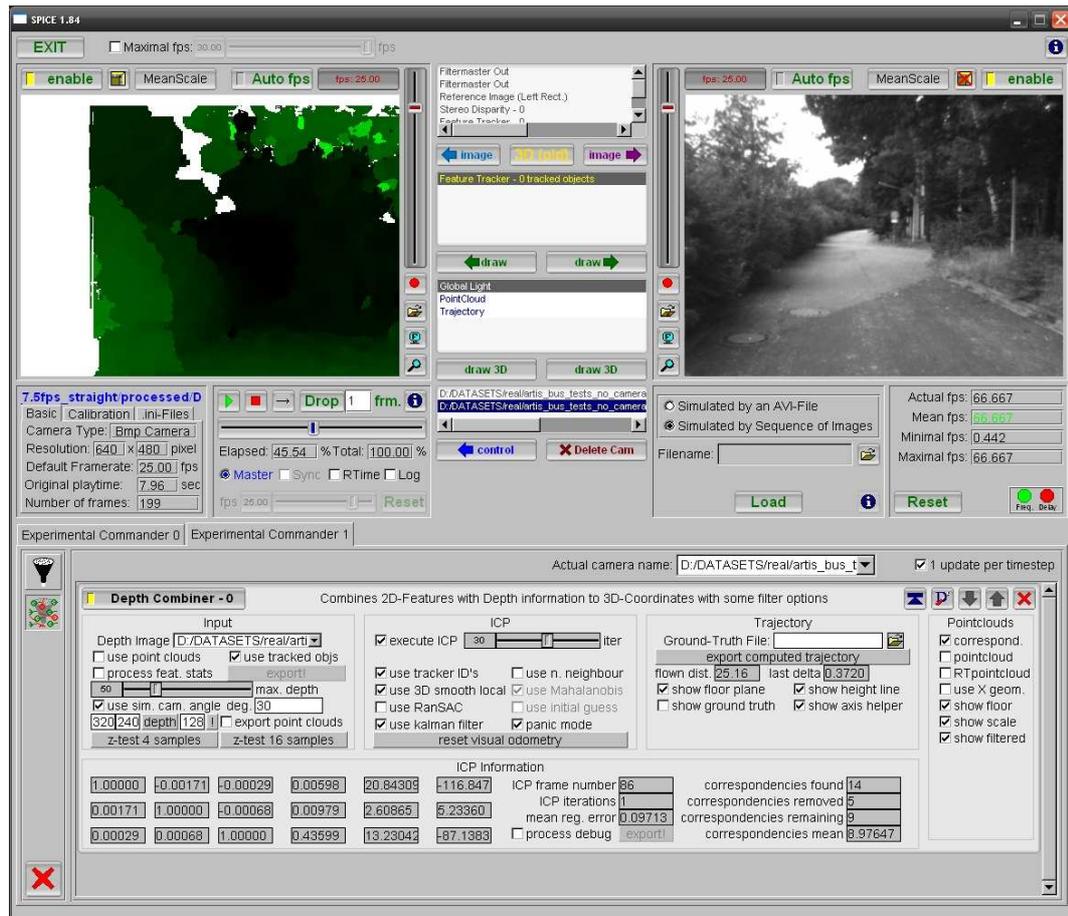


Abbildung 4.1: Die grafische Benutzerschnittstelle SPICE.

³Open Computer Vision.

⁴Open Scene Graph.

⁵Fast Light Toolkit.

4.2 Beschreibung der Implementierung

Dieser Abschnitt erklärt den funktionalen Aufbau der Implementierung.⁶ Hierzu dient die folgende Abbildung 4.2:

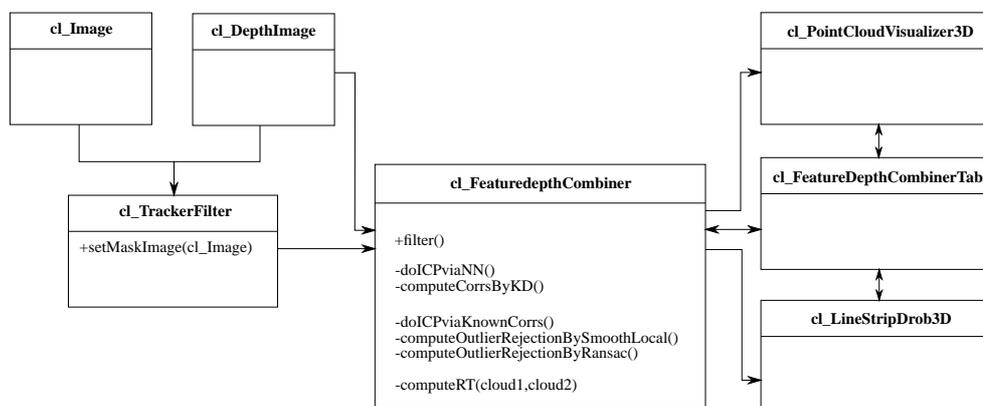


Abbildung 4.2: Der funktionale Zusammenhang der Implementierung. Pfeile markieren die Richtung des Datenstromes. Nur die wichtigsten Methoden wurden aufgeführt.

Ausgehend von zwei Bildern, einem Intensitätsbild vom Typ `cl_Image` und einem Tiefenbild vom Typ `cl_DepthImage`, werden diese zunächst an den KLT-Merkmalverfolger⁷ `cl_TrackerFilter` weitergegeben. Das Intensitätsbild dient dabei als Grundlage für die Merkmalsverfolgung, wobei das Tiefenbild als Maske für die Merkmalsgenerierung benutzt wird - es sollen ausschließlich dort Merkmale erzeugt werden, für die auch ein gültiger Tiefenwert zu finden ist.

Der TrackerFilter reicht die ermittelten 2D-Korrespondenzdaten an das Herzstück des Verfahrens, den `cl_FeatureDepthCombiner`, weiter. Dieser hat auch Zugriff auf das Tiefenbild und kombiniert zunächst einmal die 2D-Korrespondenzen mit den Tiefendaten zu räumlichen Koordinaten, also den benötigten Punktmengen für die Registrierung. Danach wird die Eigenbewegung geschätzt.

Je nachdem, welches Filterverfahren ausgewählt wurde, werden dabei verschiedene Pro-

⁶Aufgrund abteilungsinterner Richtlinien wird die englische Sprache zur Implementierung benutzt.

⁷Dieser nutzt die Implementierung aus OpenCV.

grammpfade durchlaufen. Diese Pfade sind in Abbildung 4.3 dargestellt.

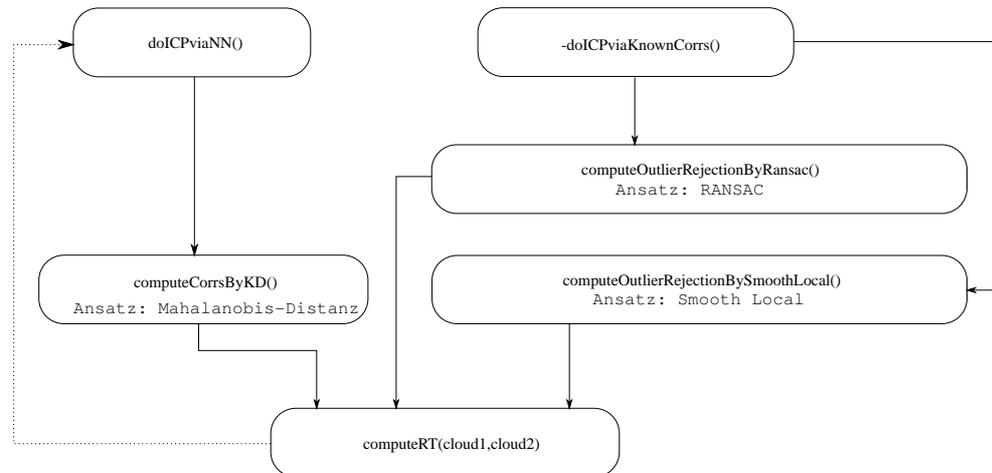


Abbildung 4.3: Die verschiedenen Programmpfade der jeweiligen Verfahren zur Filterung.

Der linke Pfad behandelt den iterativen⁸ Lösungsansatz über den ICP und wird durch die Methode `doICPviaNN()` betreten. Das hier benutzte Filterverfahren nach der *Mahalanobis-Distanz* ist direkt in der Methode `computeCorrsByKD()` implementiert. Der rechte Pfad behandelt den nicht iterativen Ansatz. Dieser wird über die Methode `doICPviaKnownCorrs()` gestartet und führt, je nach Auswahl die Filterverfahren *Smooth Local 3D* mittels der Methode `computeOutlierRejectionBySmoothLocal()`, oder *RANSAC* mittels der Methode `computeOutlierRejectionByRansac()` aus. Das Ergebnis der Filterung, die nun übrig gebliebenen Korrespondenzen, geht nun an die Registrierungsberechnung `computeRT(cloud1, cloud2)`, die mittels der Singulärwertzerlegung gelöst wird.

Das Ergebnis der Eigenbewegung wird über ein OSG-Fenster ausgegeben. Die Klasse zur Visualisierung heißt `cl_LineStripDrob3D` und stellt die Trajektorie grafisch dar. Falls man eine zusätzliche Visualisierung der Filterung der Punktmengen benötigt, kann man die weitere Darstellungsklasse `cl_PointCloudVisualizer3D` verwenden, die ebenfalls auf OSG basiert.

Sowohl die Klasse `cl_FeatureDepthCombiner` als auch die beiden OSGAusga-

⁸Erkennbar durch die gestrichelte, zurückgeführte Linie.

beklassen `cl_PointCloudVisualizer3D` und `cl_LineStripDrob3D` werden über die grafische Benutzeroberflächenklasse `cl_FeatureDepthCombinerTab` gesteuert.

4.3 Beschreibung der grafischen Schnittstelle

Die Bedienung des implementierten Gesamtverfahrens zur Schätzung der Eigenbewegung wird anhand folgender Abbildung 4.4 erklärt:

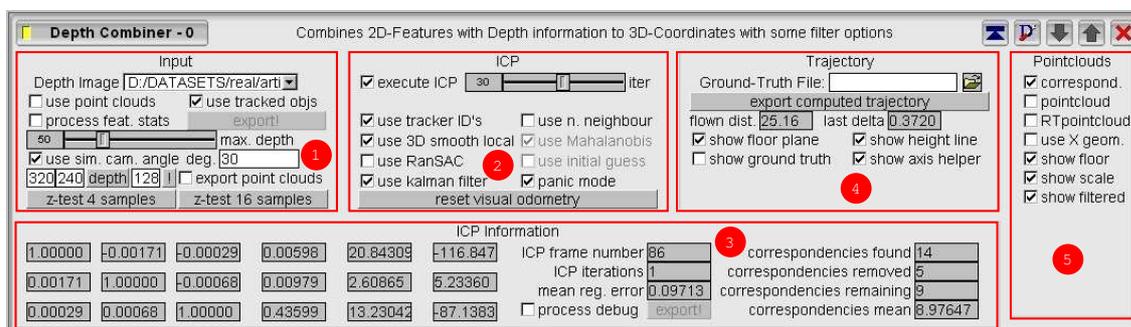


Abbildung 4.4: Die grafische Benutzerschnittstelle der Implementierung.

Zur Vereinfachung ist Abbildungen 4.4 in fünf Bereiche untergliedert, die nun der Reihe nach beschrieben werden:

1. Der erste Bereich betrifft hauptsächlich die Eingabedaten für die Berechnung der Eigenbewegung. Unter *Depth Image* legt man zunächst einmal die Referenz zum Eingabetiefenbild fest. Danach kann man sich entscheiden, ob man die Punktmengen aus dem gesamten Tiefenbild generiert haben möchte (*use point clouds*), oder ob man die aus der Merkmalsverfolgung stammenden Korrespondenzen zur Punktwolkgenerierung benutzen möchte (*use tracked objs*). Die Auswahlbox *process feat. stats* sorgt dafür, dass während der Laufzeit eine Statistik der Merkmalsverfolgung angefertigt wird, die über die Schaltfläche *export!* in eine Textdatei ausgegeben werden kann. Der Schieberegler *max depth* sorgt für eine Begrenzung der Tiefenwerte. Die resultierende Punktmenge wird also keine Tiefenwerte enthalten, die größer als

der eingestellte Wert ist. Ein besonders wichtiger Parameter ist die Einstellung des Kamerawinkels α_{cam} über *use sim. cam* und *angle deg*. Je nachdem, mit welchem Kamerawinkel ein Experiment durchgeführt wurde, muss man diesen gemessen von der Horizontalen hier einstellen. Falls man die Punktmengen in anderen Anwendungen oder zu weiteren Auswertungen benutzen möchte, so kann man diese über *export point clouds* exportieren. Die restlichen Schaltflächen dieses rot abgetrennten Bereiches dienen der schnellen Auswertung von Tiefenkarten.

2. Über den zweiten Bereich *ICP* wählt man hauptsächlich das zu verwendende Verfahren zur Schätzung der Eigenbewegung aus. Man hat zunächst zwei Auswahlmöglichkeiten: Zum einen kann man sich für die Benutzung der Merkmalskorrespondenzen entscheiden (*use tracker ID's*), oder nicht (*use n. neighbour*). Ersteres ermöglicht die Auswahl der zwei Verfahren Smooth Local 3D (*use 3D smooth local*) oder RANSAC (*use RanSAC*). Falls man sich dazu entscheidet, auf die Merkmalskorrespondenzen zu verzichten und lieber die Korrespondenzen per Nachbarschaftssuche bestimmen lassen will, so kann man nur das Verfahren der Mahalanobis-Distanz auswählen. Da die Mahalanobis-Distanz für das iterative Verfahren eingesetzt wird, kann man die Auswahlbox *use initial guess* verwenden, die dazu dient, die aktuelle Punktmenge mit der Rotation und Translation aus dem vorherigen Schritt an die andere Punktmenge schon einmal anzunähern, damit eventuell Iterationen gespart werden können. Die beiden unteren und letzten Auswahlboxen *use kalman filter* und *panic mode* können für alle Verfahren eingesetzt werden. Erstere dient rein der Tiefpassfilterung der Rotationen und Translationen und letztere dient dazu, im Falle einer Merkmalsunterversorgung oder einem äußerst schlechten Registrierungsergebnisses, die aktuelle Berechnung der Rotation und Translation zu verwerfen. So wird auf die vorherige Schätzung von \mathbf{R} und \mathbf{t} zurückzugreifen, um einem drohenden, kompletten Verlust der Bewegungsschätzung zu kompensieren.
3. Der dritte Bereich *ICP Information* dient hauptsächlich der Ausgabe von Werten, die zur Fehleranalyse benutzt werden können. Die linken drei Spalten enthalten die Komponenten der Rotation \mathbf{R} , gefolgt von den Spalten für die Translation \mathbf{t} in Kamerakoordinaten, der Position in Weltkoordinaten und der Lage in Weltkoordinaten. *ICP frame number* gibt die aktuelle Anzahl der relativen Positions- und

Lageschätzungen an. Darunter wird die Anzahl der verwendeten Iterationen und der mittleren Registrierungsfehler angegeben. Mit Hilfe der Auswahlbox *process debug* können nahezu alle anfallenden Werte der getätigten Berechnungen mitprotokolliert und über *export!* ausgegeben werden. Die letzte Ausgabespalte in diesem rot umrandeten Bereich gibt Informationen über die Gesamtzahl an Korrespondenzen (*correspondencies found*), die Anzahl der durch das eingesetzte Filterverfahren gelöschten Korrespondenzen (*correspondencies removed*), die Anzahl der noch bleibenden Korrespondenzen für die Registrierung (*correspondencies remaining*) und die im Mittel erreichte Korrespondenzanzahl (*correspondencies mean*).

4. Der vierte Bereich *Trajectory* widmet sich hauptsächlich den Einstellungsparametern für die OSG-Ausgabe der Trajektorie der geschätzten Eigenbewegung, realisiert durch die Darstellungsklasse `cl_LineStripDrob3D`. Über *Ground-truth File* lässt sich eine Referenztrajektorie in die Darstellung laden. So kann man direkt erkennen, wie gut die Schätzung durch das verwendete Verfahren ist. Die geschätzte Trajektorie lässt sich über die Schaltfläche *export computed trajectory* exportieren. Die Werte hinter *flown dist.* und *last delta* entsprechen der Länge der Trajektorie und der letzten Bewegungs-Zunahme. Abschließend kann durch Auswahlboxen die Sichtbarkeit der einzelnen Darstellungselemente des OSG-Fensters für die Trajektorie beeinflusst werden.
5. Mit dem fünften und letzte Bereich kann man die OSG-Darstellung der Punktmengen beeinflussen. Jeder der Auswahlboxen setzt dazu in der Darstellungsklasse `cl_PointCloudVisualizer3D` die entsprechenden Variablen, die die Sichtbarkeit der Darstellungselemente steuern. Abbildung 4.5 zeigt ein Ausgabebeispiel für sowohl die Trajektorien- als auch die Punktmengenansicht.

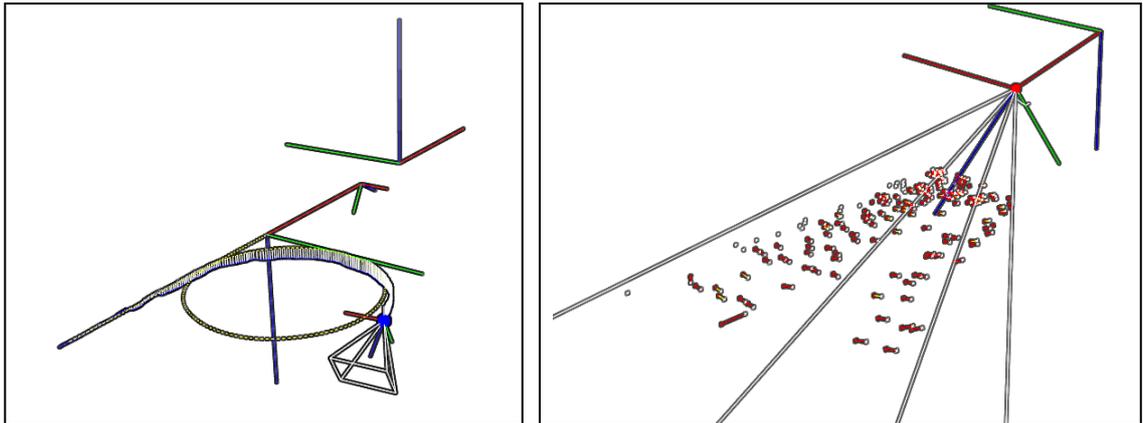


Abbildung 4.5: Die OSG-Ausgabefenster: Die Trajektorienansicht (links) und die Punktmengenansicht (rechts). Der eigentlich schwarze Hintergrund wurde zu Darstellungszwecken entfernt. Im linken Bild sind drei Darstellungselemente sichtbar: die verschiedenen Koordinatenachsen, eine Kreisbahn als Referenztrajektorie und die geschätzte Trajektorie mit Höhenlinien. Die geschätzte Trajektorie wird angeführt von einer Blickpyramide, welche die Kamera symbolisiert. Im rechten Bild sind wiederum die Koordinatenachsen und die Blickpyramide der Kamera zu sehen. Innerhalb der Blickpyramide werden die Punktmengen und Korrespondenzen dargestellt. Gültige Korrespondenzen werden mit einer gelben Linie, ungültige mit einer roten Linie dargestellt.

Kapitel 5

Experimente und Ergebnisse

In dem folgenden Kapitel wird zunächst auf die Erstellung der synthetischen Testdaten eingegangen. Anschließend werden die Versuchsansätze und Experimente zur Schätzung der Eigenbewegung präsentiert und erklärt. Darauf folgt die Auswertung der Ergebnisse. Grundlage für dieses Kapitel bilden zum einen synthetische Kamerafahrten und zum anderen eine Testreihe mit realen Bilddaten.

5.1 Erstellung von synthetischen Testdaten

Um den Algorithmus für die visuelle Odometrie testen und anpassen zu können, benötigt man zunächst Testdaten, die optimal ausgewertet werden können. Bei jedem Flugversuch des Versuchsträgers mit der Stereokamera als Nutzlast, werden die anfallenden Navigations- und Bilddaten zur Evaluierung aufgezeichnet. Die bisherigen Flugaufnahmen sind hauptsächlich im Kontext der Hindernisvermeidung mit einer genau in Flugrichtung ausgerichteter Stereokamera durchgeführt worden. Wie in Kapitel 3.1 schon beschrieben wurde, sind diese Daten nicht optimal für die visuelle Odometrie. Eine kurze Sichtung der Tiefenbilder der vergangenen Flugversuche ergab, dass ein gesonderter Flugversuch zur Untersuchung der visuellen Odometrie nötig ist.

Dieser geplante Flugversuch konnte leider nicht mehr geflogen werden, da der Versuchs-

träger inklusive der Stereokamera während einer Mission stark beschädigt wurde. Die Stereokamera ist daraufhin zum Hersteller geschickt und repariert worden. Die *alte* Hardware ist dabei durch eine neue ersetzt worden, die nicht mehr den alten Protokollstandards entspricht. Somit ist es im Rahmen dieser Arbeit nicht möglich, die Kamera im Flugversuch einzusetzen. Die einzige Möglichkeit, Bilddaten mit der Stereokamera zu generieren, besteht in der Nutzung der mitgelieferten SVS¹-Software. Die Software wird nicht im Flugversuch eingesetzt und hat zwei Nachteile: Zum einen sind nur maximal 200 Einzelbilder pro Sequenz mit einer Bildwiederholrate von 30 Hz aufnehmbar. Würde man mehr aufnehmen wollen, so erreicht man auf dem Testrechner leider nur eine durchschnittliche Bildwiederholungsrate von ca. 5 Hz, da direkt auf die Festplatte geschrieben wird. Zum anderen erhält man durch die Verwendung der SVS-Software nicht die Navigationsdaten eines Flugversuches. Dieser letzte Punkt erschwert letztendlich die Auswertung der Ergebnisse, da keine Referenztrajektorie verfügbar ist.

Aufgrund der genannten Problematik wurde ein Simulationssystem erstellt, welches die folgenden Vorteile bietet:

1. Einsparung von Zeit und Material,
2. Generierung aller Arten von Testflügen und Szenarien,
3. Vergleich mit exakter Ground-Truth möglich.

Der erste Punkt versteht sich als trivial und braucht nicht weiter ausgeführt werden. Der zweite Punkt ist insofern interessant, als das auch Flugversuche möglich sind, die normalerweise in der Realität so nicht durchführbar sind – sei es wegen Auflagen, der Gefährdung von Personen oder aufgrund von Sicherheitsbestimmungen. Es können so zum Beispiel Flüge in urbanen Gebieten simuliert, aggressivere Manöver geflogen und beliebige Kamerawinkel oder Montierungen evaluiert werden.

Der letzte Punkt der Auflistung stellt einen sehr wichtigen Aspekt dar. Der Versuchsträger kann im realen Flugversuch keine hundertprozentige Ground-Truth liefern. Die an die Navigationslösung gekoppelte „Wahrheit“ kann als einziges zum Vergleich herangezogen werden und unterliegt einer gewissen Messungenauigkeit. Der einzige Nachteil der erstellten Simulationsumgebung besteht darin, dass kein Bewegungsmodell integriert wurde.

¹Small Vision System.

Parameter wie Beschleunigung und Rauschen sind umsetzbar, jedoch nicht die wirkliche Flugdynamik des Versuchsträgers.

Natürlich läuft man Gefahr, dass der Algorithmus zu sehr an die synthetische Testumgebung angepasst ist. Jedoch bietet eine Testumgebung eine gute Möglichkeit, zunächst einmal die grundlegenden Funktionen und die Korrektheit dieses Ansatzes zu testen und zu überprüfen, so dass man mit dieser bekannten Funktionalität sich dann den Testdaten aus realen Flugversuchen nähern kann und um die gegebenenfalls nötigen, weiteren Anpassungen vorzunehmen.

5.1.1 Die Simulationsumgebung

Die Simulationsumgebung wurde mittels Blender realisiert. Blender ist eine frei verfügbare Open Source 3D-Grafik Anwendung, veröffentlicht unter der GNU GPL. Die Anwendung ermöglicht die Erstellung beliebiger 2D- und 3D-Inhalte mit Möglichkeiten zur Modellierung, Texturierung, Beleuchtung, Animation etc. Durch die offene Architektur bietet Blender plattformübergreifende Einsatzmöglichkeiten und Erweiterbarkeit [RS04].

5.1.2 Simulation der Stereokamera

Die verwendete Stereokamera von Videre Design zeichnet, wie unter Kapitel 2.2 beschrieben wird, pro Zeiteinheit ein Intensitätsbild und ein Tiefenbild auf. Um nun einen realen Flugversuch zu ersetzen, müssen diese beiden Ausgabedaten der realen Kamera durch Blender simuliert werden.

Obwohl die reale STOC-Kamera aus zwei baugleichen Einzelkameras besteht, wird in Blender nur eine Szenenkamera zur Simulation der Intensitäts- und Disparitätsbilder benötigt. Intensitätsbilder entsprechen hierbei in der simulierten Welt einer durch *Ray Tracing* (vgl. [Gla89]) erzeugten Abbildung einer 3D-Szene. Die Beleuchtung, Geometrie, Texturierung und weitere Einstellungen sind frei parametrisierbar. Für die Extraktion des Disparitätsbildes aus der virtuellen Welt benötigt man keine Rückrechnung mittels der Epipolargeometrie. Hier genügt die Abfrage der vorhandenen z -Tiefe zur Kamera, worauf auf Seite 53 noch genauer eingegangen wird.

Die relevanten intrinsischen Parameter der STOC-Kamera können weitestgehend in die simulierte Welt übertragen werden. Hierzu zählen beispielsweise der Öffnungswinkel und das Pixel-Seitenverhältnis. Die restlichen Werte wie radiale Verzerrung, Verschiebung des Bildkoordinaten-Systems und Verschiebung der Sensor-Achsen werden als ideal angenommen und nicht weiter betrachtet.

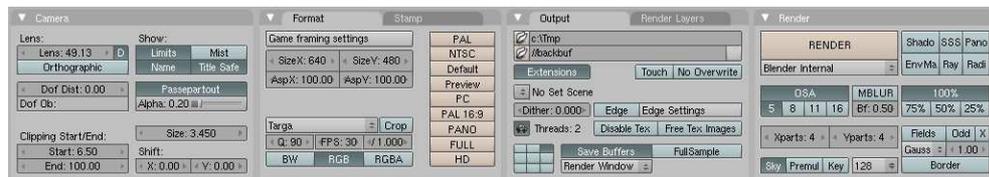


Abbildung 5.1: Kamera-Parameter und Render-Einstellungen der Simulationsumgebung.

Die wichtigsten Kamera-Einstellungen der Simulation werden in Abbildung 5.1 gezeigt. Im linken Bereich sieht man den Öffnungswinkel von $49,13^\circ$. Im mittleren Bereich wird die Auflösung der virtuellen Kamera inklusive des Pixel-Seitenverhältnisses auf 1:1 gesetzt. Da die reale Stereo-Kamera durch die Basislinie in ihrem messbaren Tiefenbereich limitiert wird (siehe 2.3), wurde in der virtuellen Kamera zusätzlich ein eingeschränkter Nahbereich (near clipping) von 6,5 m eingestellt. Dies entspricht einer maximalen Disparität von 32 Pixeln.

Die durch Blender ermittelten Tiefenwerte sind nicht a priori zur Nutzung in einem Disparitätsbild geeignet. Weder stimmt das Tiefenauflosungsverhalten mit der echten Kamera überein, noch korrespondieren die Maßeinheiten zwischen der realen und der simulierten Welt.

Um ein Tiefenbild zu erstellen, welches die gleiche Charactersitik wie die in dieser Arbeit verwendete Stereokamera aufweist, muss der Ausgabe-Tiefenwert aus Blender zusätzlich transformiert werden. Diese Transformation wird durch eine Reihenfolge von Operationen im Szenen-Knoten-Editor (siehe Abbildung 5.2) realisiert. Der Super-Knoten der Simulationswelt besteht zunächst aus einem Render-Ebenen-Knoten. Von hier aus können die Ausgabe-Schnittstellen, wie etwa die Intensitätsdaten und die z -Werte, weitergeleitet und modifiziert werden.

Im Falle des Intensitätsbildes bedarf es keiner weiteren Modifikation. Dieses Ausgabe-

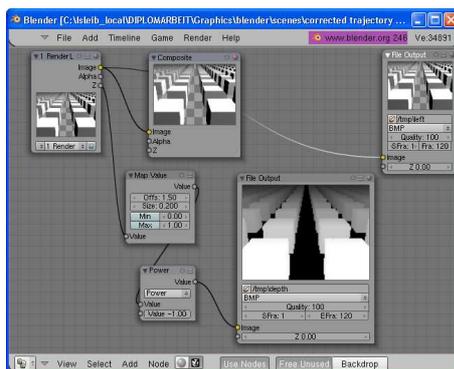


Abbildung 5.2: Die Verarbeitungsschritte in der Blender-Knotenansicht.

datum kann direkt an einen Ausgabeknoten weitergereicht werden, der dafür sorgt, dass dieses Bild im DIP-kompatiblen .bmp-Format gespeichert wird.

Bezüglich des Disparitätsbildes sind zwei Zwischenschritte vor der Ausgabe als Datei notwendig. Als erster Schritt wird der z -Wert in einen Knoten geleitet, der eine Neuordnung des z -Wertebereiches ermöglicht. Dies entspricht der Herstellung der Korrespondenz in den Maßeinheiten zwischen der virtuellen und der realen Welt. Da die echte Stereokamera eine nicht lineare Tiefenwertskalierung liefert, besteht der folgende zweite Schritt darin, die Linearisierung der Tiefe aus der Simulationswelt umzukehren. Dieses wird erreicht durch den reziproken Wert jedes z -Wertes. Zusammenfassend wird jeder z -Wert anhand Formel 5.1 verändert:

$$z_{\text{out}} = (z_{\text{in}} \cdot 0,1565)^{-1} \quad (5.1)$$

Der in Blender auftretende z -Wert z_{in} wird mit einem experimentell ermittelten Faktor von 0,1565 multipliziert und ergibt als Kehrwert den richtig skalierten Tiefenwert z_{out} . Diese Transformation sorgt für eine Übereinstimmung der Maßeinheiten: Eine Einheit in der Simulationswelt entspricht einem Meter in der Realität. Die Korrektheit der Transformation wurde überprüft, indem zur tatsächlichen Entfernung in Blender der Soll-Tiefenintensitätswert der realen Stereokamera ermittelt wurde und von dem durch Blender gelieferten Tiefenbildes abgezogen wurde. Das Ergebnis bestätigt keine Abweichung der

Intensität.² Abbildung 5.3 zeigt die verwendete Testszene.

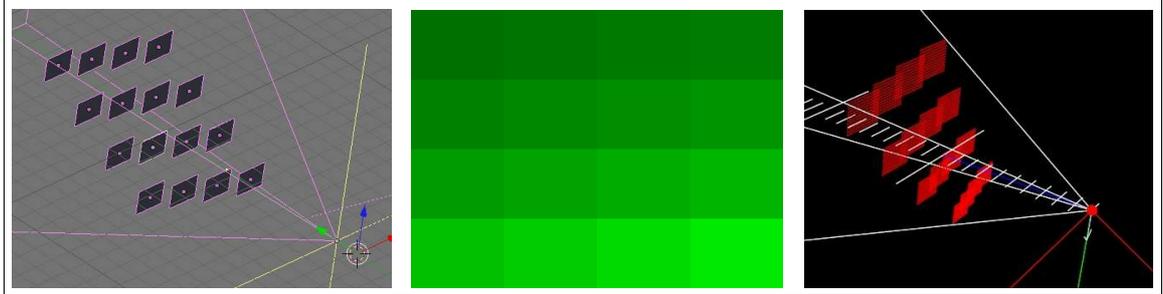


Abbildung 5.3: Transformation einer Testszene der Simulationswelt (links) in das zugehörige Tiefenbild (mitte) und dem Resultat (rechts) als Punktmenge.

5.1.3 Erstellung der Testdaten

Um möglichst gute und überprüfbare Testdaten zu erhalten, wurde eine virtuelle Welt erstellt, die aus äquidistant platzierten Quadern besteht. Sie bietet Informationen in allen räumlichen Koordinaten und eine gute Grundlage für die 3D-Registrierung der Punktmen- gen. Abbildung 5.4 zeigt zwei Ansichten dieser virtuellen Welt innerhalb von Blender.

Das linke Bild repräsentiert hierbei den Blick durch die Kamera der Szene, wobei das rechte Bild eine xy -Draufsicht zeigt, in der auch eine Trajektorie der Kamera zu sehen ist. Da die Trajektorien frei im Raum definierbar sind, werden alle erdenklichen Flugpfade realisierbar. Hinzu kommt, dass die Kameraausrichtung am simulierten Helikopter beliebig wählbar ist, so dass mit verschiedenen Kamerawinkeln experimentiert werden kann und unter anderem Seitwärtsflüge möglich sind. Des Weiteren sind auch Anpassungen in der Geschwindigkeit des Fluges realisierbar.

Damit eine robuste Merkmalsverfolgung in den Intensitätsbildern möglich wird, wurden alle Würfel mit einer Textur überzogen, welche geeignete Eigenschaften zur Merkmalsfindung und -verfolgung aufweist. Die Textur besteht aus 4×4 Feldern, wovon jedes mit einem

²Die Auswertungstabelle ist im Anhang unter 8.2.2 zu finden.

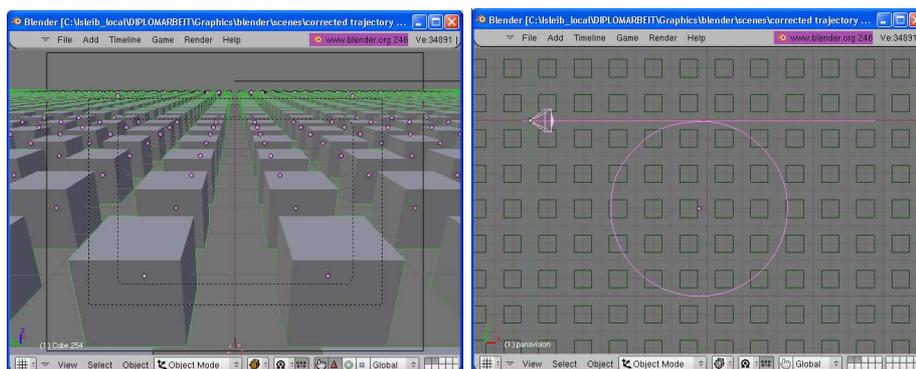


Abbildung 5.4: Kameraansicht der Simulationswelt und xy -Ansicht.

anderen binären Muster (siehe Abbildung 5.5) versehen wurde. Fehlkorrespondenzen können so vermindert werden, da im Gegensatz zu einem simplen Schachbrettmuster nun auch der innere Bereich des Musters pro Würfel eindeutig ist. Das Auftreten des Aperturproblems kann somit besser eingeschränkt werden, was eine robustere Merkmalsverfolgung begünstigt.

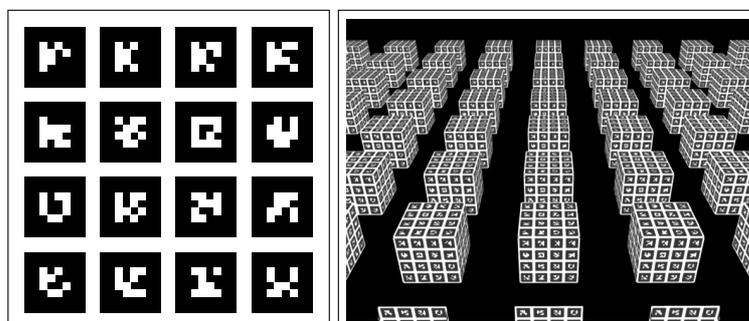


Abbildung 5.5: Marker-Textur für die Intensitäts-Merkmalverfolgung und resultierendes Simulations-Intensitätsbild.

Export der Testdaten zum DIP-Framework

Nach der Erstellung eines virtuellen Flugversuches in Blender können die Ausgabebilder nicht direkt zur Benutzung im DIP weiterverwendet werden. Mehrere Python-Scripte schließen diese Lücke und führen die folgenden Schritte am Ende der Erstellung der Flugsequenz zum Export durch:

Anpassung der Bildnummerierung Die Nummerierung in Blender startet bei 1, wohingegen die Nummerierung im DIP-Framework bei 0 beginnt. Die Benennungskonvention für Disparitätsbilder lautet D####.bmp, für die Intensitätsbilder L####.bmp. (#### entspricht dabei der Nummerierung mit führenden Nullen.)

Konvertierung der Disparitätsbilder Das Ausgabeformat der Disparitätsbilder kann innerhalb von Blender nicht auf ein 8Bit-Graustufenbild eingestellt werden. Mit Hilfe des Moduls PIL (Python Image Library) wird diese Konvertierung innerhalb des Scripts durchgeführt.

Erzeugung von Zeitstempeln Zur Synchronisierung der Intensitäts- und der Disparitätsbilder nutzt das DIP-Framework ein Zeitstempelverfahren. Diese Zeitstempel werden durch das Script in eine jeweils eigene Textdatei für die Disparitäts- und Intensitätsbilder geschrieben und entsprechen einer zeitlichen Auflösung von 30Hz.

Extraktion der Referenztrajektorie Um einen Vergleich zwischen der durch die bildbasierte Bewegungsschätzung berechneten Trajektorie und der tatsächlichen simulierten Trajektorie durchzuführen, exportiert das Script die Orientierungsdaten der simulierten Kamera in eine Textdatei. Diese Daten werden als *Ground-Truth* bezeichnet und ermöglichen einen absoluten Vergleich. In einem realen Flugversuch mit den Versuchsträgern ist momentan nur das schwächere Pendant zur Ground-Truth – die Trajektorie der Navigationslösung – nutzbar.³

³In [AKF99] wurde ein Versuchsaufbau benutzt, der auch mit einem realen Helikopter eine absolute Referenzmessungen mit sechs Freiheitsgraden erlaubt.

5.2 Versuchsansätze

Es existieren insgesamt zwei Versuchsansätze. Der erste Ansatz ist die synthetische Generierung von Kamerafahrten mit der Simulationsumgebung. Dieser Ansatz berücksichtigt rein die theoretischen Eigenschaften der Stereokamera. Faktoren wie etwa die Vibration des Versuchsträgers oder das Rauschverhalten der Stereokamera werden nicht simuliert. Grundgedanke dieses Ansatzes ist die Überprüfung der Funktionalität des Verfahrens. Der zweite Ansatz zeigt, ob sich das Verfahren auch auf reale Daten anwenden lässt.

5.2.1 Synthetische Kamerafahrten

Insgesamt wurden vier Kamerafahrten generiert. Die ersten drei finden in einer Ebene statt, die letzte beinhaltet zusätzlich eine Höhenänderung. Die folgende Übersicht stellt diese im Einzelnen vor:

Quadrat Dieser Flugversuch wird mit einer im Lot montierten Kamera durchgeführt und beschreibt ein Quadrat von 10m Kantenlänge. Die Kamera führt an drei Ecken jeweils eine Kursänderung um $+90^\circ$ durch.

Schleife Die Schleife wird mit einem Kamerawinkel von 30° geflogen. Eingeleitet mit einem Geradeausflug wird eine 360° -Drehung nach rechts durchgeführt. Anschließend wird wieder ausgeleitet in einen Geradeausflug, der in Verlängerung zur Einleitung liegt.

“Hummelflug“ Der Hummelflug wird ebenfalls mit einem Kamerawinkel von 30° geflogen. Er ist der längste simulierte Flug, beinhaltet mehrere Rechts- als auch Linkskurven und ist geschlossen.

Dynamisch Der letzte Flug simuliert einen echten dynamischen Helikopterflug. Angefangen mit einem Geradeausflug beinhaltet er ein Rollen der Kamera während einer 90° Kurve mit einem anschließenden Steigflug.

Abbildung 5.6 veranschaulicht die zu den vier zuvor genannten synthetischen Kamerafahrten gehörenden Referenztrajektorien.

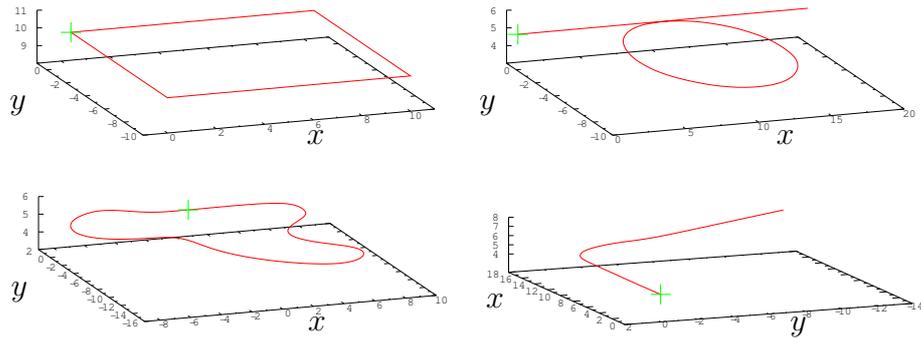


Abbildung 5.6: Die Referenztrajektorien der synthetischen Kamerafahrten. Das Kreuz in den jeweiligen Plots markiert die Startposition. Initialflugrichtung ist entlang der positiven x -Achse. Man beachte die getauschten Achsen im unteren, rechten Plot.

5.2.2 Reale Kamerafahrten

Die realen Kamerafahrten wurden nach den unter Abbildungen 5.7 und 5.8 gezeigten Konfigurationen durchgeführt.

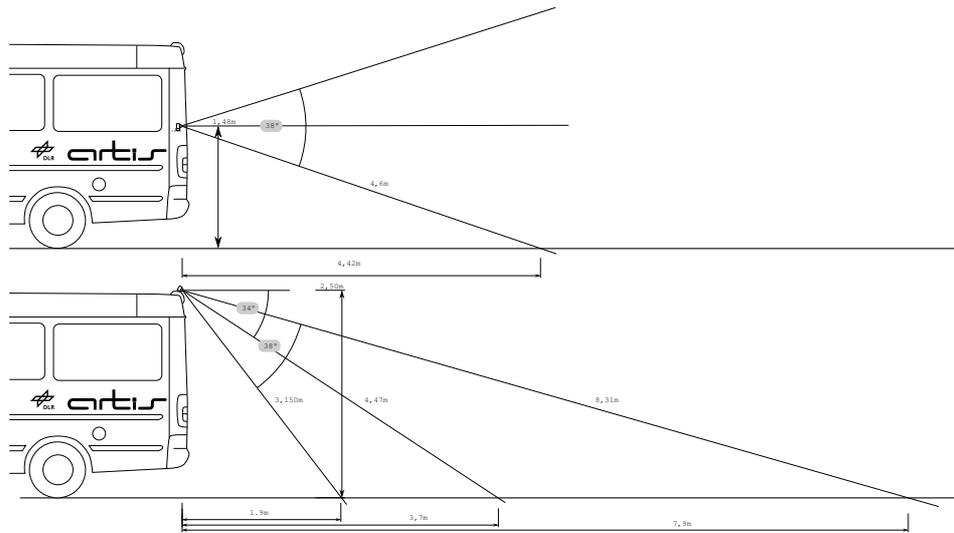


Abbildung 5.7: Versuchskonfigurationen für die realen Kamerafahrten (schematisch).

Wie man in den genannten Abbildungen sehen kann, wurde mit zwei verschiedenen Kamerawinkeln experimentiert. In der ersten Versuchsreihe wurde die Kamera auf der Heck-



Abbildung 5.8: Die zwei Montierungen der Stereokamera.

ablage des ARTIS-Bodenstationsfahrzeugs montiert, um eine Kameraausrichtung genau in Fahrtrichtung zu simulieren. Die zweite Versuchsreihe sieht vor, Bilddaten mit einem Kamerawinkel von ca. 34° zu erzeugen. Hierzu wurde die Kamera auf dem Rücklicht des Fahrzeugdaches angebracht. Da die Kamera in beiden Versuchsreihen nach hinten ausgerichtet war, diente eine Invertierung der generierten Bildsequenzen dazu, die Kamerafahrten in Fahrtrichtung zu erhalten.

Die maximale Sequenzlänge von 200 Bildern lässt nur vergleichsweise einfache Kamerafahrten zu. Des Weiteren ist der ARTIS-Bus nicht mit der Avionik des Versuchsträgers ausgestattet und daher wurden nachvollziehbare, gerade Strecken gefahren. Um die Vergleichbarkeit zu ermöglichen, wurden diese Strecken mit einem Maßband vermessen. Um jedoch nicht zu kurze Streckenlängen zu erhalten, sind die Bildsequenzen hauptsächlich mit nur 7,5 Hz aufgenommen worden.

5.3 Durchführung der Schätzung der Eigenbewegung

In Kapitel 3.2 wurden drei verschiedene Verfahren zur Filterung der Korrespondenzen vorgestellt. Diese Verfahren berechnen auf verschiedene Weise die Korrespondenzen zweier Punktmengen und geben die so gefilterten Punktmengen an die Registrierung weiter. Zwei Verfahren – *Smooth Local 3D* und *RANSAC* – nutzen dabei Korrespondenzen aus Merkmalsverfolgung, das Verfahren über die *Mahalanobis-Distanz* jedoch nicht. Diese insgesamt drei Verfahren werden auf die jeweils vier synthetischen und zwei realen Kamerafahrten angewendet.

Für die synthetischen Datensätze wurden folgende Einstellungen für die Merkmalsverfol-

gung verwendet:

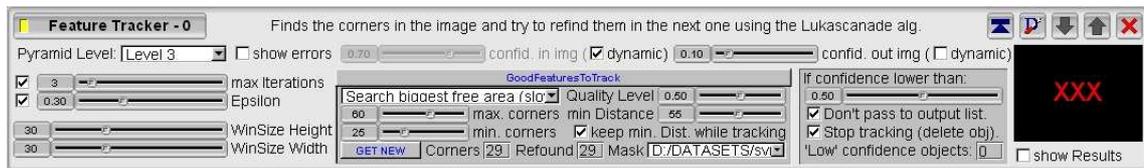


Abbildung 5.9: Parameter der Merkmalsverfolgung für die synthetischen Kamerafahrten.

Zur robusten Schätzung der Eigenbewegung werden mindestens acht Punkte benötigt. Eine Erhöhung der Mindestgrenze für die Merkmalsverfolgung ist zwingend notwendig, da die Anzahl der Merkmale nie konstant bleibt und Ausreißer, oder für das Verfahren schlechte Merkmale, immer dabei sein können. Die Punktmenge wird experimentell so groß gewählt, dass man nach der Filterung durch die verschiedenen Verfahren eine immer noch ausreichend große Punktmenge hat.

Die restlichen Parameter für die Verfahren zur Schätzung der Eigenbewegung sind momentan fest implementiert und werden in Tabelle 5.1 aufgelistet.

Der Wert, der pro Szene jedoch angepasst werden muss, ist der Kamerawinkel α_{cam} , da sonst die Trajektorie in einer zur xy -Ebene rotierten Ebene berechnet wird. In allen Berechnungen der Trajektorie wird ein Kalman-Filter zur Tiefpassfilterung der Rotationen und der Translationen eingesetzt. Die Tests mit der Mahalanobis-Distanz werden zur Beschleunigung des Verfahrens mit aktiviertem Initial-Guess durchgeführt.

Für die realen Datensätze wurden folgende Einstellungen für die Merkmalsverfolgung verwendet:

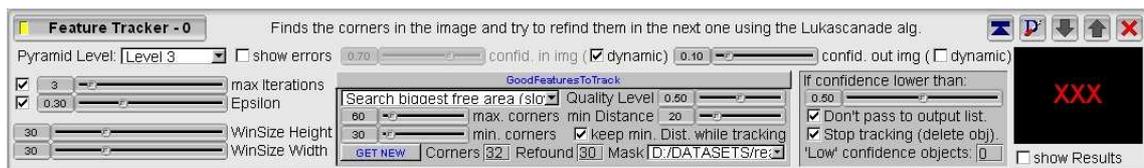


Abbildung 5.10: Parameter der Merkmalsverfolgung für die realen Kamerafahrten.

Alle eingestellten Parameter für sowohl die beiden Merkmalsverfolgungen (siehe Abbildungen 5.9 und 5.10) als auch die in Tabelle 5.1 erwähnten, sind experimentell ermittelt

Parameter	Wert	Erklärung
EPSILON_FINISH	0.0000005	Wenn der mittlere Registrierungsfehler zwischen letzter und aktueller Schätzung geringer als dieser Schwellwert ist, wird der ICP abgebrochen.
MAX_TRANSLATION_ERROR	0.2	Falls die SVD oder der ICP einen zu großen Sprung in der Bewegung berechnet hat, wird dies als Fehler angesehen und \mathbf{R} , \mathbf{t} aus t-1 angenommen.
MAX_REGISTRATION_ERROR	0.2	Falls die SVD oder der ICP einen zu großen Registrierungsfehler hat, wird dies auch als Fehler angesehen und ebenfalls \mathbf{R} , \mathbf{t} aus t-1 angenommen.
MAX_ROTATION_ERROR	$\text{math}::\text{c_PI} / 13$	Falls die SVD oder der ICP einen zu große Sprung in der Rotation ($> 13,846^\circ$) berechnet hat, wird dies auch als Fehler angesehen und \mathbf{R} , \mathbf{t} aus t-1 angenommen.
ITER_WHILE_MMMD_MORE	2	Der <i>mittlere</i> mittlere Bewegungsunterschied wird zu klein.
ITER_WHILE_MMMD_DIFF_MORE	5	Der Unterschied zwischen dem letzten und dem aktuellen <i>mittleren</i> mittleren Bewegungsunterschied wird ebenfalls zu klein.
MIN_MOVES	10	Mindestanzahl der Korrespondenzen, die an die SVD durch den <i>Smooth Local 3D</i> weitergereicht werden.
MAHALANOBIS_SCALE	0.2	Wenn der Entfernungsunterschied zweier Merkmale einer Korrespondenz über diesem Kovarianzskalierungsfaktor liegt, wird die Korrespondenz verworfen.
MAX_NUM_RANSAC_RUNS	foundCorrs · 2	Maximale Anzahl der durch RANSAC versuchten Ermittlung der besten Konsensus-Menge. Es werden die Anzahl der aus der Merkmalsverfolgung stammenden Korrespondenzen mal zwei genommen.

Tabelle 5.1: Allgemeine Parameter zur Durchführung der Experimente.

worden. Im Fall der Merkmalsverfolgung bedeutet dies, dass zum einen eine Statistik über die Merkmale ausgewertet wurde und dementsprechend die Parameter je nach Verbesserung oder Verschlechterung der Statistik angepasst wurden. Zum anderen wurden auch über eine direkte visuelle Begutachtung der Merkmalsverfolgung bei Ablauf einer Sequenz Anpassungen vorgenommen. Die so ermittelten Einstellungen wurden dann für die Experimente festgesetzt und verwendet.

Die restlichen Werte aus Tabelle 5.1 sind wie folgt ermittelt worden: *EPSILON_FINISH* wurde so klein gewählt, dass die Anzahl Iterationen möglichst groß wurde und somit das Ergebnis des ICP nicht durch eine eventuell ausgelassene Iteration ungenauer wird. Die Parameter *MAX_TRANSLATION_ERROR*, *MAX_REGISTRATION_ERROR* und *MAX_ROTATION_ERROR* sind so ermittelt worden, dass nur die größten Fehler in der Schät-

zung in der Eigenbewegung gefiltert wurden: Falls eine geschätzte Trajektorie durch eine Fehlschätzung unterbrochen wurde, so sind in den Auswertungsdaten der betreffenden Bewegungsschätzung die verantwortlichen Werte ermittelt und als Grenzwerte festgesetzt worden. Die nachfolgenden drei Parameter für den Smooth Local 3D-Ansatz wurden bis auf *MIN_MOVES* aus [Goo04] übernommen. *MIN_MOVES* wurde auf einen Wert von 10 gesetzt, damit bei vielen Ausreißern die besten 10 Korrespondenzen für die Berechnung zur Verfügung stehen. Der Parameter *MAHALANOBIS_SCALE* ist durch eine rein visuelle Kontrolle der Filterung festgesetzt worden. Die OSG-Ausgabe der Punktmengenansicht zeigt sowohl die ungefilterten als auch die gefilterten Korrespondenzen an. Hierdurch konnte bei Ablauf einer Sequenz und der Betrachtung der ermittelten Korrespondenzen der Parameter so eingestellt werden, dass ein plausibles Filterergebnis erzielt wurde. Der letzte Parameter *MAX_NUM_RANSAC_RUNS* wurde so hoch gewählt, so dass auch eine wiederholte Ausführung der Bewegungsschätzung mit dem RANSAC-Verfahren nicht zu gravierend unterschiedlichen Ergebnissen führt.

5.4 Ergebnisse der synthetischen Kamerafahrten

Dieser und der folgende Abschnitt 5.5 enthalten die Ergebnisse der jeweiligen Versuche. Alle Auswertungen werden dabei nach dem folgenden Schema abgehandelt: Es werden zuerst drei grafische Darstellungen gezeigt. Die linke enthält einen 3D-Plot, in dem die durch das ausgewählte Verfahren berechnete Trajektorie gleichzeitig mit der Referenztrajektorie dargestellt wird. Die mittlere Darstellung ist ähnlich zu der ersten, nur wird hier die Sicht auf die xy -Ebene beschränkt. Sie entspricht also einer Draufsicht auf die Szene. Die rechte und damit letzte Darstellung zeigt den Abweichungsverlauf zur Referenztrajektorie für die jeweiligen Komponenten x, y und z , wobei senkrechte Striche den Zeitpunkt einer markanten Bewegungsänderung markieren, zum Beispiel den Beginn oder das Ende einer Kurve oder einen Kurvenwechsel etc. Als Abschluß auf die grafischen Darstellungen folgt jeweils eine kurze Beschreibung.

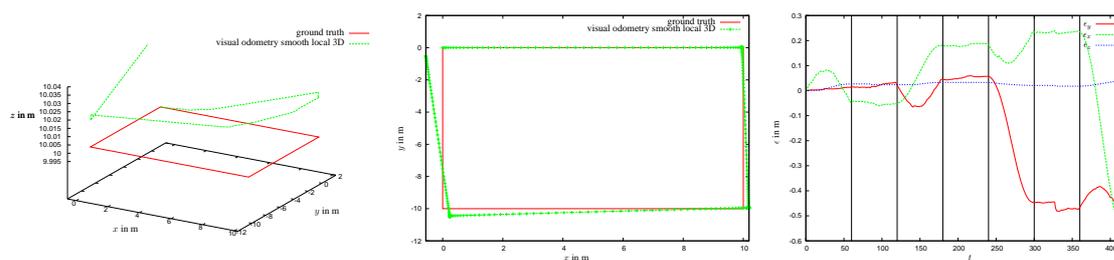


Abbildung 5.11: Kamerafahrt "Quadrat" mit Filterverfahren: "Smooth Local 3D".

5.4.1 Quadrat

Zu *Abbildung 5.11 (Smooth Local 3D)*: Im mittleren Bild ist zu erkennen, dass die Referenztrajektorie relativ gut verfolgt wird. Die Länge der geraden Segmente sind äußerst genau rekonstruiert worden, wobei es an jeder Ecke, wo nur eine Drehung um die z -Achse stattfindet, zu einem Fehler in der Gesamtrotation von eigentlich $+90^\circ$ kommt. Dies bedeutet, dass das Rechteck pro Ecke immer mehr geöffnet wird und die Ausgangsposition nicht mehr erreicht wird. Die Höhenabweichung z ist minimal (weniger als 4 cm), wobei die die Abweichung in xy immer mehr zunimmt, bis zu einem Fehler von knapp 60 cm.

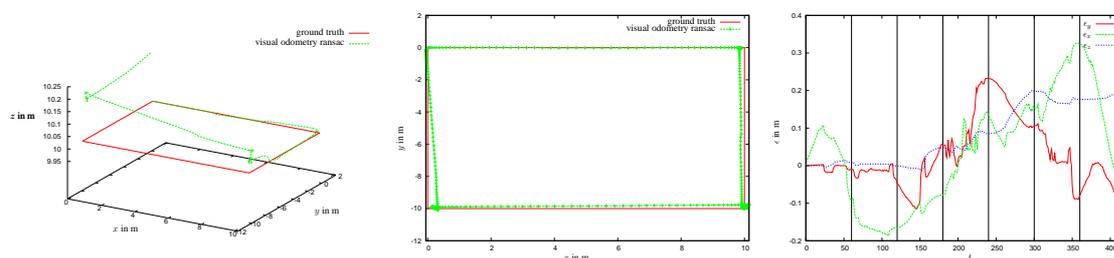


Abbildung 5.12: Kamerafahrt "Quadrat" mit Filterverfahren: "RANSAC".

Zu *Abbildung 5.12 (RANSAC)*: Hier fällt direkt auf, dass sich die berechnete Trajektorie zunächst besser an der Referenz hält, jedoch die Abweichung in der Höhe immer mehr zunimmt (maximal 20 cm). Alle Achsen sind eindeutig mehr verrauscht. Die ersten beiden Rotationen an den jeweiligen Ecken werden sehr gut geschätzt, lediglich die letzte Rotation erfährt eine Abweichung, so dass sich das Rechteck wieder öffnet. Auffallend ist auch der Translationsanteil, der bei den Rotationen mit auftritt. Die Stabilität entlang der

Geraden ist sehr gut, jedoch wird die Länge falsch geschätzt. Auf einer geraden Strecke von 10 m werden nur ca. 97% bis 99% dieser Strecke erreicht.

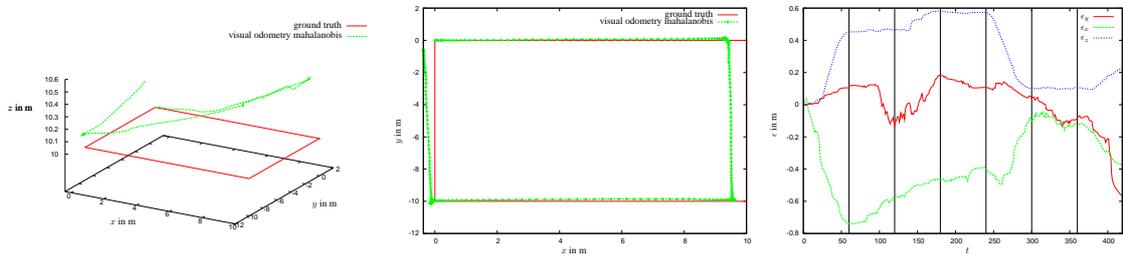


Abbildung 5.13: Kamerafahrt "Quadrat" mit Filterverfahren: "Mahalanobis".

Zu *Abbildung 5.13 (Mahalanobis)*: Auch hier ergibt sich ein ähnliches Bild: Die Rotationen in den ersten beiden Ecken werden gut geschätzt, die Länge der geraden Segmente wird etwas unterschätzt. Die dritte Ecke öffnet das Rechteck wieder. An den Fehlerkurven sieht man eindeutig, dass schon gegen Anfang des Kamerafluges die x - und z -Fehler sehr schnell sehr groß werden.

5.4.2 Schleife

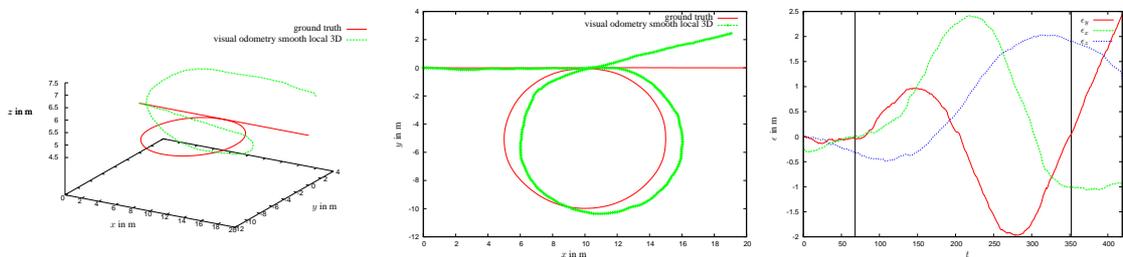


Abbildung 5.14: Kamerafahrt "Schleife" mit Filterverfahren: "Smooth Local 3D".

Zu *Abbildung 5.14 (Smooth Local 3D)*: Die Abweichung in der Höhe und der Rotationsfehler sind hier prägend. Wird zunächst die Anfangsgerade gut geschätzt, so wird die Kurve verspätet eingeleitet und verfrüht wieder ausgeleitet, so dass keine kompletten 360° geschätzt wurden. Die Abweichung in der Höhe nimmt hauptsächlich im Verlauf der Kur-

ve zu und beträgt maximal ca. 2 m. Der Kurvendurchmesser ist jedoch relativ gut gehalten worden.

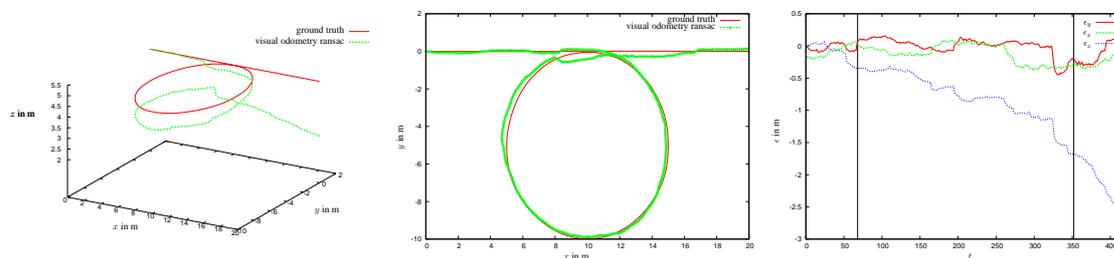


Abbildung 5.15: Kamerafahrt "Schleife" mit Filterverfahren: "RANSAC".

Zu *Abbildung 5.15 (RANSAC)*: Hier wird zunächst die Höhe und die Richtung gut gehalten. Ab dem Start der Kurve fällt auch hier ein kontinuierlicher Fehler in der Höhenschätzung auf, der jedoch viel systematischer scheint. Auf 100 Einzelbilder werden ca. 70 cm an Höhenfehler generiert. Die Kurve wird sehr gut eingeleitet und verfolgt, lediglich gegen Ende kommt es zu einer Abweichung, so dass auch hier nicht ganze 360° erreicht wurden. Der maximale Höhenfehler beträgt hier ca. 2,6 m.

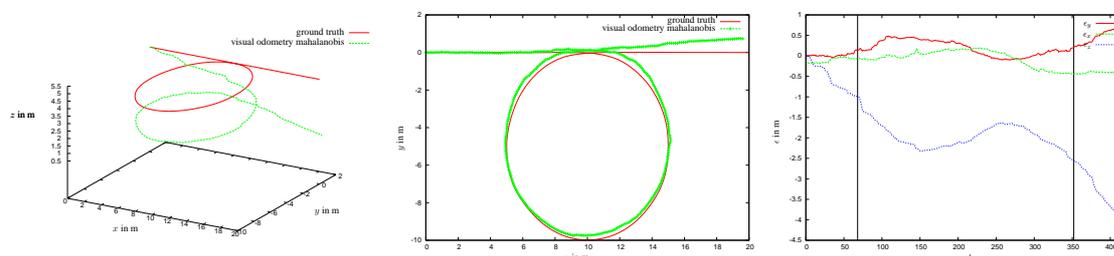


Abbildung 5.16: Kamerafahrt "Schleife" mit Filterverfahren: "Mahalanobis".

Zu *Abbildung 5.16 (Mahalanobis)*: Ähnlich sieht es bei dieser berechneten Trajektorie aus, jedoch beginnt der Höhenfehler schon von Anfang an zuzunehmen und erreicht einen Maximalwert von fast 4 m. Richtung und Kurve werden jedoch, bis auf die falsche Ausleitung der 360° -Kurve, gut gehalten.

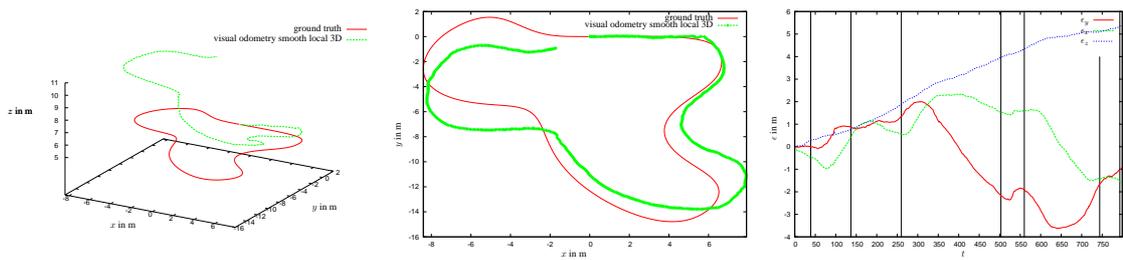


Abbildung 5.17: Kamerafahrt “Hummelflug“ mit Filterverfahren: “Smooth Local 3D“.

5.4.3 Hummelflug

Zu *Abbildung 5.17 (Smooth Local 3D)*: Der Verlauf der berechneten Trajektorie wird hauptsächlich durch die Unterschätzung der Rotationen bestimmt. Schon bei der ersten Kurve ergibt sich eine kleine Abweichung, die dann bei den weiteren Kurven immer mehr zunimmt. Die xy -Endposition weicht gegen Ende zwar nur um ca. 1,5 m ab, jedoch zeigt, durch den stetigen Anstieg des Höhenfehlers, die z -Position eine Abweichung von mehr als 5 m an.

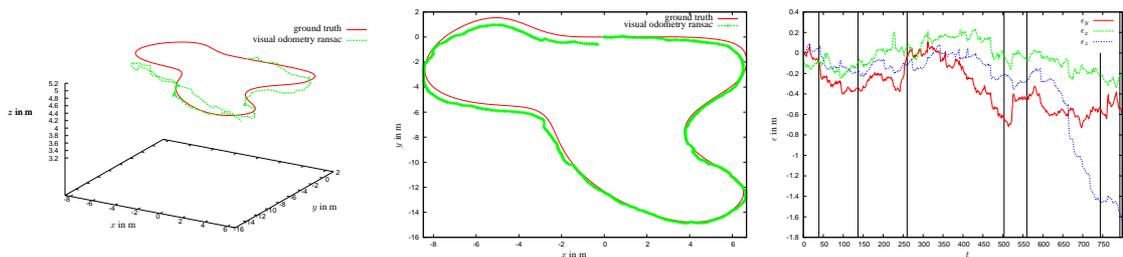


Abbildung 5.18: Kamerafahrt “Hummelflug“ mit Filterverfahren: “RANSAC“.

Zu *Abbildung 5.18 (RANSAC)*: Hier ergibt sich ein besseres Bild: Der berechnete Kurvenflug hält sich sehr nah an der Referenztrajektorie und schließt gut ab. Die Maximalabweichung in der xy -Ebene schwankt im Intervall von $[-0,7\text{ m}, 0,2\text{ m}]$. Nur der Höhenfehler nimmt auch hier zu, jedoch erst im letzten Viertel des Fluges und beträgt gegen Ende ca. 1,6 m.

Zu *Abbildung 5.19 (Mahalanobis)*: Die berechnete Trajektorie erreicht hier per Zufall, zumindest in der xy -Ebene, eine fast geschlossene Flugbahn. In der Höhe wird ein kon-

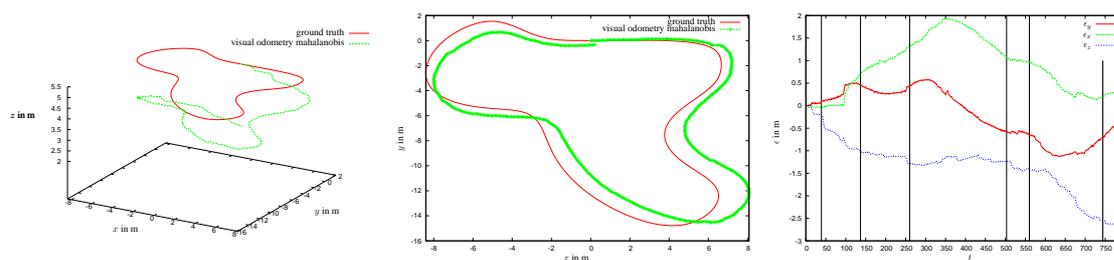


Abbildung 5.19: Kamerafahrt ‘Hummelflug‘ mit Filterverfahren: ‘Mahalanobis‘.

tinuierlicher Fehler gemacht, der sich auf fast 3 m akkumuliert. Zu Zeit $t = 100$ findet eine sehr falsche Schätzung der Rotation statt, so dass ab hier die berechnete Trajektorie beginnt, abzuweichen.

5.4.4 Dynamisch

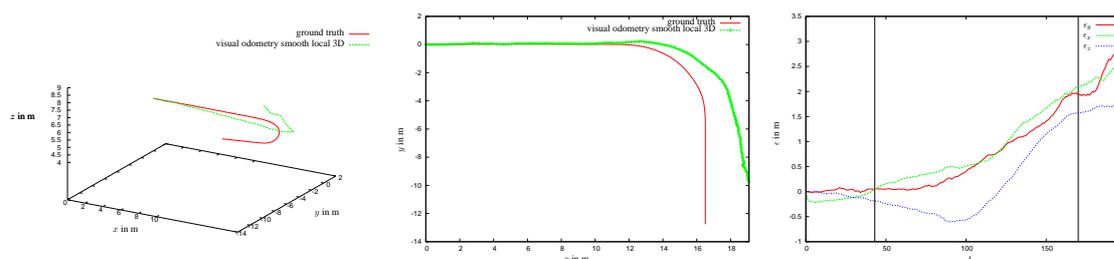


Abbildung 5.20: Kamerafahrt ‘Dynamisch‘ mit Filterverfahren: ‘Smooth Local 3D‘.

Zu *Abbildung 5.20 (Smooth Local 3D)*: Dieser Flug wird zunächst gut begonnen. Bei der Einleitung des Kurvenfluges wird jedoch eine Rotationsschätzung in die entgegengesetzte Richtung durchgeführt, später aber in die richtige Richtung korrigiert. Die Gesamtheit der Fehler beginnt bei Einleitung der Kurve und steigt immer mehr an, bis gegen Ende die Abweichung letztendlich im Mittel ca. 2,4 m beträgt.

Zu *Abbildung 5.21 (RANSAC)*: Die Referenztrajektorie wird hier zunächst in der xy -Ebene gut geschätzt, jedoch zeichnet sich der Start der Zunahme der Abweichung deutlich ab $t = 100$ ab. Angefangen mit der Zunahme des Höhenfehlers, folgt die Abweichung in y . Der Fehler in der x -Komponente beginnt als letztes. Insgesamt ist der Höhenfehler am

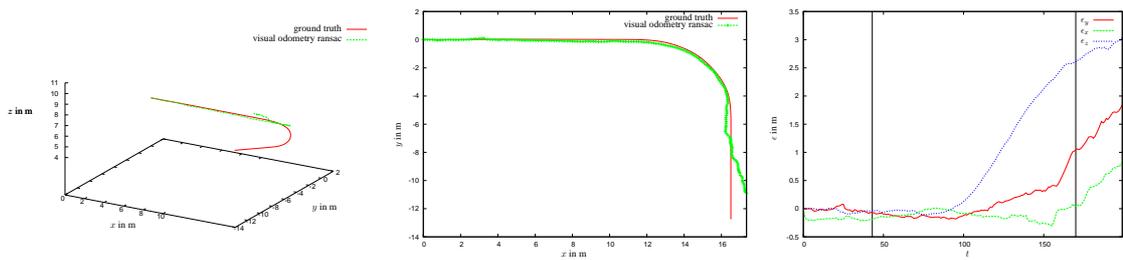


Abbildung 5.21: Kamerafahrt "Dynamisch" mit Filterverfahren: "RANSAC".

Ende mit ca. 3 m am stärksten ausgeprägt, wobei der Fehler von y ca. 1,5 m und von x ca. 0,75 m beträgt.

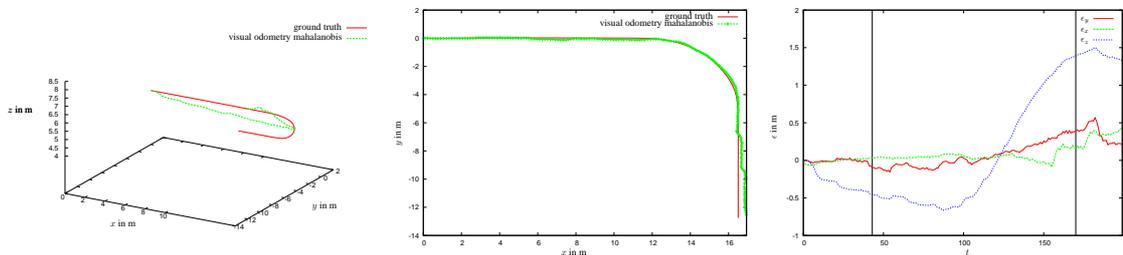


Abbildung 5.22: Kamerafahrt "Dynamisch" mit Filterverfahren: "Mahalanobis".

Zu *Abbildung 5.22 (Mahalanobis)*: Der Fehlerkurvenverlauf und die Draufsicht zeigen hier eindeutig ein relativ stabiles Verhalten in der xy -Ebene. Die Abweichung beträgt hier bei weniger als 0,5 m. Insgesamt wird die Referenztrajektorie gut verfolgt, jedoch gibt es hier eine Komponente, die hauptsächlich falsch geschätzt wird. Die z -Komponente verläuft zunächst im negativen Fehlerbereich bis ca. 0,5 m und steigt dann später auf den Maximalwert von fast 1,5 m.

5.5 Ergebnisse der realen Kamerafahrten

In diesem Abschnitt werden die Ergebnisse der realen Kamerafahrten gezeigt. Hierzu muss vorweg gesagt werden, dass alle drei Verfahren Schwierigkeiten hatten, die Trajektorie zu rekonstruieren. Der Ansatz über RANSAC hat sogar total versagt. Eine genaue Analyse

der Ergebnisse und der Fehlfunktion wird dazu Kapitel 6 durchgeführt.

Da keine echte Referenztrajektorie zu den realen Kamerafahrten vorhanden ist, wurde eine exakte Geradeausfahrt mit homogener Geschwindigkeit für diesen Versuch angenommen. Ein Vergleich über die x -Komponente, welche der Fahrtrichtung und somit auch der Geschwindigkeit entspricht, ist also nur eingeschränkt möglich. Die Länge der Versuchsfahrt von ca. 63 m ist durch eine Messung per Maßband bestimmt worden. Die Sicht der Kamera ist genau in Fahrtrichtung, also 0° .

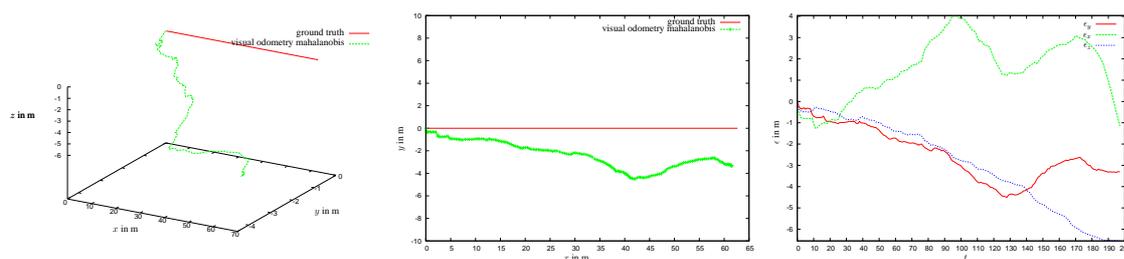


Abbildung 5.23: Reale Kamerafahrt mit Filterverfahren: “Mahalanobis“.

Zu *Abbildung 5.23 (Mahalanobis)*: In der xy -Ansicht ist eindeutig eine Drift nach rechts feststellbar. Diese nimmt in den ersten 5 m sprunghaft bis auf 1 m in der y -Komponente zu und verläuft dann weicher bis zu einem maximalen y -Fehler von etwas über 4 m bei einem x von ca. 42 m. Danach nähert sich die berechnete Trajektorie wieder etwas an die Referenz an und beendet mit einem Fehler in y von ca. 3,5 m. Die Gesamtlänge der Messfahrt wurde gut geschätzt. Betrachtet man den Fehler in z , so nimmt dieser fast linear zu. Insgesamt beträgt der Fehler in der Höhenschätzung am Ende mehr als 6 m.

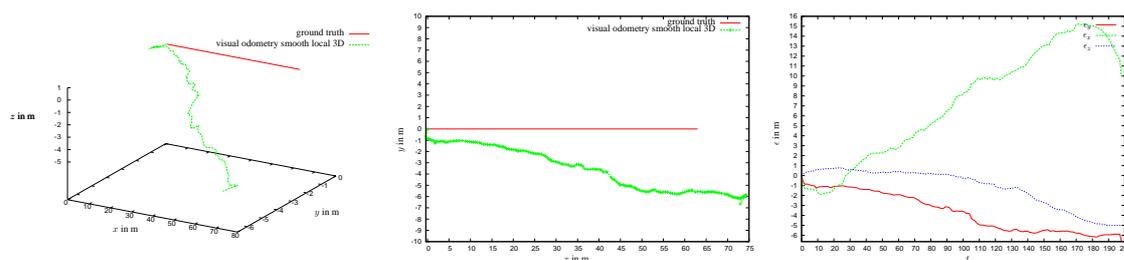


Abbildung 5.24: Reale Kamerafahrt mit Filterverfahren: “Smooth Local 3D“.

Zu *Abbildung 5.24 (Smooth Local 3D)*: Auch hier ist eindeutig eine Drift nach rechts

feststellbar, wobei ein extremer Anstieg des y -Fehlers zu knapp 1 m schon zwischen den ersten drei Bildern stattfindet. Danach verläuft die Drift stetig weiter nach rechts, bis zu einem Gesamtfehler von ca. 6 m. Insgesamt wird eine schnellere Bewegung entlang der x -Achse geschätzt, so dass die Länge der Messfahrt um ca. 12 m zu hoch ausfällt. Der Verlauf des Höhenfehlers ist bis zur Mitte der Messungen gut, erfährt dann aber eine Zunahme, so dass dieser letztendlich bei 5 m liegt.

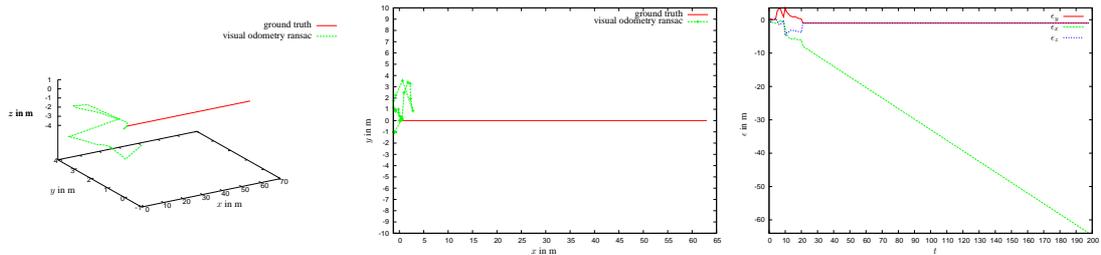


Abbildung 5.25: Reale Kamerafahrt mit Filterverfahren: “RANSAC“.

Zu *Abbildung 5.25 (RANSAC)*: Hier sieht man das komplette Versagen der Schätzung der Eigenbewegung. Die ersten 20 Messpunkte streuen sich um den Start der Referenztrajektorie, wobei danach keine Trajektorie mehr zustande kommt.

Die zweite reale Kamerafahrt, welche mit einem Kamerawinkel von 33° durchgeführt wurde, lieferte keine zufriedenstellenden Bilddaten. Die Intensitätsdaten waren im Nachhinein nicht zur robusten Schätzung der Eigenbewegung geeignet, so dass keine Trajektorien durch die einzelnen Verfahren generiert werden konnten. Die Gründe hierfür und für das Versagen des RANSAC werden im kommenden Kapitel analysiert.

Kapitel 6

Evaluation der Ergebnisse

Dieses Kapitel dient der Evaluation der unter Kapitel 5.4 und 5.5 gezeigten Ergebnisse. Es werden zuerst die synthetischen, danach die realen Kamerafahrten evaluiert. Darauf folgt eine Gesamtbewertung. Abschließend wird im Abschnitt 6.4 ein Feldversuch präsentiert, der die Entfernungsmessungen der verwendeten Stereokamera mit einem Lasermesssystem vergleicht.

6.1 Evaluation der synthetischen Kamerafahrten

Die Genauigkeit der geschätzten Eigenbewegung fällt je nach Filterungs-Verfahren sehr unterschiedlich aus, da die jeweiligen Ansätze eine sehr unterschiedliche Herangehensweise bieten. Man kann jedoch bei genauer Betrachtung die Aussage machen, dass unter Verwendung des *RANSAC* die insgesamt besten Ergebnisse erzielt wurden. Das Verfahren *Smooth Local 3D* neigt immer dazu, die Rotationen in den Bilddaten nur verzögert zu erkennen, so dass immer zu wenig von der tatsächlichen Kurve rekonstruiert wird. Der Ansatz über die *Mahalanobis-Distanz* liefert etwas schlechtere Ergebnisse als der *RANSAC*, jedoch ist die Kurvenerkennung ähnlich reaktiv. Es scheint, als ob der *Smooth Local 3D* durch die Analyse der Bewegungsvektoren die Rotationen zu sehr ausglättet. Bezüglich der Translationen kann festgestellt werden, dass alle drei Verfahren hier ähnlich gute Ergebnisse liefern.

Ein genereller Fehler besteht in der Fehlschätzung der Höhe. Alle Verfahren sind in der xy -Ebene stabiler als in der Höhen-Komponente. Gerade bei der Kombination von Translation und Rotation, sei es auch nur in der Ebene, wird eine Höhenänderung generiert. Dieser systematische Fehler kann vermutlich auf die diskretisierten Tiefenwerte zurückgeführt werden. Der ersten Versuch *Quadrat* wurde mit einer exakt nach unten ausgerichteten Kamera durchgeführt. Man kann hier eindeutig feststellen, dass der Höhenfehler minimal ausfällt. Der mögliche Fehler in der Tiefenschätzung wird durch die Ausrichtung der Kamera also auf einen kleinen Disparitätsbereich eingeschränkt. Die nächsten Versuche wurden jedoch mit einem Kamerawinkel von 30° durchgeführt, so dass erstmals die Diskretisierung zum Tragen kommen kann. Diese Diskretisierung kann bei solchen Kamerawinkeln dafür sorgen, dass eine Bodenebene mit zunehmender Entfernung ein ungewünschtes Gefälle bekommt und somit fälschlicherweise ein Ansteigen oder Abfallen der Trajektorie bewirkt.

6.2 Evaluation der realen Kamerafahrten

Insgesamt zeigt sich hier gegenüber den synthetischen Kamerafahrten eine Verschlechterung der Schätzung der Eigenbewegung. Aus den Bilddaten der Kamerafahrt mit Sicht in Fahrtrichtung kann nur von den Verfahren *Mahalanobis-Distanz* und *Smooth Local 3D* eine Trajektorie generiert werden. Der Grund für das Versagen des *RANSAC*-Ansatzes muss noch genauer untersucht werden. Eventuell war die Untergrenze für die Anzahl der Versuche des *RANSAC* zu optimistisch gewählt. Die Robustheit der Merkmalsverfolgung sowie die verrauschten Tiefendaten führen zu Punktkorrespondenzen, die mehr Ausreißer beinhalten, so dass die Anzahl der Versuche nicht mehr ausreichen.

Die beiden Trajektorien in Abbildungen 5.23 und 5.24, die geschätzt werden konnten, zeigen ähnliche Ergebnisse: in beiden Fällen wird von der Referenztrajektorie insgesamt ca. 6 m nach rechts abgewichen. Zum einen ist hierfür teilweise die Drift verantwortlich, zum anderen kann auch eine nicht exakte Ausrichtung der Kamera hierzu geführt haben, da beide Verfahren ein ähnliches Verhalten aufweisen. Das implementierte Verfahren geht von einer Montierung der Kamera aus, die *nur* einen Freiheitsgrad – den Kamerawinkel α_{cam} – berücksichtigt. Ist die räumliche Kalibrierung nicht korrekt, so würde dies, wie bei

der Angabe eines falschen Kamerawinkels, zu einer falschen Orientierung der geschätzten Trajektorie führen. Höhenfehler sind bei beiden Verfahren zu finden, wobei der Ansatz über *Smooth Local 3D* mit einer Abweichung von 5 m besser ausfällt, als der Ansatz über die *Mahalanobis-Distanz*, die bei fast 7 m liegt.

Der größte Unterschied in den beiden Verfahren liegt jedoch in der Genauigkeit der Schätzung der Gesamtlänge der Referenztrajektorie. Der Ansatz über die *Mahalanobis-Distanz* schafft es nahezu, die Gesamtlänge von ca. 63 m zu schätzen, wohingegen der Ansatz über *Smooth Local 3D* die Trajektorie bei fast 75 m enden lässt. Es scheint, dass die tiefenabhängige Filterung der *Mahalanobis-Distanz* Wirkung zeigt.

Die Bilddaten der zweiten Versuchsfahrt konnten leider nicht zur Bewegungsschätzung verwendet werden. Die Merkmalsverfolgung hatte extreme Schwierigkeiten, Merkmale zu verfolgen. Dieses Problem ist hauptsächlich auf die starken Änderungen in der Belichtung der Bilddaten zurückzuführen. Abbildung 6.1 zeigt hierzu die wichtigsten Daten zur Fehlerauswertung.

Aus den beiden oberen Teilabbildungen geht eindeutig hervor, dass der Registrierungsfehler – dieser wird durch den obersten Funktionsgraphen dargestellt – mehrmals ein eklatantes Ausmaß erreicht. Normalerweise bewegt sich der Registrierungsfehler im einstelligen Zentimeterbereich, jedoch werden hier mehrmals Registrierungsfehler von ca. 10 m erreicht. Jeder dieser Spitzenwerte führt zu einem großen Versatz in sowohl der Berechnung der Translation als auch der Rotation und bedingt gravierende Sprünge in der rekonstruierten Trajektorie.¹

Der Grund hierfür ist in den beiden unteren Teilabbildung zu erkennen. Links wird eine Statistik der Merkmalsverfolgung gezeigt. Jedes Merkmal hat eine Lebenszeit, bestimmt durch den Zeitpunkt seiner Generierung und der Anzahl seiner erfolgreichen Verfolgungen. Häufig kommen Lebenszeiten zustande, die einen Wert nahe der 20 oder größer haben. Die mittlere Lebenszeit der Merkmale sollte aber aufgrund der Geschwindigkeit der Kamerafahrt jedoch bei ca. 10 Bildern liegen. Der Grund für diese unerwartet hohe Lebenszeit ist im rechten Bild (Intensitätsbild mit überlagerten Merkmalen) zu sehen: der unteren Bildbereich enthält Merkmale, die einen Vektor entgegen der eigentlichen Be-

¹Diese Sprünge sind auch in Abbildung 5.25 auf Seite 70 zu sehen.

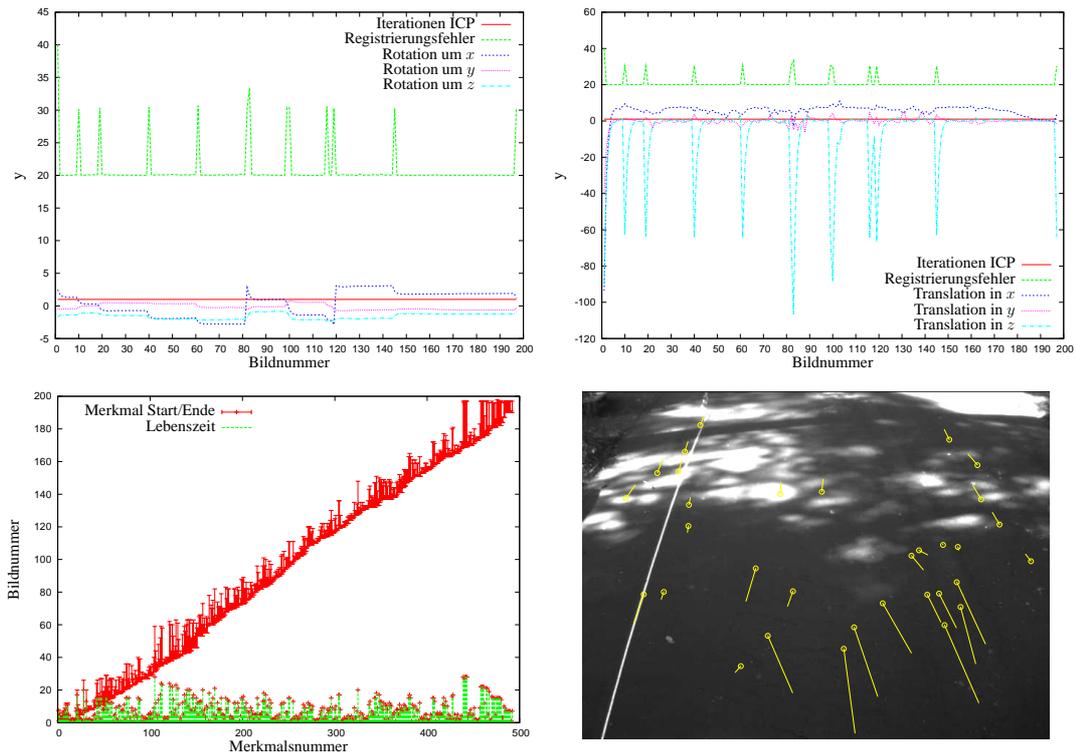


Abbildung 6.1: Problemanalysedaten der Kamerafahrt mit einem Kamerawinkel von 33° . Oben: links ist Verlauf der Rotation, rechts der Verlauf der Translationen zu sehen. Die oberste Linie entspricht dem gemessenen, mittleren Registrierungsfehler, verschoben um $+20$ in y . Unten: links ist eine Statistik über die Merkmalsverfolgung, rechts ein Intensitätsbild der Bildsequenz mit überlagerten Merkmalen.

wegungsrichtung haben. Die Merkmalsverfolgung hat dazu geneigt, Merkmale, die unten aus dem Bild wandern *müssten*, *vor sich her zu kehren*. Normalerweise würde man denken, die Filterung einer Korrespondenz nach der Konfidenz, also der Sicherheit über die Verfolgung eines Merkmals, würde solche Fehler ausschließen. Die Konfidenz lag aber auch bei diesen Merkmalen bei einem Wert von über 0,9. Somit war sich die Merkmalsverfolgung fälschlicherweise sicher, immer dasselbe Merkmal zu verfolgen zu haben.

6.3 Gesamtevaluation der Verfahren

Man kann zunächst einmal sagen, dass die synthetischen Kamerafahrten robuster als die realen geschätzt werden. Bezüglich der synthetischen Kamerafahrten liefert der Filterungsansatz über *RANSAC* die besten Ergebnisse. Tabelle 6.1 zeigt hierzu zusammenfassend eine Gegenüberstellung der einzelnen Verfahren. Basierend auf der Auswertung des euklidischen Abstands zwischen der Referenztrajektorie und der geschätzter Trajektorie kann man sehen, dass der *RANSAC*-Ansatz in drei von vier Fällen die geringsten Abweichung zur Referenz liefert. *Smooth Local 3D* und *Mahalanobis* sind bis auf einen Fall zueinander komplementär: dort wo der eine bessere Ergebnisse liefert, ist der andere immer schlechter. Anhand der gezeigten Funktionsgraphen sieht man, dass alle Ansätze einen sehr unterschiedlichen Fehlerverlauf zeigen. Wo der *RANSAC*- und der *Smooth Local 3D*-Ansatz relativ linear zunehmende Fehlerverläufe zeigen, zeigt der *Mahalanobis*-Ansatz deutlich einen wechselnden Fehlerverlauf, der aus groben Schwankungen besteht. Zum einen sagt dies aus, dass mit dem *RANSAC*- und dem *Smooth Local 3D*-Ansatz eine eher kontinuierliche Abweichung zur Referenz erzeugt wird. Zum anderen liefert der *Mahalanobis*-Ansatz einen schwankenden Schätzungsfehler. Hierdurch ergibt sich eine zufällige Verbesserung des Gesamtfehlers, da eine Fehleinschätzung auch eine erneute Annäherung, zurück zur Referenz bedeuten kann.

Da nur wenig geeignetes Auswertungsmaterial für die realen Kamerafahrten zur Verfügung stand, ist hier eine eindeutige, generelle Aussage über das bessere Verfahren nicht möglich. Allerdings ist klar erkennbar, dass die Abweichung von der Referenztrajektorie deutlich schneller als in den simulierten Kamerafahrten geschieht. Nicht nur die Tiefenwerte haben zur Verschlechterung beigetragen, gerade die Merkmalsverfolgung hat Schwachstellen im Umgang mit realen Bilddaten aufgezeigt. Da ein Helikopter von der Größe des maxiARTIS hauptsächlich im Außeneinsatz geflogen wird, muss eine Invarianz gegenüber der Belichtung erzielt werden. Solange die Merkmalsverfolgung keine robusten Daten liefern kann, werden die nachfolgenden Berechnungen darunter zu leiden haben.

Insgesamt kann man feststellen, dass der präsentierte Ansatz zur bildbasierten Schätzung der Eigenbewegung funktioniert und brauchbare Daten liefern kann.

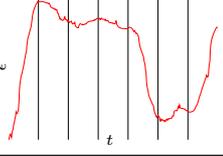
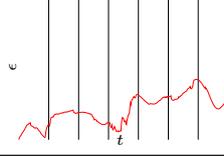
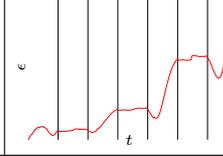
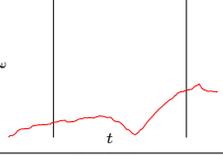
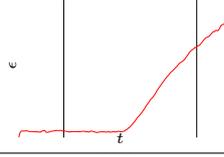
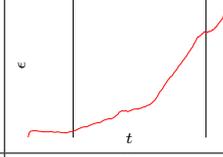
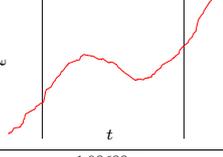
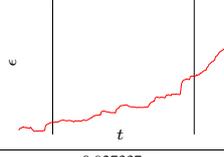
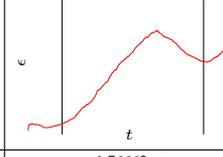
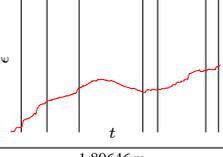
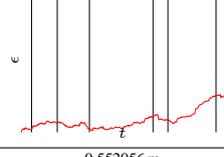
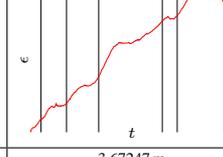
Quadrat, 420 Einzelbilder	Mahalanobis	RANSAC	Smooth Local 3D
Tendenzielles Fehlerverhalten			
Mittlerer, euklidischer Abstand	0,53742 m	0,18921 m	0,250481 m
Abweichung Endposition	0,711219 m	0,250529 m	0,762678 m
Dynamisch, 200 Einzelbilder	Mahalanobis	RANSAC	Smooth Local 3D
Tendenzielles Fehlerverhalten			
Mittlerer, euklidischer Abstand	0,700708 m	1,11999 m	1,4186 m
Abweichung Endposition	1,41075 m	3,65039 m	4,35497 m
Schleife, 420 Einzelbilder	Mahalanobis	RANSAC	Smooth Local 3D
Tendenzielles Fehlerverhalten			
Mittlerer, euklidischer Abstand	1,98688 m	0,927337 m	1,76662 m
Abweichung Endposition	4,10542 m	2,62236 m	2,87355 m
Hummelflug, 800 Einzelbilder	Mahalanobis	RANSAC	Smooth Local 3D
Tendenzielles Fehlerverhalten			
Mittlerer, euklidischer Abstand	1,80646 m	0,552056 m	3,67247 m
Abweichung Endposition	2,88574 m	1,76173 m	5,6905 m

Tabelle 6.1: Zusammenfassende Übersicht die Einzelverfahren. Alle hier gezeigten Werte basieren auf der Auswertung des euklidischen Abstands zwischen den Referenztrajektorie und der geschätzten Trajektorie. Die Funktionsgraphen zeigen den Abstandsfehlerverlauf für das jeweilige Verfahren. Die senkrechten Striche markieren den Zeitpunkt einer markanten Bewegungsänderung, zum Beispiel den Beginn oder das Ende einer Kurve oder Kurvenwechsel etc.

6.4 Referenzmessung der Tiefenwerte der Stereokamera

Wie in den Grundlagen unter 2.2 schon erwähnt, weist die verwendete Stereokamera eine von der Entfernung abhängige Tiefengenauigkeit auf. Der dort gezeigte Verlauf des Graphen 2.6 basiert auf einem optimalen, aber immer noch theoretischen Modell. Um eine Aussage über die im realen Einsatz erzielbare Tiefengenauigkeit der Stereokamera zu treffen – und um damit gleichzeitig auch einen Anteil an den schlechten Ergebnissen der realen Kamerafahrten zu zeigen – wurde ein Feldexperiment mit einer neu kalibrierten Stereokamera durchgeführt. Der Aufbau des Experiments ist in der folgenden Abbildung 6.2 zu sehen:

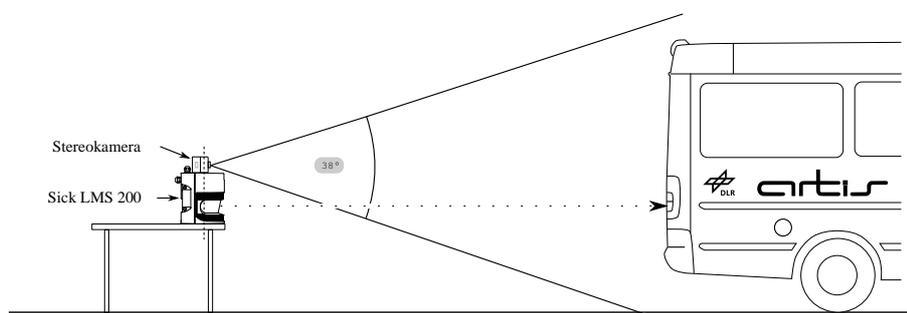


Abbildung 6.2: Der Experimentalaufbau der Ground-Truth-Messung.

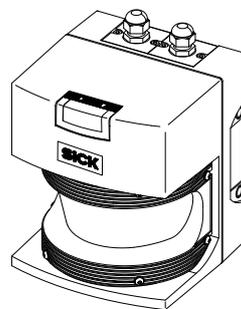
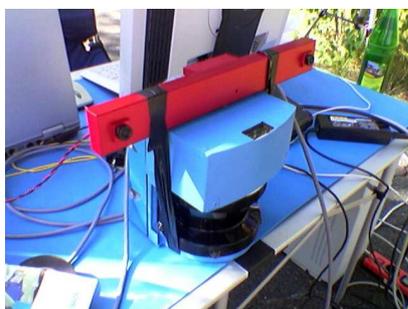


Abbildung 6.3: Foto des Versuchsaufbaus (links) und Produktabbildung des Lasermesssystems SICK LMS 200 (rechts).

Die Stereokamera wurde auf dem Gehäuse eines SICK LMS 200 Lasermesssystems [SIC06]

(Abbildung 6.3) montiert, so dass die Sensorflächen beider Geräte in der gleichen Ebene liegen² und somit die Entfernungsmessung vergleichbare Werte liefern kann. Die Messsysteme sind stationär auf einem Tisch angebracht und auf ein mobiles Objekt – hier der ARTIS-Bus – ausgerichtet. Durch die Verwendung des Busses wird sichergestellt, dass zum einen der Laser eine gute Reflexion erhält und zum anderen die Stereokamera ein ausreichend großes Objekt zur Messung der Disparität hat. Es wurden insgesamt 24 Messungen durchgeführt, angefangen bei einer Entfernung zum ARTIS-Bus von ca. 6,5 m bis zu einer Gesamtentfernung von ca. 50 m.

Abbildung 6.5 zeigt die auf das Wesentliche reduzierten Ergebnisse der Messungen inklusive der dazugehörigen Tiefenbilder.³ Zunächst kann man eindeutig sehen, dass die Tiefengenauigkeit wie erwartet mit der Entfernung abnimmt. Jedoch nimmt der mit der Tiefe zunehmende Fehler schneller zu, als im theoretischen Modell⁴ angenommen (siehe Abbildung 6.4). Bei einer Entfernung von 20 m sollte dem Modell nach die Tiefenabweichung bei ca. 0,5 m liegen – die real gemessene Tiefenabweichung beläuft sich aber auf ca. 5,88 m. Letztendlich wird der Fehler derart groß, so dass schon bei einer Entfernung von 48 m die Abweichung größer als die Entfernung wird (ca. 50,76 m). Die Messwertkurve zeigt einen systematischen Fehlerverlauf, der der quadratischen Charakteristik des Modells entspricht, aber um ein Vielfaches gestaucht ist. Aus diesen Ergebnissen kann man schlussfolgern, dass die Stereokamera für ein auf Punktmengen basierendes Verfahren womöglich nur bedingt geeignet ist, gerade dann, wenn man eine große Flughöhe für den Helikopter voraussetzt. Nur in einem extremen Nahbereich (< 10 m) scheint die Kamera relativ zuverlässig Entfernungswerte zu liefern, um genaue Resultate zur Schätzung der Eigenbewegung liefern zu können.

Da die durchgeführte Referenzmessung nur ein einzelnes Experiment darstellt, ist fraglich, inwiefern man diese Ergebnisse zur Erstellung einer Ausgleichsfunktion nutzen kann. Jedoch kann man klar sagen, dass dieses Experiment zur Verifizierung wiederholt werden muss.

²angedeutet durch die senkrecht verlaufende, gestrichelte Linie.

³Eine detaillierte Messwerttabelle ist im Anhang 8.2 zu finden.

⁴siehe Gleichung 2.2 auf Seite 19.

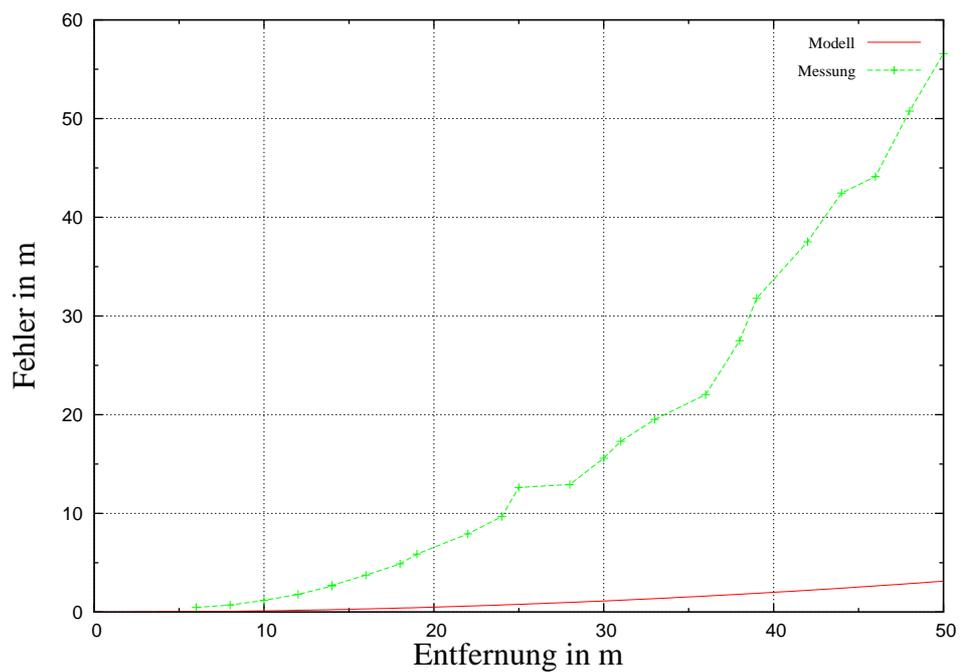


Abbildung 6.4: Vergleich der Tiefenauflösung zwischen Modell und Feldversuch.

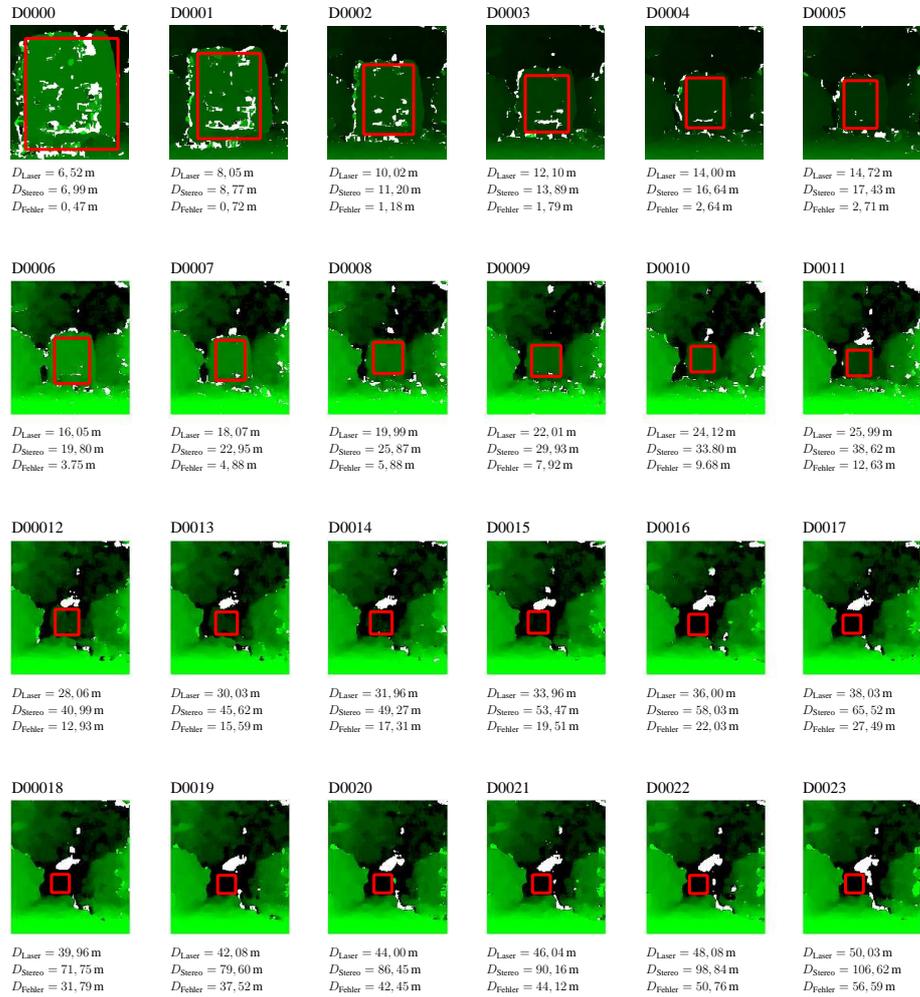


Abbildung 6.5: Tiefenbilder mit jeweiligen Entfernungs- und Fehlerangaben. Die Umrangungen entsprechen dem für die Auswertung benutzten Bildbereich. Bilder D0006-D0023 wurden zu Darstellungszwecken durch eine Farbwertanpassung verändert. Eine ausführliche Wertetabelle im Anhang unter 8.2 zu finden.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Gesamtlösung zur bildbasierten Schätzung der Eigenbewegung eines unbemannten Helikopters entwickelt und bewertet. Dabei erschließt sich, dass die bildbasierte Schätzung der Eigenbewegung aus mehreren Teilschritten besteht, die aufeinander aufbauend, ein Gesamtsystem bilden.

Am Anfang der Teilschritte steht die Erzeugung von Tiefenbildern, die in diesem Falle über ein Stereosystem generiert werden. Mit zunehmender Entfernung wird eine Stereokamera immer ungenauer in der Tiefenschätzung. Deshalb wird versucht, durch die Verwendung einer schräg nach unten gerichteten Kamera, die Merkmalsausbeute zu erhöhen, da hierdurch der Nahbereich, also der *genauere* Messbereich der Kamera, besser ausgenutzt wird.

Darauf folgt die Merkmalsdetektion und deren Verfolgung im Ortsraum. Die Informationen aus der Merkmalsverfolgung werden mit den Tiefenbildern der Stereokamera zu korrespondierenden 3D-Punktmengen verknüpft, aus denen schließlich die Eigenbewegung geschätzt werden soll. Da diese Punktmengen jedoch mit Ausreißern behaftet sind und auch die Korrespondenzen nicht immer richtig durch die Merkmalsverfolgung bestimmt werden, muss eine Filterung der Korrespondenzen erfolgen.

Diese Filterung bestimmt den Schwerpunkt dieser vorliegenden Arbeit. Es werden hierzu insgesamt drei Verfahren untersucht. Das erste Verfahren zur Filterung verwendet die *Mahalanobis-Distanz*. Sie erlaubt die Einbeziehung der Zunahme der Tiefenungenauigkeit

in die Entscheidung, ob eine Korrespondenz verworfen wird, oder nicht. Hierbei wird ein iterativer Ansatz über den ICP verfolgt, der nur die Punktmengen, jedoch nicht die Korrespondenzen aus der Merkmalsverfolgung benötigt. Die beiden anderen Verfahren zur Filterung verwenden diese Korrespondenzen und filtern sie auf eine jeweils unterschiedliche Art. Der Ansatz über *Smooth Local 3D* untersucht dabei die Bewegungsvektoren der Korrespondenzen und filtert all jene heraus, deren Bewegungsunterschied zu der Hauptbewegung der anderen über einem Schwellwert liegt. Der Ansatz über *RANSAC* geht einen anderen Weg: es wird so oft eine Stichprobe aus der Korrespondenzmenge als Modell genommen und verglichen, so dass mit einer gewissen Sicherheit davon ausgegangen wird, das richtige Modell gefunden zu haben.

Der letzte Schritt zur bildbasierten Schätzung der Eigenbewegung besteht in der Akkumulation der von der Singulärwertzerlegung berechneten relativen Rotationen und Translationen zu einer Trajektorie, die den zurückgelegten Bewegungsverlauf der Kamera und somit auch des Helikopters widerspiegelt.

Anhand von sowohl simulierten als auch realen Kamerafahrten wird die Funktionalität der Verfahren analysiert. Hierzu kann man sagen, dass in der simulierten Umgebung alle Verfahren bis zu einem gewissen Grad brauchbare Ergebnisse liefern, jedoch der Ansatz über *RANSAC* insgesamt die besten Ergebnisse liefert. Der Ansatz über *Smooth Local 3D* hat hauptsächlich Probleme mit der richtigen Schätzung der Rotationen. Der Ansatz über die *Mahalanobis-Distanz* ist etwas schlechter als der *RANSAC*-Ansatz einzuordnen.

Die Experimente mit realen Bilddaten werden hauptsächlich durch die Vorarbeit der Merkmalsverfolgung und der Ungenauigkeit der Tiefenwerte beeinflusst. Hier scheint schon der Anfang der gesamten Verarbeitungskette nicht optimal zu funktionieren. Nur die Verfahren *Smooth Local 3D* und *Mahalanobis-Distanz* haben eine Trajektorie bei einem der Versuche ergeben, der *RANSAC* hat keine Trajektorie generieren können. Die beiden geschätzten Trajektorien aus den Realbilddaten zeigen eine im Vergleich zu den simulierten Kamerafahrten schnellere Drift, wobei die *Mahalanobis-Distanz* das beste Ergebnis liefern konnte – jedoch ist hier eine Verallgemeinerung aufgrund der geringen Anzahl an realen Experimenten nicht möglich. Die Orientierung der geschätzten Trajektorie und somit auch eine ungewollte Abweichung, hängt von einer exakten, räumliche Kalibrierung der Kamera zum Versuchsträger ab.

Es ist wichtig zu sagen, dass alle Gesamtsysteme zur rein bildbasierten Bewegungsschätzung auf relativen Berechnungen beruhen, die immer einen gewissen Fehler beinhalten und je nach Robustheit der einzelnen Verfahren, unweigerlich zur Drift führen, es sei denn man stützt den Prozess durch andere Sensoren. Diese Stützung oder auch *Fusion* mit anderen Sensoren, wie zum Beispiel der IMU, wäre eine Möglichkeit, die Robustheit der gesamten Lösung zu erhöhen.

Bevor man aber versucht, nur durch die Fusion das Gesamtsystem zu verbessern, sollten kommende Untersuchungen verstärkt in der ersten Stufe¹ der Verarbeitungskette durchgeführt werden. Hierzu zählt zum einen die genauere Evaluation der Möglichkeiten zur robusten Merkmalsverfolgung bei Realbilddaten. Es sollten Bemühungen unternommen werden, eine Invarianz gegenüber der Belichtung zu schaffen. Des Weiteren müssen zusätzliche Tests mit der Stereokamera durchgeführt werden, um eine Aussage darüber zu treffen, inwiefern eine Verbesserung der Kalibrierung auch eine Verbesserung der Tiefenauflösung bringt, da die in dieser Arbeit durchgeführte Referenzmessung zur Genauigkeit der Tiefenwerte der Stereokamera sehr ernüchternd ausgefallen ist.

¹Gemeint sind die Intensitätsbild- und Tiefenkartengewinnung durch die Stereokamera und die Merkmalsverfolgung.

Kapitel 8

Anhang

8.1 Spezifikation der Stereokamera

STOC-Prozessor	Xilinx Spartan 3 - 1000 (FPGA)
Prozessor Performanz	35 Gops bei 85 MHz
Anzahl der Disparitäten	64; 1/8 Pixel interpolation
Fenstergröße für Korrelation	15x15 Pixel
Laplacian of Gaussian Kernelfilter	9x9 Pixel
Post-Filterung	Textur- und Eindeutigkeitsüberprüfung
Sensor	1/3“ CMOS (Micron MT9V022) 640×480 Globaler Verschluss Progressive Scan Monochrom
Format	8 Bit Monochrom 10 Bit Disparität
Bildwiederholraten	7.5, 15, 30 Hz 6.5, 12.5, 25 Hz Max 30 Hz bei 640×480
Belichtung	Linie bis Vollformat
Verstärkung	0 - 12 dB (1x - 4x)
Spannungsversorgung	IEEE 1394 Bus-Spannung maximal 14 V
Stromverbrauch	Bildsensor: 1,45 W STOC-Platine: 0,84 W Gesamt: 2,29W
Synchronisierung	Intern: Pixel-Gerastet (pixel-locked) Extern: 60 μ s
Linsen	Miniaturlinsen mit jeweils 4,1 mm Brennweite
Öffnungswinkel	37.85° vertikal, 49.13° horizontal
Abmessungen	33.5 cm × 4.3 cm × 4.1 cm (bxhxt)
Gewicht	454 g (mit Miniaturlinsen)
Stereo-Basislinie	30 cm
SVS-Software	Linux Kernel 2,4 / 2,6
Umgebungsbedingungen	0-40°C, <80% rel. Luftfeuchtigkeit (nichtkondensierend)

Tabelle 8.1: Technische Spezifikation der verwendeten Stereokamera.

8.2 Auswertungsdaten

8.2.1 Referenzmessung der Tiefenwerte der Stereokamera

Nr.	Datei	D_{LASER}	D_{Stereo}	D_{Fehler}	#Pixel	Grauwert	#Pixel _{total}	Tiefenwertstreuung	Disparität	$D_{\text{StereoDisparität}}$
1	D0000	6,52m	6,99m	0,47m	1272	116	34155	D_{max} 7,16m D_{min} 6,81m D_{Δ} 0,35m	119	6,98m
					4490	117	67,32			
					4275	118				
					4624	119				
					5604	120				
					1575	121				
					1153	122				
					1153	122				
2	D0001	8,05m	8,77m	0,72m	1070	92	23146	D_{max} 9,03m D_{min} 8,75m D_{Δ} 0,28m	94,5	8,80m
					3062	93	78,81			
					4070	94				
					3967	95				
					5155	96				
					918	97				
					918	97				
					918	97				
3	D0002	10,02m	11,20m	1,18m	1188	72	16128	D_{max} 11,54m D_{min} 10,93m D_{Δ} 0,61m	74	11,23m
					3572	73	85,04			
					4329	74				
					3332	75				
					1294	76				
					1294	76				
					1294	76				
					1294	76				
4	D0003	12,10m	13,89m	1,79m	1444	58		D_{max} 14,33m D_{min} 13,62m D_{Δ} 0,71m	59,5	13,97m
					2541	59				
					4398	60				
					1114	61				
					1114	61				
					1114	61				
5	D0004	14,00m	16,64m	2,64m	461	48	10368	D_{max} 17,31m D_{min} 15,98m D_{Δ} 1,33m	50	16,62m
					3169	49	82,98			
					2933	50				
					1337	51				
					703	52				
					703	52				
					703	52				
6	D0005	14,72m	17,43m	2,71m	1250	46	8736	D_{max} 18,06m D_{min} 16,96m D_{Δ} 1,10m	47,5	17,50m
					2187	47	85,2			
					3237	48				
					366	49				
					366	49				
					366	49				
7	D0006	16,05m	19,80m	3,75m	403	40	9152	D_{max} 20,77m D_{min} 18,88m D_{Δ} 1,89m	42	18,78m
					2506	41	80,69			
					2603	42				
					1325	43				
					548	44				
					548	44				
					548	44				
8	D0007	18,07m	22,95m	4,88m	345	34	6745	D_{max} 24,44m D_{min} 22,46m D_{Δ} 1,98m	35,5	23,41m
					469	35	80,01			
					2992	36				
					1591	37				
					1591	37				
					1591	37				
9	D0008	19,99m	25,87m	5,88m	918	31	5576	D_{max} 26,80m D_{min} 25,18m D_{Δ} 1,62m	32	25,97m
					3095	32	80,38			
					469	33				
					469	33				
					469	33				
10	D0009	22,01m	29,93m	7,92m	427	25	5355	D_{max} 33,24m D_{min} 28,65m D_{Δ} 4,59m	27	30,77m
					250	26	86,76			
					792	27				
					2798	28				
					379	29				
					379	29				
					379	29				
11	D0010	24,12m	33,80m	9,68m	293	23	4200	D_{max} 36,13m D_{min} 31,96m D_{Δ} 4,17m	24,5	33,93m
					1410	24	82,67			
					1635	25				
					134	26				
					134	26				
					134	26				
12	D0011	25,99m	38,62m	12,63m	283	20	3294	D_{max} 41,55m D_{min} 36,13m D_{Δ} 5,42m	21,5	38,76m
					1655	21	96,11			
					909	22				
					319	23				
					319	23				
					319	23				

Tabelle 8.2: Auswertungsdaten der Stereo-Referenzmessung, Messungen 1-12 von 24.

Nr.	Datei	D_{LASER}	D_{Stereo}	D_{Fehler}	#pixel	Grauwert	#pixel _{total}	Tiefenwertstreuung	Disparität	$D_{\text{StereoDisparität}}$
13	D0012	28,06m	40,99m	12,93m	203 395 763 592 205	18 19 20 21 22	2530 85,3	D_{max} 46,16m D_{min} 37,77m D_{Δ} 8,39m	20	41,55m
14	D0013	30,03m	45,62m	15,59m	229 529 825 509 295	16 17 18 19 20	2650 90,08	D_{max} 51,93m D_{min} 41,55m D_{Δ} 10,38m	18	46,16m
15	D0014	31,96m	49,27m	17,31m	360 510 881 327	15 16 17 18	2544 81,68	D_{max} 55,39m D_{min} 46,16m D_{Δ} 9,23m	16,5	50,41m
16	D0015	33,96m	53,47m	19,51m	216 669 837	14 15 16	1950 88,31	D_{max} 59,35m D_{min} 51,93m D_{Δ} 7,42m	15	55,39m
17	D0016	36,00m	58,03m	22,03m	294 868 433	13 14 15	1824 87,45	D_{max} 63,92m D_{min} 55,93m D_{Δ} 7,99m	14	59,35m
18	D0017	38,03m	65,52m	27,49m	312 502 1164	11 12 13	2244 88,15	D_{max} 75,54m D_{min} 63,92m D_{Δ} 11,62m	12	69,24m
19	D0018	39,96m	71,75m	31,79m	269 498 749	10 11 12	1760 86,14	D_{max} 83,09m D_{min} 69,24m D_{Δ} 13,85m	11	75,45m
20	D0019	42,08m	79,60m	37,52m	304 424 364	9 10 11	1305 83,68	D_{max} 92,32m D_{min} 75,54m D_{Δ} 16,78m	10	83,09m
21	D0020	44,00m	86,45m	42,45m	186 512 378 159	8 9 10 11	1376 89,75	D_{max} 103,86m D_{min} 75,54m D_{Δ} 28,32m	9,5	87,71m
22	D0021	46,04m	90,16m	44,12m	400 675 206	8 9 10	1320 122,2	D_{max} 103,86m D_{min} 83,09m D_{Δ} 20,77m	9	92,32m
23	D0022	48,08m	98,84m	50,76m	332 532 409	7 8 9	1344 94,72	D_{max} 118,70m D_{min} 92,32m D_{Δ} 26,38m	8	103,86m
24	D0023	50,03m	106,62m	56,59m	175 372 352 123	6 7 8 9	1080 94,63	D_{max} 138,48m D_{min} 92,32m D_{Δ} 46,16m	7,5	111,28m

Tabelle 8.3: Auswertungsdaten der Stereo-Referenzmessung, Messungen 13 bis 24.

8.2.2 Tiefenwert-Testtabelle der Simulation

Punkt	Koordinate	D_{Blender}	D_{DIP}	$\epsilon_{D_{\text{DIP}} \rightarrow \text{Blender}}$	$\epsilon_{\text{Intensität}}$
P01	80,600	14,50	14,61510	0,11511	0
P02	240,600	14,00	14,11110	0,11114	0
P03	400,600	13,50	13,52800	0,02803	0
P04	560,600	13,00	13,09510	0,09514	0
P05	80,180	12,50	12,59150	0,09148	0
P06	240,180	12,00	12,03600	0,03597	0
P07	400,180	11,50	11,52740	0,02741	0
P08	560,180	11,00	11,06010	0,06008	0
P09	80,300	10,50	10,56060	0,06059	0
P10	240,300	10,00	10,04230	0,04228	0
P11	400,300	9,50	9,51681	0,01681	0
P12	560,300	9,00	9,04360	0,04360	0
P13	80,420	8,50	8,52548	0,02548	0
P14	240,420	8,00	8,02398	0,02398	0
P15	400,420	7,50	7,50868	0,00868	0
P16	560,420	7,00	7,02529	0,02529	0

Tabelle 8.4: Testwerttabelle zur Abbildungstreue der Blender-Kamera. Die letzte Spalte gibt den Fehler zum zur Tiefe zugehörigen Intensitätswert der echten Stereokamera an.

Literaturverzeichnis

- [AHB87] ARUN, K. S. ; HUANG, T. S. ; BLOSTEIN, S. D.: Least-squares fitting of two 3-D point sets. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 (1987), Nr. 5, S. 698–700. – ISSN 0162–8828
- [AKF99] AMIDI, O. ; KANADE, T. ; FUJITA, K.: A visual odometer for autonomous helicopter flight. In: *Robotics and Autonomous Systems* 28 (1999), Nr. 2-3, S. 185–193
- [BM92] BESL, P. ; MCKAY, N.: A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2, S. 239–256. – ISSN 0162–8828
- [Bou02] BOUGUET, J.-Y.: *Pyramidal Implementation of the Lucas Kanade Feature Tracker*. 2002. – Intel Corporation, Microprocessor Research Labs. OpenCV Documents.
- [CLD07] CORKE, P. ; LOBO, J. ; DIAS, J.: An Introduction to Inertial and Visual Sensing. In: *The International Journal of Robotics Research* 26 (2007), Nr. 6, S. 519–535. – ISSN 0278–3649
- [CMM06] CHENG, Y. ; MAIMONE, M. ; MATTHIES, L.: Two years of visual odometry on the Mars Exploration Rovers. In: *Journal of Field Robotics, Special Issue on Space* 24 (2006), März, Nr. 3, S. 169–186. – ISSN 1556–4959
- [DBT03] DITTRICH, J. S. ; BERNATZ, A. ; THIELECKE, F.: Intelligent Systems Research Using a Small Autonomous Rotorcraft. In: *2nd AIAA Unmanned Un-*

- limited Systems, Technologies, and Operations - Aerospace, Land, and Sea Conference and Workshop Exhibit* (2003), 9. ISBN 1-56347-640-1
- [FB81] FISCHLER, M. A. ; BOLLES, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381-395. – ISSN 0001-0782
- [Gla89] GLASSNER, A. (Hrsg.): *An Introduction to Ray Tracing*. London, UK : Academic Press Ltd., 1989. – 327 S. – ISBN 0-12-286160-4
- [Goo04] GOORMANN, L.: *Objektorientierte Bildverarbeitungsalgorithmen zum relativen Hovern eines autonomen Helikopters*, DLR / FH Braunschweig/Wolfenbüttel, Diplomarbeit, 2004
- [Gut04] GUTH, O.: *Biologisch inspirierte Bildverarbeitungsalgorithmen zur Realisierung eines Geländefolfluges für einen autonomen Kleinhubschrauber*, DLR / FH Braunschweig/Wolfenbüttel, Diplomarbeit, 2004
- [Hor90] HORN, B. K. P.: *Recovering Baseline and Orientation from Essential Matrix*. Journal of the Optical Society of America, Januar 1990
- [HS88] HARRIS, C. ; STEPHENS, M.: A combined corner and edge detector. In: *Fourth Alvey Vision Conference* (1988), S. 147-151
- [HZ04] HARTLEY, R. I. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. 2. Auflage. Cambridge University Press, 2004. – ISBN 978-0521540513
- [KS07] KELLY, J. ; SUKHATME, G. S.: An Experimental Study of Aerial Stereo Visual Odometry. In: *IFAC - International Federation of Automatic Control Symposium on Intelligent Autonomous Vehicles* (2007), September
- [MS06] MILELLA, A. ; SIEGWART, R.: Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point. In: *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems* (2006), S. 21. ISBN 0-7695-2506-7

- [NBM⁺04] NESNAS, I.A.D. ; BAJARACHARYA, M. ; MADISON, R. ; BANDIARI, E. ; KUNZ, C. ; DEANS, M. ; BUALAT, M.: Visual Target Tracking for Rover-based Planetary Exploration. In: *Proceedings of the IEEE Aerospace Conference 2* (2004), März, S. 747–761
- [NLHS05] NÜCHTER, A. ; LINGEMANN, K. ; HERTZBERG, J. ; SURMANN, H.: 6D SLAM with approximate data association. In: *Proceedings of the 12th International Conference on Advanced Robotics* (2005), Juli, S. 242–249
- [NNB04] NISTÉR, D. ; NARODITSKY, O. ; BERGEN, J. R.: Visual Odometry. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2004), S. 652–659. – ISBN 0–7695–2158–4
- [Nor70] NORMSTELLE LUFTFAHRT: *LN 9300: Luft- und Raumfahrt; Begriffe, Größen und Formelzeichen der Flugmechanik*. Beuth Verlag GmbH, Berlin, 1970
- [RS04] ROOSEDAAL, T. ; SELLERI, S.: *The Official Blender 2.3 Guide: Free 3D Creation Suite for Modeling, Animation, and Rendering*. San Francisco, CA, USA : No Starch Press, 2004. – ISBN 1593270410
- [SIC06] SICK AG WALDKIRCH: *LMS200/211/221/291 Dokumentation deutsch*. 12-2006. Nimburger Straße 1, 79276 Reute, Germany, 2006
- [SK08] SICILIANO, B. (Hrsg.) ; KHATIB, O. (Hrsg.): *Springer Handbook of Robotics*. Berlin : Springer, 2008. – 1611 S. – ISBN 978–3–540–23957–4
- [SP07] SUENDERHAUF, N. ; PROTZEL, P.: *Stereo Odometry – A Review of Approaches*. 2007. – unveröffentlicht
- [SS02] SCHARSTEIN, D. ; SZELISKI, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. In: *International Journal of Computer Vision* 47 (2002), Nr. 1-3, S. 7–42. – ISSN 0920–5691
- [ST94] SHI, J. ; TOMASI, C.: Good Features to Track. In: *IEEE Conference on Computer Vision and Pattern Recognition* (1994), Juni, S. 593–600. ISBN 0–8186–5825–8

- [TBF05] THRUN, S. ; BURGARD, W. ; FOX, D.: *Probabilistic Robotics*. MIT Press, 2005. – 667 S. – ISBN 0-262-20162-3
- [TK91] TOMASI, C. ; KANADE, T.: Shape and Motion from Image Streams: a Factorization Method - Part 3 Detection and Tracking of Point Features / Computer Science Department, Carnegie Mellon University. Pittsburgh, PA, April 1991 (CMU-CS-91-132). – Forschungsbericht
- [Ume91] UMEYAMA, S.: Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991), April, Nr. 4, S. 376–380. – ISSN 0162-8828
- [Vid07] VIDERE DESIGN: *Stereo-on-a-Chip [STOC] Stereo Head User Manual*. 1.3. Menlo Park California 94025 USA, 2007
- [Wil98] WILLIAMSON, T.: *A High-Performance Stereo Vision System for Obstacle Detection*. Pittsburgh, PA, Carnegie Mellon University, Diss., September 1998
- [WL90] WALTZ, E. L. ; LLINAS, J.: *Multisensor Data Fusion*. Norwood, MA, USA : Artech House, Inc., 1990. – 450 S. – ISBN 0890062773
- [Zha94] ZHANG, Z.Y.: Iterative Point Matching for Registration of Free-Form Curves and Surfaces. In: *International Journal of Computer Vision* 13 (1994), Oktober, Nr. 2, S. 119–152. – ISSN 0920-5691
- [ZON⁺06] ZHU, Z. ; OSKIPER, T. ; NARODITSKY, O. ; SAMARASEKERA, S. ; SAWHNEY, H. S. ; KUMAR, R.: An Improved Stereo-based Visual Odometry System. In: *Performance Metrics for Intelligent Systems* (2006), S. 149–156