Kamerabasierte Egomotion-Bestimmung mit natürlichen Merkmalen zur Unterstützung von Augmented-Reality-Systemen

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der Fakultät für
Bauingenieur-, Geo- und Umweltwissenschaften

des Karlsruher Institutes für Technologie (KIT)

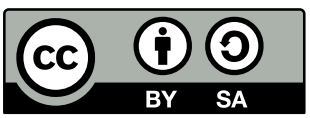
genehmigte
DISSERTATION

von
Dipl.-Ing. Sven Wursthorn
aus Karlsruhe

Tag der mündlichen Prüfung:
30. April 2014

Hauptreferent:	Prof. Dr.-Ing. habil. Stefan Hinz
Korreferent:	Prof. Dr.-Ing. Olaf Hellwich
Korreferent:	Prof. Dr.-Ing. habil., Dr. h.c. Hans-Peter Bähr

Karlsruhe 2014



Kurzfassung

Durch die heute schon selbstverständliche Integration von Positionierungs- und Orientierungssensoren in der überwiegenden Mehrheit verfügbarer Smartphones und Tablets kann auf ortsbezogene Informationen aller Art rund um den aktuellen Aufenthaltsort des Nutzers automatisch zugegriffen werden, ohne dass dieser die Verbindung von Ort und Information selbst herstellen muss.

Mit der integrierten Kamera ist es möglich, vollständig im Raum lokalisierte und orientierte Ansichten der direkten Umgebung auf dem Display anzeigen zu können. Anwendungen aus dem Bereich der Augmented Reality haben dadurch Einzug in das Alltagsleben erhalten. Ortsbezogene Informationen können so direkt mit dem Kamerabild überlagert werden.

Zum Erreichen einer guten Überlagerung müssen zwei Probleme gelöst werden: Zum einen werden für diese Überlagerungen detaillierte und aktuelle dreidimensionale Daten von der direkten Umgebung des Nutzers benötigt, damit die vom Computer generierten Ansichten der zusätzlichen Informationen räumlich in die Umgebung integriert werden können. Zum Anderen kann die hierfür benötigte Genauigkeit und Stabilität von den integrierten Sensoren nicht geliefert werden.

Daher liegt es nahe, das Kamerabild selbst zur Schätzung und Stabilisierung der äußeren Orientierung zu nehmen. Mit Stereokamerasystemen und Tiefenbildkameras kann sowohl die Erfassung der Umgebungsgeometrie als auch die Eigenbewegungsschätzung durchgeführt werden. Aktuelle Entwicklungen deuten bereits heute auf die Integration solcher Kamerasysteme in künftige mobile Geräte hin.

Dadurch motiviert werden in dieser Arbeit die Möglichkeiten und Verfahren zur Eigenbewegungsschätzung mit Hilfe von Stereokamerasystemen und Tiefenbildkameras untersucht. Dieses Forschungsfeld ist auch unter dem Namen *Visual Odometry* bekannt.

Nach einer kurzen Einführung in Augmented Reality werden zunächst die Grundlagen der Bewegungsschätzung mit Bildern analysiert. Anschließend werden die genutzten Aufnahmesysteme vorgestellt: Stereokamera und Kinect Tiefenbildkamera. Mit einer Diskussion über die Möglichkeiten der Merkmalsextraktion und -Verfolgung in Bildsequenzen zum Gebrauch in Augmented-Reality-Anwendungen schließt der erste Teil.

Im zweiten Teil werden die Anwendungsgebiete und vorhandene Verfahren aus dem Bereich der Stereo-Egomotion analysiert und ein eigener Ansatz, der sowohl mit Stereobildsequenzen als auch mit Tiefenbildsequenzen zurechtkommt, vorgestellt. Die eigenen Lösungen werden mit ausgesuchten Verfahren abschließend mit einem kombinierten Aufnahmesystem verglichen und bewertet.

Abstract

Today it is common place to instantly access spatial information near the user's actual location with onboard positioning and orientation sensors built into the majority of mobile phones and tablet computers. The link between spatial information and actual location is done automatically without the need for the user's interference.

Once oriented, the built-in camera can show views of the surrounding environment on the display with known postion and orientation. With this, augmented reality applications have entered everyday life. The camera view can now be augmented with spatial information.

Two problems have to be solved to achieve an accurate overlay: On the one hand, up-to-date three-dimensional information of the surrounding space is necessary to overlay computer generated content and let it fit into the space in front of the camera. On the other hand, the integrated postioning and orientation sensors are not precise enough, to deliver the accuracy and stability needed.

Hence it is natural to use the camera image itself for estimating and stabilizing its pose. With stereo camera systems and depth cameras it is possible to both capture 3D geometry and etsimate ego-motion. Current development indicates the integration of such camera systems in the future.

Motivated by this fact possibilities and methods for image based ego-motion estimation with both stereo camera systems and depth cameras will be analyzed. The research topic is also known as *visual odometry*.

After a short introduction to augmented reality, the fundamentals of image based ego-motion estimation will be analyzed and the camera systems used throughout this work will be introduced and compared. A discussion on methods for detection and tracking of image features, especially for augmented reality, will close the first part of this work.

In the second part, applications and plattforms of stereo ego-motion systems will be covered. A solution for an ego-motion estimation method that will work with both stereo cameras and depth cameras will be introduced. This self-developed method will be compared with existing solutions with a combined system that is able to conjointly capture all images of both camera systems.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Der Zugang zu digitalen Informationen ist heute durch die beinahe flächendeckende Verbreitung mobiler, netzwerkfähiger Endgeräte allgegenwärtig. Zudem sind viele dieser Informationen ortsbezogen.

Durch die heute schon selbstverständliche Integration von Positionierungs- und Orientierungssensoren in der überwiegenden Mehrheit verfügbarer Smartphones und Tablets kann auf ortsbezogene Informationen aller Art rund um den aktuellen Aufenthaltsort des Nutzers automatisch zugegriffen werden, ohne dass dieser die Verbindung von Ort und Information selbst herstellen muss.

Mit der integrierten Kamera ist es möglich, vollständig im Raum lokalisierte und orientierte Ansichten der direkten Umgebung auf dem Display anzeigen zu können. Damit haben Anwendungen aus dem Bereich der Augmented Reality - Erweiterte Realität - Einzug in das Alltagsleben erhalten. Ortsbezogene Informationen können so direkt mit dem Kamerabild überlagert werden.

Gerade auf für Jedermann verfügbaren mobilen Endgeräten sind derartige Anwendungen oft noch verspielt und bilden eine nette Ergänzung. Das Konzept der Erweiterten Realität ist jedoch so stark, dass der Wunsch nach mehr Professionalität und exakter Überlagerung virtueller Inhalte zu besseren integrierten Sensoren führen wird.

Für diese neuen Anwendungen müssen vom direkten Nahbereich des Nutzers aktuelle, detaillierte 3D-Informationen vorliegen. Stereo- oder Multikamerasysteme und Tiefenbildsensoren können diese Informationen liefern. Aber wie realistisch ist die allgemeine Verfügbarkeit optischer 3D-Messsysteme in mobilen Endgeräten?

Die Firma Google z. B. hat vor kurzem einen Smartphoneprototypen mit integriertem 3D-Sensor und einer Fisheyekamera vorgestellt, der damit beworben wird, die Umgebung dreidimensional erfassen zu können. PrimeSense, die Entwickler der

Kinect Tiefenbildkamera, haben ein Sensormodul zur Integration in mobile Geräte im Angebot.

Dies führt zur wissenschaftlichen Motivation dieser Arbeit. Mit derartigen, bildgebenden Sensoren muss es auch möglich sein, die Eigenbewegungen zu schätzen und damit die bisher üblichen und oft auch unzureichenden Positionierung- und Orientierungssensoren zu unterstützen oder gar ersetzen zu können.

1.2 Zielsetzung der Arbeit

Die obige Motivation ergibt folgende Fragestellungen, deren Beantwortung die Ziele dieser Arbeit definieren.

- Welche bildgebenden Systeme können zur direkten 3D-Objektaufnahme im Nahbereich des Nutzers eines mobilen Augmented-Reality-Systems zum Einsatz kommen?

  Die einfachste Kamerakonfiguration, mit der man über den Umweg der Triangulation 3D-Objektkoordinaten messen kann ist ein Stereokamerasystem. Tiefenbildkameras wie die Kinect Kamera liefern direkt dichte 3D-Punkte.

- Wie kann eine Eigenbewegungsschätzung mit diesen Systemen durchgeführt werden?

  Diese Fragestellung führt zu dem großen Forschungsfeld der Eigenbewegungsbestimmung von Kamerasystemen - der *Visual Odometry*. Der Schwerpunkt liegt hier auf Verfahren, die mit Stereobildpaaren und Tiefenbildern arbeiten.

  Es sollen aber auch andere, besonders für die Erweiterte Realität geeignete, Verfahren untersucht werden.

  Sind die Verfahren aus den unterschiedlichen Anwendungsdomänen identisch oder gibt es zwischen Verfahren aus den Bereichen Robotik, Fahrerassistenzsysteme, unbemannte Fluggeräte oder Fussgänger Unterschiede?

- Kann das gleiche Verfahren sowohl auf ein Stereokamerasystem als auch auf eine Tiefenbildkamera angewandt werden?

  Sind die Bedingungen zur Schätzung von Eigenbewegungen bei Stereobildfolgen und Tiefenbildfolgen unterschiedlich? Können Gemeinsamkeiten für einen generischen Ansatz gefunden werden?

- Wie können die unterschiedlichen Verfahren zur Eigenbewegungsbestimmung verglichen werden?

  Wie kann eine Bewegungsfolge sowohl mit einem Stereokamerasystem als auch mit einer Tiefenbildkamera gemeinsam aufgenommen und das Ergebnis bewertet werden? Welche Bedingungen müssen erfüllt sein, um die Ergebnisse mit Referenzdaten vergleichen zu können? Wie werden die Referenzdaten erzeugt?

1.3 Gliederung der Arbeit

Kapitel 2-4 beschäftigen sich mit den Anforderungen und Rahmenbedingungen sowie den methodischen Grundlagen und Bildsensoren für die Egomotionbestimmung.

Darauf folgend bilden Kapitel 5 und 7 den wissenschaftlichen Kern der Arbeit. Hier werden zuerst vorhandene Ansätze aus unterschiedlichen Anwendungsdomänen analysiert. Aus den zuvor beschriebenen Grundlagen wird ein eigenes Verfahren zur Bewegungsschätzung konzipiert und die Implementierung mit ausgesuchten Verfahren verglichen.

Der aktuelle Stand wissenschaftlicher Arbeiten wird jeweils in den betreffenden Kapiteln beschrieben. Auf ein separates State of the Art Kapitel über alle verwendeten Methoden würde aufgrund ihrer Heterogenität den Rahmen der Arbeit sprengen. Im einzelnen ist die Arbeit wie folgt gegliedert:

- Kapitel 2 geht auf die Erweiterte Realität und deren Anforderungen an das Tracking ein.
- Kapitel 3 beschreibt zuerst die geometrischen Grundlagen zur Berechnung von Eigenbewegungen mit Stereokamerasystemen.
- Kapitel 4 beschreibt die Aufnahmesysteme, die in dieser Arbeit verwendet wurden: ein Stereokamerasystem und eine Kinect Tiefenbildkamera.
- Kapitel 5 beschreibt Verfahren zur automatischen Extraktion und Verfolgung von Bildmerkmalen, die für das Ermitteln der Kameraeigenbewegung erforderlich sind.
- Kapitel 6 beschreibt den Gesamtablauf verschiedene Systeme und Verfahren zur Berechnung der Eigenbewegungen.
- Kapitel 7 vergleicht und bewertet die beschriebenen Verfahren.
- Mit Zusammenfassung und Ausblick schließt die Arbeit in Kapitel 8.

2 Erweiterte Realität

Unter Erweiterter Realität (engl. augmented reality, AR) versteht man die Überlagerung der Sinneseindrücke, die ein Mensch von seiner Umgebung aufnimmt, mit zusätzlichen, vom Computer generierten Informationen in Echtzeit. Die überlagerten, virtuellen Inhalte fügen sich nahtlos in die vom Nutzer wahrgenommene Umgebung ein und haben einen direkten Bezug darauf. Im Unterschied zur AR bewegt sich der Nutzer bei Anwendungen aus dem Bereich der virtuellen Realität in einer vollständig vom Computer generierten, künstlichen Welt.

Eine Definition des Begriffs „augmented reality“ taucht bereits im Jahr 1992 [Caudell und Mizell 1992] in einer Veröffentlichung eines Projektes zur Unterstützung der Monteure beim Kabelverlegen in Flugzeugen bei Boeing auf. Milgram definierte unter dem Oberbegriff „Mixed Reality“ [Milgram und Kishino 1994] den Bereich zwischen der realen Welt und der rein virtuellen Umgebung (siehe Abb. 2.1, S. 7). Der Begriff „augmented virtuality“ definiert die Erweiterung einer virtuellen Um-

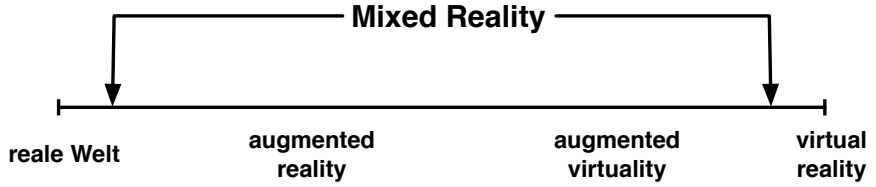


Abbildung 2.1: Milgrams „Reality-Virtuality Continuum“: Einordnung der einzelnen Definitionen, nach [Milgram und Kishino 1994]

gebung mit Repräsentationen echter Objekte, wie z. B. Livebilder von Personen in virtuellen Videokonferenzen [Regenbrecht u. a. 2003].

Ein AR-System (ARS) muss nach der allgemein anerkannten Definition von Azuma [Azuma 1997] folgende drei Bedingungen erfüllen:

- Es werden reale und virtuelle, vom Computer generierte Objekte kombiniert;

- das System ist interaktiv und operiert in Echtzeit;
- die virtuellen Inhalte überlagern die realen Objekte in der Umgebung des Anwenders.

Durch die Überlagerung in Echtzeit steht eine leicht verständliche Repräsentation von raumbezogenen Datenbeständen zur Verfügung, die sofort mit der direkten Umgebung verglichen und auf Plausibilität geprüft werden können. Der Beobachter stellt Verbindungen zwischen Realität und digitalen Daten her, die ohne den Einsatz eines AR-Systems möglicherweise unerkannt geblieben wären. Mit AR kann der

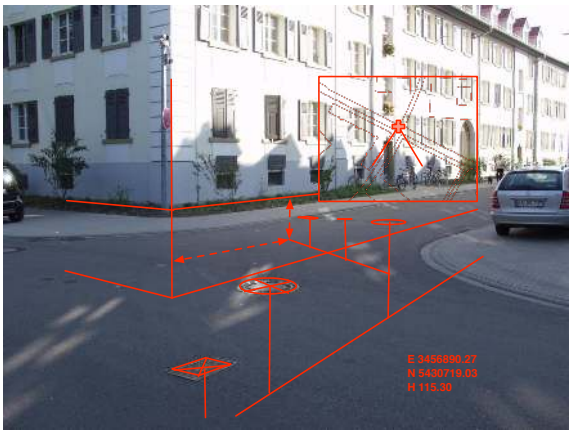


Abbildung 2.2: Simuliertes Beispiel eines AR-basierten Kanalinformationssystems

Blick auf verdeckte Strukturen und Objekte wie z. B. ein Kanalsystem „freigelegt“ werden, indem diese Daten mit der Sicht auf die Straßendecke überlagert werden (siehe Abb. 2.2, S. 8).

Zu einer der ersten Arbeiten im Bereich AR kann man die Arbeiten von Sutherland aus dem Jahr 1968 über Prototypen für Head-Mounted-Displays (z. B.) zählen [Sutherland 1968]. Erste Anwendungen außerhalb einer kontrollierten Laborumgebung wurden erst viel später entwickelt. Im „MARS“-Projekt (Mobile Augmented Reality Systems) wurde 1997 ein mobiles Outdoor-AR-System für die Fußgängernavigation entwickelt [Feiner u. a. 1997]. Pfeile und ortsbezogene Textmarken mit Informationen zu Gebäuden und Hörsälen sollten den Nutzer bei der Navigation auf einem Campusgelände unterstützen. Derartige mobile Systeme mussten damals auf einem Rucksacksystem getragen werden. Ein ähnliches System des IPF hatte etwa 10 kg Gewicht [Wursthorn, Coelho u. a. 2004]. Heute sind unzählige AR-Anwendungen für die Anzeige ortsbezogener Textmarken im Livebild einer typischen Smartphonekamera wie z. B. „Layar“ oder „Wikitude“ in den einschlägigen App-Stores erhältlich.

Wichtige Forschungsthemen innerhalb AR Themenbereiches umfassen die robuste Positionierung und Orientierung des Systems in Echtzeit [Reitmayr und Drummond

2006], Displaytechnologien wie Head-Mounted-Displays und Projektoren, Interaktionsmöglichkeiten, Datenaufnahme [Piekarski und Thomas 2002b; Sands u. a. 2004]. Anwendungen für die Erweiterte Realität findet man im Bereich der Medizin, Planung, im Katastrophenschutz [Leebmann 2004; Wursthorn, Coelho u. a. 2004], Marketing und Unterhaltung.

2.1 Anforderungen an AR-Systeme

AR ist nicht allein auf die Erweiterung der visuellen Wahrnehmung beschränkt, sondern umfasst alle Sinne. Im Bereich der Blindennavigation werden z. B. akustische Signale genutzt, um die natürliche Umgebung mit Zusatzinformationen zu erweitern. Die visuelle Wahrnehmung dominiert aber alle anderen Sinne und „überschreibt" alle anderen Sinneseindrücke. „Was wir sehen *ist* wahr" [R. B. Welch 1978]. Für die weiteren Betrachtungen in dieser Arbeit ist die Grundkomponente eines AR-Systems daher das Displaysystem. Integriert in einem typischen Smartphone oder Tablet-Computer erhält man ein AR-System bestehend aus einer Kamera, deren Livebild auf dem Display darstellbar ist. Sensoren zur Positionierung und Orientierung liefern die nötigen Informationen, um über geeignete Transformationen mit Hilfe von CPU und GPU 3D-Objekte in das Livebild der Kamera integrieren zu können (siehe Abb. 2.3, S. 11). Daten können über die drahtlose Netzwerkverbindung bei Bedarf nachgeladen werden.

Aus den Definitionen von Azuma [Azuma 1997] (s. o.) lassen sich die Anforderungen an AR-Systeme ableiten. Die Forderung nach Echtzeit bedeutet eine als flüssig wahrnehmbare Bildwiederholungsrate für die im Displaysystem dargestellten 3D-Objekte. Diese beginnt beim Menschen bei 15 Hz. In Kinofilmen werden 24 Hz oder gar 48 Hz (High Frequency Rate) eingesetzt. Typische Ein-Chip-System Kameras, wie sie in Smartphones oder Tablets eingebaut sind, erreichen oft 30 Hz. LC-Displays in PCs, Tablets und Smartphones verwenden meist Wiederholraten von 60 Hz. Die Zeit, die dem AR-System zum Messen von Position und Orientierung, zur Aufnahme eines Kamerabildes und zur Berechnung der Darstellung der 3D-Objekte zur Verfügung steht, beträgt somit im Bereich 15 Hz bis 60 Hz zwischen maximal 66,7 ms und minimal 16,7 ms. Die Systemverzögerung, die durch längere Berechnungszeiten entsteht, kann dann der größte Fehler in der Überlagerung von realen und virtuellen 3D-Objekten sein [Holloway 1995].

Ein Beispiel mit den technischen Daten eines realen Tablets soll dies verdeutlichen. Das Tablet habe ein 16 cm breites Display mit 1024×768 Bildpunkten. Die eingebaute Kamera habe einen horizontalen Öffnungswinkel von 44,8°. Dies entspricht 22,9 Pixel pro 1° Öffnungswinkel. Bei einem gleichförmigen Schwenk von 50° innerhalb einer Sekunde und einer Bildwiederholungsrate von 30 Hz entspricht die Bewegung 1,67° pro Bild oder, auf das Display bezogen, etwas mehr als 38 Bildpunkte. Bei einer Systemverzögerung von 33,3 ms würde das berechnete Bild der 3D-Objekte

vom Kamerabild um 38 Pixel in der Horizontalen abweichen. Die Abweichung der Überlagerung mit dem Kamerabild sollte nicht größer als drei bis vier Bildpunkte sein, um nicht als störend wahrgenommen zu werden. Um einen Bildpunkt in diesem Beispiel einhalten zu können, müsste die horizontale Orientierung auf 0,044° genau bekannt sein. Bei einem Objektabstand von 10 m müsste auch die Position besser als 1 cm genau bekannt sein, um eine Abweichung der Überlagerung von unter einem Bildpunkt einhalten zu können.

Die genaue Überlagerung von realen mit virtuellen 3D-Objekten stellt daher hohe Ansprüche an die Positionierung und Orientierung eines AR-Systems, die darüber hinaus auch in der Frequenz der Bildwiederholungsrate der Kamera zur Verfügung stehen muss.

2.2 Komponenten eines AR-Systems

Man kann die notwendige Komponenten eines AR-Systems in drei Gruppen aufteilen:

- Hardware zur Darstellung überlagerter Inhalte
- Raumbezogene Informationen, die überlagert werden sollen und Anwendungen
- Sensoren und Verfahren zur Positionierung und Orientierung, die eine Überlagerung der raumbezogenen Information mit der Umgebung des Anwenders erlauben.

Diese drei Gruppen werden im Folgenden besprochen.

2.2.1 Displaysysteme

Smartphone- oder Tabletdisplays hält der Nutzer einfach in sein Gesichtsfeld um die Überlagerungen mit den Bildern der Frontkamera sehen zu können. Um bei mobilen Systemen die Hände frei zu haben, können Displays direkt vor dem Auge des Nutzers getragen werden (Head-Mounted-Display). Bei videobasierten Displaysystemen („Video-See-through"-Displays) wird die Umgebung mit einer Videokamera erfasst. Der Computer kombiniert die Videobilder mit den Bildern, die er aus den virtuellen Inhalten passend zur Abbildungsvorschrift der Kamera generiert hat. Das Ergebnis wird auf den eingebauten LCD Displays angezeigt (siehe Abb. 2.3, S. 11), Das *ARvision-3D* System von *Trivisio* (siehe Abb. 2.4, S. 12) ist ein Stereo Video-See-through-Display mit zwei Kameras. Befestigungspunkte, Blickwinkel und Abstand der Kameras voneinander sind so gewählt, dass sie dem natürlichem Sehen des Menschen möglichst entsprechen. Abweichungen von den natürlichen Darstellungsparametern, wie zum Beispiel eine größere Basis bei Stereosystemen oder ungewöhnliche

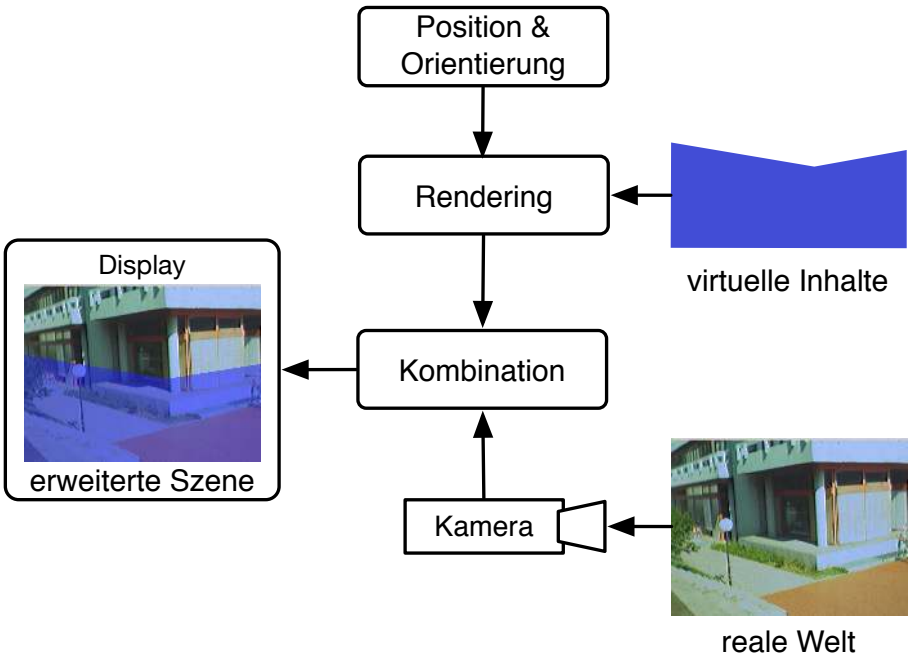


Abbildung 2.3: Videobasiertes AR: Daten werden in das Abbildungssystem der Kamera transformiert und mit der von der Kamera erfassten Umgebung überlagert. Das Ergebnis kann nur auf einem Display betrachtet werden [Wursthorn, Hering Coelho u. a. 2005]

Blickwinkel erschweren dem Benutzer die Orientierung und Fortbewegung im Gelände.

Durchsichtdisplays („See-through"-Displays) überlagern diese Inhalte über einen halbdurchlässigen Spiegel mit der Sicht des Nutzers von seiner Umgebung (siehe Abb. 2.5, S. 13). Das Displaysystem *AddVisor 150* der Firma *Saab* stellt auf diese Weise Bilder mit einer Auflösung von 1280×1024 Pixel pro Auge dar. Eine Sonderstellung nimmt das *Nomad* System von *MicroVision* ein (siehe Abb. 2.6, S. 14). Über einen halbdurchlässigen Spiegel projiziert ein Laserstrahl das Bild zeilenweise direkt auf die Netzhaut des Betrachters. Das Bild hat eine Auflösung von 800×600 Bildpunkten mit 32 Helligkeitsstufen bei einer Wiederholungsfrequenz von 60 Hz. Es deckt ein Gesichtsfeld von 23°×17° ab. Das entspricht etwa der Bildfläche eines 17 Zoll Monitors, die man in Entfernung einer Armlänge wahrnehmen würde. Gegenüber LCD-basierten Displaysystemen bietet es eine ausreichend helle Darstellung und hohen Kontrast im Gelände. Nachteile von Netzhautdisplays sind zum einen die geringe Akzeptanz bei den Nutzern, die beim Gedanken an einen Laserstrahl direkt auf ihre Netzhaut gesundheitliche Bedenken haben und zum anderen die monochrome Darstellung.

Durchsichtdisplays beeinträchtigen die natürliche Sicht des Nutzers durch die halbdurchlässigen Spiegel, die nur einen Teil des Lichtes durchlassen (z. B. Nomad:

Abbildung 2.4: Beispiel für ein Video-See-through-Display mit Stereokameras. ARvision-3D HMD-System von Trivisio. Quelle: `http://www.trivisio.de`

45 %) und durch Rahmen oder Halterungen der Spiegel, die das Gesichtsfeld etwas einschränken.

Weiterhin muss es einem Benutzer erlauben, sich sicher fortzubewegen. Mit einem Durchsichtdisplay bleibt die Sicht auf die Umgebung selbst bei einem Systemausfall im Gegensatz zu videobasierten Systemen erhalten. Außerdem kann die Umgebung bei videobasierten Systemen nur mit der Auflösung der Videokameras, bzw. der Displays, dargestellt werden. Beim ARvision System sind dies z. B. 800×600 Bildpunkte bei einem Gesichtsfeld von 32°×24° pro Kamera. Das entspricht einer Winkelauflösung von 0,04°, die im Vergleich zu der des Menschen (ca. 1") bedeutend schlechter ist. Mit derart geringen Auflösungen ist es nicht möglich Text auf einem Zettel oder einem Buch zu lesen. Verzögerungen zwischen der Aufnahme der Kamerabilder und der tatsächlichen Darstellung sind für den Nutzer ungewohnt.

Neben dem Hinzufügen von Objekten ist das Verdecken derzeit nur mit videobasierten Displays möglich, abgesehen von experimentellen Prototypen, wie z. B. in [Kiyokawa u. a. 2003] vorgestellt.

Bei Indoor-Anwendungen kann ein Projektor für Überlagerungen auf speziell vorbereitete Flächen oder normale Gegenstände zum Einsatz kommen [Bimber und Raskar 2005]. Mit mobilen Projektoren oder gar in Smartphones integrierte Projektoren kann auch ein mobiles System aufgebaut werden.

Die einzelnen Displayarten sind in Tabelle 2.1 gegenübergestellt und bezüglich einiger Eigenschaften bewertet.

Die Kalibrierung eines Displaysystems ermittelt die Abbildungsvorschrift vom Objekt zur Anzeige und die relativen Transformation zu Lage- und Orientierungssensoren [Leebmann 2003]. Das Problem, das in der Tabelle bewertet wird, bezieht sich dabei auf die Nutzerfreundlichkeit einer Displaykalibrierung. Im Falle eines See-through-Systems ist der Erfolg auch vom Willen des Nutzers abhängig. Es gestaltet sich schwierig eine einmal ermittelte Transformation Display-Auge bzw. Display-Pupille konstant zu halten. [Suthau 2006] löst dieses Problem mit einem integrierten Eye-Tracker, der die Lage der Pupille kontinuierlich verfolgt.

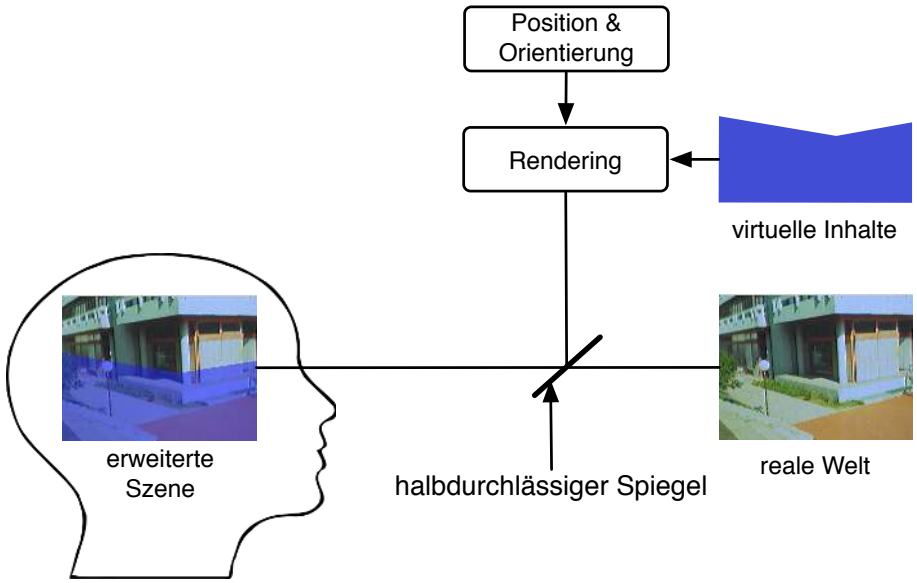


Abbildung 2.5: AR mit See-through-Displaysystem: Daten werden über einen halbdurchlässigen Spiegel in das Gesichtsfeld des Nutzers projiziert [Wursthorn, Hering Coelho u. a. 2005]

2.2.2 Positionierung und Orientierung

Für die Betrachtung der Möglichkeiten zur Positionierung und Orientierung bei ARS muss zwischen In- und Outdoor-Anwendungen unterschieden werden. Grundsätzlich kann eine Kombination unterschiedlicher Lösungen die Zuverlässigkeit und Robustheit erhöhen [G. Welch und Foxlin 2002].

Innerhalb geschlossener Räume können Trackingsysteme installiert werden, die je nach System bis zu sechs Freiheitsgrade in einer hohen Frequenz liefern. Im Freien kann die Systemumgebung in der Regel nicht mit Trackingsystemen präpariert werden.

GNSS (Global Navigation Satellite System) und IMU (Inertial Measurement Unit) sind in den heutigen Mobilgeräten verfügbar und liefern eine ausreichende Genauigkeit für Navigationslösungen. Für eine Überlagerung sind diese Sensoren aber zu ungenau (s. o.). Aus diesem Grund wird die eingebaute Kamera genutzt, um die Positionsschätzung zu verbessern [Daniel u. a. 2010; Reitmayr und Drummond 2006; D. Wagner u. a. 2008].

2.2.3 Raumbezogene Informationen

Bei einem Desktop-Geoinformationssystem hat der Anwender üblicherweise Steuerungsmöglichkeiten um innerhalb der Daten mit Operationen wie Zoom oder Pan zu navigieren. Bei AR wird sich der Anwender selbst zum Ort seines Interesses begeben.

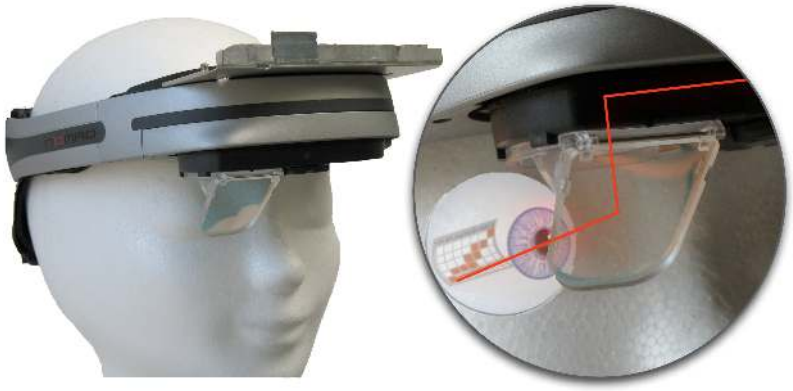


Abbildung 2.6: Monokulares Netzhautdisplay „Nomad“ von Microvision

Die Informationen seiner direkten Umgebung werden entsprechend der verwendeten Technologie mit der realen Ansicht dieser Umgebung überlagert.

Der Maßstab der raumbezogenen Information kann aus diesem Grund nicht beliebig klein sein. Ideal wäre ein Maßstab 1:1 entsprechend dem Maßstab der Umgebung, den auch der Anwender mit seinen eigenen Sinnen wahrnimmt. Die üblichen Daten raumbezogener Informationssysteme wie 3D-Stadtmodelle, Leitungskataster oder Touristeninformationen sind nicht für derartige Maßstäbe ausgelegt.

Für eine perfekte Überlagerung sind detaillierte Informationen über die 3D-Beschaffenheit des unmittelbaren Nahbereichs um den AR-Nutzer nötig. Damit können wichtige Fragen während des AR-Betriebs geklärt werden. Wie etwa die Frage nach Bezugsebenen auf oder an denen virtuelle Objekte abgelegt bzw. hinzugefügt werden können.

Coelho hat in [Wursthorn, Coelho u. a. 2004] ein ARS für den Hochwasserschutz beschrieben, mit dem ein virtueller, simulierter Hochwasserstand in das Sichtfeld einer Kamera vor Ort eingeblendet werden konnte. In der virtuellen Wasserebene waren allerdings keine Gebäude ausgespart. Aus diesem Grund mussten die Gebäudegeometrien in das ARS integriert werden, nur um die Verdeckungen der Wasseroberfläche zu erhalten.

[Piekarski und Thomas 2002b] nutzt eine am Kopf des Nutzers befestigte Kamera um die relative Lage der Hände mit daran angebrachten Markern bestimmen zu können. Damit ist es möglich, virtuelle Objekte interaktiv zu verändern. Es fehlt aber der Raumeindruck für die Hände selbst, da virtuelle Objekte nicht von ihnen verdeckt werden können.

	Videobasiert		**z. B. „See-through“**		**Projektor**
	Smartphone, Tablet	z. B. + Kamera	LCD basiert	Netzhaut	
Sicht der Umgebung	muss in Gesichtsfeld gehalten werden	nur durch Kameras; ungewohnt	fast ungestört; natürlich		normal
Outdoor	keine Hände frei	Abhängig von Kameraauflösung	schlechte Helligkeit	gut	schlechte Helligkeit
Kalibrierung	einmalig		vor jeder Nutzung		einmalig
Sonstiges	allgemein verfügbar	unbequem; Gewicht		Akzeptanz des LASERs	

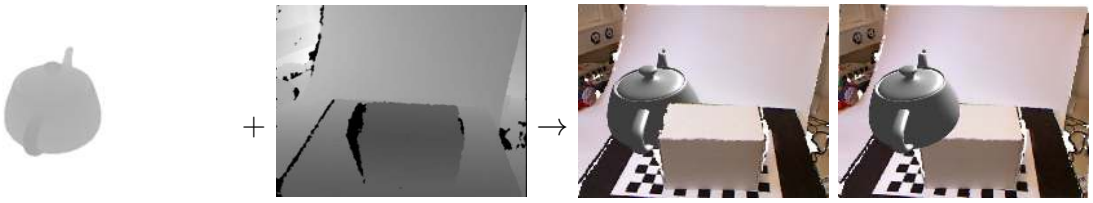
Tabelle 2.1: Vergleich der Darstellungsmöglichkeiten für mobile AR-Systeme

2.3 Tiefenbilder

Die Berücksichtigung von Verdeckungen ist wichtig. Wenn sich ein Gegenstand hinter einem anderen Gegenstand befindet, wird dieser normalerweise durch den vorderen Gegenstand verdeckt. Um diesen einfachen räumlichen Eindruck bei AR zu erhalten, muss der verdeckende Gegenstand im AR-System bekannt sein, damit die verbleibende Ansicht des hinteren Gegenstandes entsprechend erzeugt werden kann.

Es ist nicht in jeder AR-Anwendung erwünscht, die virtuellen Objekte zu verdecken bzw. diese möglichst realistisch in die Umgebung einzupassen, etwa durch zusätzliche Berücksichtigung des Schattenwurfes. Ohne Verdeckung kann aber der Raumeindruck bei der Betrachtung nicht unterstützt werden. Bei monokularen Displaysystemen kann der Raumeindruck nur durch eine zumindest räumlich realistische Integration in die Umgebung geschaffen werden.

[Benko u. a. 2012] verwenden eine Kinect Tiefenbildkamera, um die direkte Umgebung des Nutzers dreidimensional erfassen zu können. Mit Hilfe dieser 3D-Informationen werden sowohl die Verdeckungen durch Hände als auch Interaktion mit virtuellen Objekten möglich.



(a) Z-Buffer **(b)** Tiefenbild **(c)** $(a) + (b)$ **(d)** ohne

Abbildung 2.7: Verdeckung (c) virtueller Objekte (a) durch reale Objekte (b). (d): Überlagerung an gleicher Position im Bild ohne Verdeckungsberechnung

Stereo- oder Tiefenbildkameras können die 3D-Geometrie im Sichtfeld der AR-Anwendung in Echtzeit erfassen und so ad hoc die nötigen Informationen für eine Verdeckungsberechnung liefern (siehe Abb. 2.7, S. 16). So kann ein dichtes Tiefenbild direkt mit dem Z-Buffer des Grafiksystems verknüpft werden [Souma u. a. 2012] um 3D-Objekte der Computergrafik direkt von echten Objekten verdecken zu lassen. Auf diese Weise kann nicht nur der Eindruck einer natlosen Überlagerung im Nahbereich verbessert werden, sondern überhaupt erst die gegenseitige räumliche Anordnung virtueller und realer Objekte zueinander realisiert werden. In Abb. 2.7 wird bspw. die virtuelle Teekanne (2.7a) durch die reale Schachtel verdeckt (2.7c). Damit entsteht der gewünschte Eindruck, dass die Kanne hinter der Schachtel steht. Bei einer einfachen Überlagerung virtueller Inhalte in das Bild der realen Szene ohne Berücksichtigung der Tiefe steht die Kanne scheinbar auf der Schachtel (2.7d).

Für die weitere Nutzung in einer AR-Anwendung ist es aber auch von Vorteil, Flächen in Echtzeit zu extrahieren, wie in [Holz u. a. 2012]. Auf diese Weise kann bspw. ein virtueller Ball eine echte Ebene herunterrollen oder die virtuelle Teekanne auf einer realen Ebene abgestellt werden.

Da die Positionierung und Orientierung durch vorhandene GNSS und IMU oft unzureichend ist, muss das Kamerasystem, die Tiefenbildkamera oder eine Kombination von beiden für die Bewegungsschätzung mit genutzt werden.

Die grundsätzlichen Möglichkeiten der Bewegungsberechnung mit einem Stereokamerasystem werden im nächsten Kapitel behandelt (Kap. 3). Auf die beiden Kamerasysteme (Stereokamera und Tiefenbildkamera) wird in Kapitel 4 eingegangen.

3

Mathematische Grundlagen der Bewegungsberechnung

In diesem Kapitel werden mögliche Verfahren zur Berechnung der Eigenbewegung einer Stereokamera beschrieben. Die Berechnung der Eigenbewegung eines Kamerasystems wird allgemein unter dem Begriff *Visual Odometry* (VO) zusammengefasst. David Nistér soll 2004 als Erster den Begriff in seiner gleichnamigen Veröffentlichung [Nistér u. a. 2004] geprägt haben [Scaramuzza und Fraundorfer 2011]. Der Begriff ist von der Wegmessung eines Fahrzeuges oder Roboters, z. B. aus der fortlaufenden Zählung der Radumdrehungen, abgeleitet. An Stelle von Radumdrehungen oder Ähnlichem werden bei der visuellen Odometrie die relativen Eigenbewegungen des Kamerasystems aus Kamerabildfolgen abgeleitet.

Wird ein Stereokamerasystem verwendet, spricht man von *Stereo Visual Odometry* oder auch *Stereo Egomotion*. Allgemein ermöglichen odometrische Verfahren nur relative Positionsbestimmungen, da sie mit Bewegungsdifferenzen zwischen zwei Messzeitpunkten arbeiten. Soll die zurückgelegte Strecke als Trajektorie vorliegen, müssen die einzelnen, relativen Positionsänderungen nacheinander akkumuliert werden. Damit wird auch der Fehler akkumuliert. Mit zunehmender Laufzeit wird somit auch die Abweichung vom tatsächlich zurückgelegten Weg größer. Es kommt zu einer Drift.

SLAM steht für Simultanious Location and Mapping. Dabei soll meist ein mobiler Roboter gleichzeitig seine ihm noch unbekannte Umgebung kartieren und seine eigene Position und Orientierung in dieser Karte schätzen [Siciliano und Khatib 2008]. SLAM ist eine komplexere Aufgabe als VO, da hier die Merkmale der Umgebung gespeichert werden müssen. Beim wiederholten Aufsuchen bereits vermessener Bereiche sollten diese wieder als solche erkannt und die Schätzung der gemessenen Umgebungspunkte mit Hilfe der aktuellen und den bereits vorhandenen Messungen erneuert werden. Durch das wiederholte Aufsuchen erzeugte Schleifen können die Drift der resultierenden Fahrzeugtrajektorie deutlich reduzieren [Paz u. a. 2008].

Eine Stereokamera ist eine Kameraanordnung mit zwei identischen Kameras, die über eine feste, bekannte Basis verbunden sind und deren gegenseitige Raumlage über die relative Orientierung beschrieben wird. Die absolute Orientierung eines Stereomodells in der Photogrammetrie beschreibt die Stereokonfiguration einer Kamera mit zwei Aufnahmestandpunkten. Die Basis ist hier nicht fest und damit nicht vorab bekannt. Es wird neben der Lage zum geodätischen Bezugssystem zusätzlich noch ein Maßstab benötigt.

Die äußere Orientierung einer Kamera im Raum beschreibt deren Lage und Position im übergeordneten Koordinatensystem (auch Kamera *Pose* in der Computer Vision). Als euklidische Bewegung $\boldsymbol{M}_{t,t+1}$ (M für *motion*) des Stereokamerasystems wird die formtreue Transformation bestehend aus Rotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$ verstanden, die zwischen zwei benachbarten Aufnahmezeitpunkten t und $t+1$ stattfindet. $\boldsymbol{M}$ ist eine homogene 4×4 Matrix:

$$\boldsymbol{M}_{t,t+1} = \begin{bmatrix} \boldsymbol{R}_{t,t+1} & \boldsymbol{T}_{t,t+1} \\ 0^T & 1 \end{bmatrix}, \tag{3.1}$$

wobei $\boldsymbol{R}_{t,t+1} \in SO(3)$ die 3×3 Rotationsmatrix und $\boldsymbol{t}_{t,t+1}$ der 3×1 Translationsvektor zwischen den Aufnahmezeitpunkten sind. Auf homogene Koordinaten wird weiter unten in Abschn. 3.1.1 eingegangen.

Aus einer Stereobildsequenz mit n Aufnahmezeitpunkten kann die Trajektorie des Stereokamerasystems über die Akkumulation der Bewegungen $\boldsymbol{M}_{1,\ldots,n}$, beginnend mit einer bekannten Kamerapose zum Startpunkt $t = 0$, berechnet werden.

Eine *Bildsequenz* ist in dieser Arbeit als eine Folge von Bildern der selben Kamera, die mit einer konstanten Frequenz aufgenommen werden, definiert. Die Aufnahmefrequenz wird entweder in Hz oder FPS (*frames per second*) angegeben.

Eine *Stereobildsequenz* besteht aus zwei Bildsequenzen - jeweils für linke und rechte Kamera, wobei die Aufnahmefrequenz beider Kameras identisch ist und das Stereobildpaar zeitgleich aufgenommen wird. Die relative Orientierung der Stereobildpaare einer Stereobildsequenz wird als konstant angenommen. Beide Kameras sind so auf einer festen Basis montiert, dass deren relative Orientierung vorab, zusammen mit der inneren Orientierung der linken und rechten Kamera, bestimmt werden kann. Spätestens zum Zeitpunkt der Auswertung der Stereobildsequenz liegt das Stereobildpaar als Epipolarbildpaar vor.

In dieser Arbeit wird der Begriff Stereobildpaar synonym zum Begriff Epipolarbildpaar verwendet, wenn es nicht anders angegeben wird.

Im Kontrast zur Bildsequenz steht die ungeordnete *Bildsammlung* (siehe Gegenüberstellung in Tab. 3.1), wie man sie z. B. für die Objektrekonstruktion in der Photogrammetrie benötigt. Hier werden mehrere konvergente Aufnahmen eines Objektes benötigt, die aber nicht zwingend in konstanten Zeitschritten aufgenommen werden müssen. Es können auch verschiedene Kameras zum Einsatz kommen. Für die automatische Auswertung müssen die inneren- und äußeren Orientierungen aller

	Bildsequenz	Stereobildsequenz	Bildsammlung
Anzahl Kameras	1	2	1... Anzahl Bilder
Zeitliche Folge	konstant, bekannt		ungeordnet
Weitere Bedingungen	-	Epipolarbildpaar	-

Tabelle 3.1: Gegenüberstellung der Definitionen für die Begriffe *Bildsequenz*, *Stereobildsequenz* und *Bildsammlung*

Aufnahmen bekannt sein. Dazu muss zunächst eine ausreichende Menge homologer Merkmale in den Bildern gefunden werden. Die Strategien zum Auffinden homologer Merkmale in Bildsammlungen unterscheiden sich von denen einer Bildsequenz, bei der die bekannte zeitliche Abfolge gewisse Einschränkungen in der zu erwartenden Bewegung der Merkmale im Bild erlaubt. Eine Bildsequenz ist eine Untermenge bzw. Spezialisierung einer Bildsammlung.

Zur rein bildbasierten Berechnung der Bewegung eines Stereokamerasystems im Raum stehen vier Bilder zur Verfügung, je ein Stereobildpaar pro Zeitpunkt. Damit die einzelnen Bilder miteinander in Beziehung gebracht werden können, müssen in ihnen zunächst homologe Merkmale extrahiert werden. Die Arbeitsschritte zur automatischen Merkmalsextraktion und -Verfolgung in digitalen Bildsequenzen werden in Kapitel 5 diskutiert. In Abb. 3.1 ist die Aufnahmeanordnung skizziert: die festen 3D-Objektpunkte $X_{1...3}$ werden jeweils auf dem linken und rechten Bild des sich bewegenden Stereokamerasystems zu den beiden Aufnahmezeitpunkten t und $t+1$ als 2D-Bildpunkte x' abgebildet. In diesem Kapitel seien die 2D-3D-Punktkorrespondenzen, die 2D-Punktkorrespondenzen zwischen den einzelnen Bildern sowie die innere und relative Orientierung der Kameras bekannt. Es muss aber mit einem gewissen Anteil von Ausreißern in Form von falschen 2D-3D-Zuordnungen und nicht-homologen Bildmerkmalen gerechnet werden.

Je nach Verwendung der vier Bilder stehen verschiedene Möglichkeiten zur Bewegungsberechnung zur Verfügung, die im Folgenden beschrieben werden. Eine gute Übersicht zu Visual Odometry im Allgemeinen wird in einem Tutorial von Davide Scaramuzza gegeben [Scaramuzza und Fraundorfer 2011]. Der grobe Ablauf eines Stereo Odometrie Verfahrens ist in Abb. 3.2 dargestellt:

Zentrales Element sind die homologen Merkmale zur Bewegungsberechnung. Es gibt drei Arten dieser Merkmale:

- **2D-Punktkorrespondenzen** in Bildkoordinaten zwischen zwei Aufnahmezeitpunkten (Abschnitt 3.3).

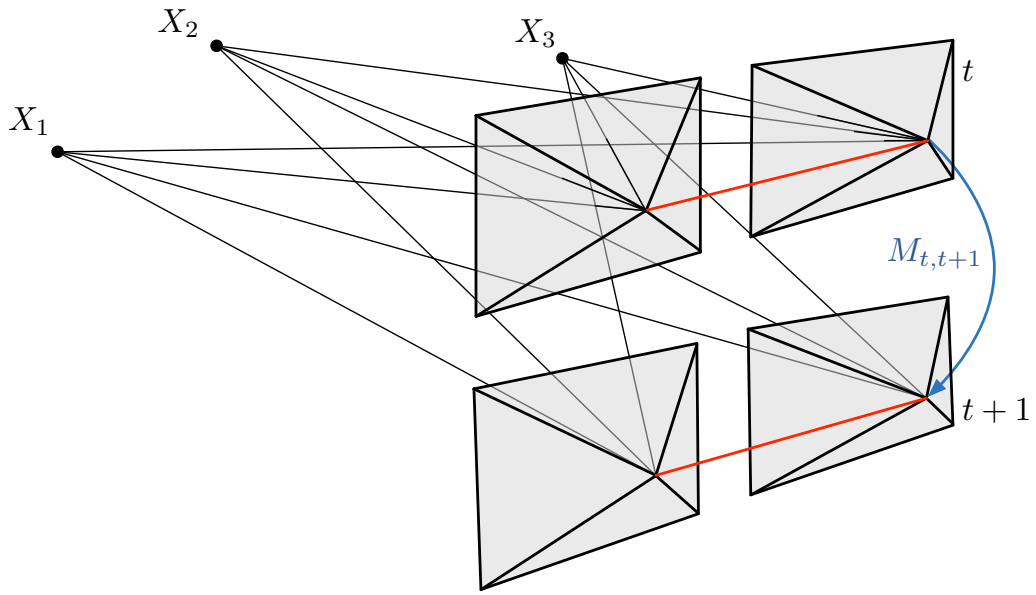


Abbildung 3.1: Zwei Zeitpunkte t und $t+1$ eines bewegten Stereokamerasystems. Linke und rechte Kamera sind über eine feste Basis verbunden. Die 3D-Objektpunkte X_n werden in den vier Bildern abgebildet. Mit bekannten Punktkorrespondenzen soll die Bewegung $M_{t,t+1}$ zwischen den beiden Aufnahmezeitpunkten ermittelt werden

- **3D-Punktkorrespondenzen** aus einer Triangulation der 2D-Punktkorrespondenzen in den aufeinander folgenden Epipolarbildpaaren (Abschnitt 3.4.1).
- **2D-3D-Punktkorrespondenzen**: neben den triangulierten 3D-Punkten werden auch die dazugehörigen 2D-Bildpunkte verwendet (Abschnitt 3.4.2).

Die Art der verwendeten Punktkorrespondenzen stellt ein wesentliches Unterscheidungsmerkmal von Stereo-VO-Verfahren dar. Eine Auswahl von Verfahren wird später, in Kapitel 7, in einem gemeinsamen Testrahmen gegeneinander verglichen.

Vor der Betrachtung der Bewegungsberechnung wird kurz auf die Aufnahmemodelle von monokularen Kameras und Stereokameras eingegangen. Dabei werden nur jeweils die relevanten Eigenschaften für dieses Kapitel behandelt. In Kapitel 4, Aufnahmesysteme werden diese dann ausführlicher besprochen.

Die Grundlagen dieses Kapitels entstammen den Standardwerken [Faugeras und Luong 2001; R. Hartley und Zisserman 2004; McGlone 2004] und den darin enthalten Quellen, wenn nicht anders angegeben.

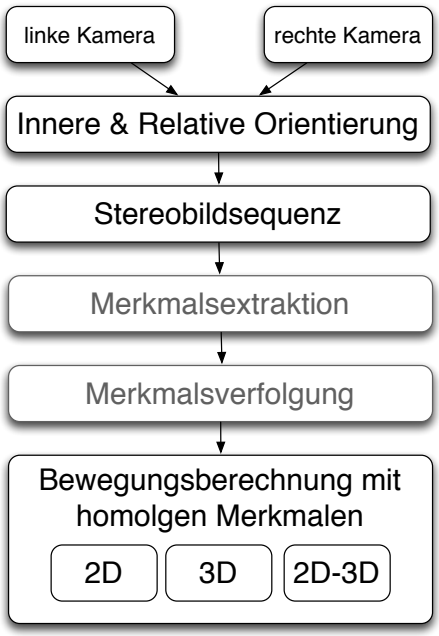


Abbildung 3.2: Allgemeiner Ablauf zur Berechnung der Eigenbewegung. Merkmalsextraktion und Merkmalsverfolgung werden in Kap. 5 behandelt

3.1 Aufnahmemodell einer Lochkamera

Die Abbildung eines 3D-Objektpunktes $\boldsymbol{X}$ auf dem Bildsensor als 2D-Bildpunkt x' der Kamera wird in der Photogrammetrie vollständig durch die Parameter der äußeren und inneren Orientierung beschrieben. Die äußere Orientierung einer Kamera bestimmt die räumliche Lage des Projektionszentrums bezogen auf das Objektkoordinatensystem und wird durch sechs Parameter definiert: drei Rotations- (ω, ϕ, κ) und drei Translationsparameter (X_0, Y_0, Z_0). Die innere Orientierung beschreibt alle Parameter, die für die Rekonstruktion des Strahlengangs vom gemessenen Bildpunkt $\boldsymbol{x'} = (x', y')^T$ in Richtung des Objektpunktes $\boldsymbol{X} = (X, Y, Z)^T$ benötigt werden.

Die Lochkamera beschreibt eine ideale Kamera bei der die 3D-Objektpunkte durch eine Zentralprojektion als 2D-Bildpunkte abgebildet werden. Das Abbildungsmodell ist eine perspektivische Abbildung, bei der alle Projektionsstrahlen durch ein gemeinsames Projektionszentrum $\boldsymbol{X_0}$ gehen. Geraden im Objektraum werden auch als Geraden im Bildraum abgebildet (Kollinearität).

Wird das Bildkoordinatensystem der Kamera durch ein euklidisches Koordinatensystem festgelegt, ist die Lochkamera durch die Angabe des Abstandes vom Projektionszentrum zur Bildebene mit der Kamerakonstanten c und der Lage des Bildhauptpunktes $H(x'_0, y'_0)$ vollständig beschrieben.

Der Zusammenhang zwischen Objekt- und Bildkoordinaten wird über die Kollinearitätsgleichungen hergestellt. Die Abbildung eines Bildpunktes $\boldsymbol{x'}$ in den Objektraum $\boldsymbol{X}$ lautet

$$\boldsymbol{X} = \boldsymbol{X_0} + m \cdot \boldsymbol{R} \cdot \boldsymbol{x'}, \tag{3.2}$$

wobei m der Maßstabsfaktor ist. Mit der Einführung des Bildhauptpunktes (x'_0, y'_0) und Auflösung nach den Bildkoordinaten ergeben sich die Kollinearitätsgleichungen (mit $z' = c$):

$$x' = x'_0 + z' \cdot \frac{r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)} \tag{3.3}$$

$$y' = y'_0 + z' \cdot \frac{r_{12} \cdot (X - X_0) + r_{22} \cdot (Y - Y_0) + r_{32} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)}. \tag{3.4}$$

3.1.1 Homogene Koordinaten

Homogene Koordinaten vereinfachen die Darstellung räumlicher Beziehungen. Zum einen lassen sich alle geradentreuen Abbildungen als Matrix-Vektor-Multiplikation darstellen, so dass Verkettungen oder Inversion solcher Abbildungen unmittelbar angegeben werden können. Zum Anderen können Verknüpfungen räumlicher Elemente als Bilinearformen dargestellt werden wodurch die Formulierung von Bedingungen stark vereinfacht wird [Förstner 2000]. Vor allem aber können damit auch Punkte im Unendlichen, wenn diese z. B. an Merkmalen am Horizont detektiert werden, repräsentiert werden. Diese Eigenschaften sind bei der Modellierung des Abbildungsprozesses in der Kamera von Vorteil.

Homogene Koordinaten entstehen aus euklidischen Koordinaten $(x, y)^T$ durch hinzufügen einer weiteren Koordinate w (*homogene Koordinate*) und einer freien Skalierung λ:

$$\boldsymbol{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ mit } \quad |\boldsymbol{x}^2| = u^2 + v^2 + w^2 \neq 0. \tag{3.5}$$

Die euklidischen Koordinaten können mit $x = \frac{u}{w}$ und $y = \frac{v}{w}$ für $w \neq 0$ berechnet werden. Für $w = 0$ erhält man einen Punkt $\boldsymbol{x_\infty} = [u, v, 0]^T$ in unendlicher Entfernung. Ein euklidischer Punkt $[x, y, z]^T$ im $\mathbb{R}^3$ wird in homogenen Koordinaten als 4-Vektor dargestellt und ist somit überparametrisiert.

3.1.2 Das projektive Abbildungsmodell

Die allgemeinste lineare Transformation homogener Koordinaten ist die projektive Abbildung. Für einen n-dimensionalen Punkt $\boldsymbol{X}$ gilt:

$$\boldsymbol{x'} = \boldsymbol{H}\boldsymbol{X}, \tag{3.6}$$

Raum	**DOF**	$\boldsymbol{T}$	**Invarianten**			
			Strecken	Längenverhältnisse, Winkel	Teilverhältnisse, Parallelität	Inzidenz, Doppelverhältniss
euklidisch	6	$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix}$	✓	✓	✓	✓
metrisch	7	$\begin{bmatrix} \sigma\boldsymbol{R} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix}$		✓	✓	✓
affin	12	$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix}$			✓	✓
projektiv	15	$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{v^T} & \lambda \end{bmatrix}$				✓

Tabelle 3.2: Hierarchie der Transformationen und deren Invarianten. Nach [Schreer 2005; Pollefeys 1999; Faugeras und Luong 2001]. DOF=Freiheitsgrade

wobei $\boldsymbol{H}$ (eng. *homography* für projektive Abbildung) eine nicht singuläre $(n+1) \times (n+1)$-Matrix im projektiven Raum $\mathbb{P}^n$ ist. Es gilt $\boldsymbol{H} = \lambda\boldsymbol{H}$, da eine Skalierung $\boldsymbol{x'}$ ändern würde, nicht aber den zu Transformierenden Punkt. $\boldsymbol{H}$ ist daher homogen. Eine Projektive Transformation hat im $\mathbb{P}^2$ acht Freiheitsgrade, im $\mathbb{P}^3$ sind es 15 Freiheitsgrade [R. Hartley und Zisserman 2004] und ist bis auf einen Skalierungsfaktor definiert. Durch die steigende Verallgemeinerung der Transformationsvorschrift von der euklidischen Transformation über die affine Transformation zur projektiven Transformation erhöht sich die Anzahl der möglichen Freiheitsgrade. Diese Hierarchie der räumlichen Transformationen ist in Tab. 3.2 zusammengefasst. Darin ist $\boldsymbol{R}$ eine Rotationsmatrix, $\boldsymbol{A}$ eine 3×3 invertierbare Matrix mit acht Freiheitsgraden, $\boldsymbol{t}$ ein Translationsvektor, $\boldsymbol{v}$ ein allgemeiner 3-Vektor. σ und λ sind Skalare.

Der Nachweis der Kollinearität der projektiven Abbildung ist laut [R. Hartley und Zisserman 2004] folgendermaßen gegeben: Es seinen $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ Punkte auf einer Geraden $\boldsymbol{l}$, d.h. es gilt $\boldsymbol{l}^T\boldsymbol{x}_i = 0$ für $i = 1 \ldots 3$. Sei $\boldsymbol{H}$ eine nicht singuläre 3×3 Matrix, so dass $\boldsymbol{l}^T\boldsymbol{H}^{-1}\boldsymbol{H}\boldsymbol{x}_i = 0$. Daraus folgt, dass die Punkte $\boldsymbol{H}\boldsymbol{x}_i$ alle auf der Geraden $\boldsymbol{H}^T\boldsymbol{l}$ liegen und die Kollinearität bei der Transformation erhalten bleibt.

Die Umkehr der Aussage, dass jede projektive Abbildung (auch Projektivität) geradentreu ist, führt zum Hauptsatz der projektiven Geometrie: Eine eindeutige Abbildung des $\mathbb{P}^n$, die Geraden in Geraden abbildet, ist für $n > 2$ eine Projektivität. Sie lässt sich in homogenen Koordinaten als $\boldsymbol{x'} = \boldsymbol{H}\boldsymbol{x}$ darstellen.

Eine Gerade $\boldsymbol{l}$ in der Ebene wird in der impliziten Form durch die homogenen Koordinaten a, b und c dargestellt:

$$\boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \text{ mit } \quad |\boldsymbol{l}^2| = a^2 + b^2 + c^2 \neq 0. \tag{3.7}$$

Die Komponenten a und b definieren den Normalenvektor, damit hat die Gerade die Richtung $[b, -a]^T$ mit einem Winkel $\phi = \text{atan}(\frac{-a}{b})$. Die ideale Gerade $\boldsymbol{l_\infty} = [0, 0, 1]^T$ im Unendlichen stellt den Horizont dar. Die idealen Punkte $\boldsymbol{x_\infty} = [x, y, 0]^T$ liegen auf dem Horizont $\boldsymbol{l_\infty}$.

Ein Punkt liegt in $\mathbb{P}^2$ auf einer Geraden bzw. ein Gerade geht durch einen Punkt, wenn die Inzidenz-Bedingung

$$\boldsymbol{x} \cdot \boldsymbol{l} = \boldsymbol{x}^T \boldsymbol{l} = \boldsymbol{l}^T \boldsymbol{x} = ax + by + cw = 0 \tag{3.8}$$

gilt. Für einen idealen Punkt auf der idealen Linie sieht man z. B. , dass $[0, 0, 1][x, y, 0]^T = 0$ erfüllt ist. Eine Gerade $\boldsymbol{l}$ durch zwei Punkte $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ ist durch das Kreuzprodukt $\boldsymbol{l} = \boldsymbol{x_1} \times \boldsymbol{x_2}$ definiert. Der Schnittpunkt $\boldsymbol{x}$ zwischen zwei Geraden wiederum ergibt sich aus deren Kreuzprodukt: $\boldsymbol{x} = \boldsymbol{l_1} \times \boldsymbol{l_2}$. Drei Geraden schneiden sich in genau einem Punkt, wenn $\det([\boldsymbol{l_1}\ \boldsymbol{l_2}\ \boldsymbol{l_3}]) = 0$ ist und drei Punkte sind kollinear, wenn $\det([\boldsymbol{x_1}\ \boldsymbol{x_2}\ \boldsymbol{x_3}]) = 0$ gilt. Diese gemeinsamen Eigenschaften von Punkt und Linie im projektiven Raum wird Dualität genannt. Da Strahlenbüschel (engl. *pencil of lines*) wegen des Prinzipes der Dualität einem Satz kollinearer Punkte entsprechen, folgt daraus, dass es sich bei Strahlenbüschel, genau wie bei kollinearen Punkten, um einen eindimensionalen projektiven Raum handelt. Die Dualität wurde von Paul C. V. Hough in der Hough-Transformation erstmals zur Linienerkennung aus Punktmengen eingesetzt [Schreer 2005]. Im $\mathbb{P}^3$ gilt die Dualität für Punkt $\boldsymbol{x}$ und Ebene $\boldsymbol{\pi}$ analog zum zweidimensionalen Fall.

Winkel und Längen sind in der projektiven Abbildung nicht mehr vergleichbar. Kreise werden zu allgemeinen Kegelschnitten [R. Hartley und Zisserman 2004]. Dabei kann nicht entschieden werden, ob ein Kegelschnitt eine Hyperbel, eine Parabel oder eine Ellipse ist. Parallele Linien schneiden sich in einem Punkt im Unendlichen. Die Inzidenz von Punkt und Gerade, die Kollinearität von Punktetripeln und das Doppelverhältnis von vier kollinearen Punkten bleiben aber erhalten. Das Doppelverhältnis ist wie folgt definiert [R. Hartley und Zisserman 2004]:

$$\text{Doppelverhältnis}(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}) = \frac{|\boldsymbol{x_1 x_3}| \cdot |\boldsymbol{x_2 x_4}|}{|\boldsymbol{x_1 x_4}| \cdot |\boldsymbol{x_2 x_3}|}, \tag{3.9}$$

wobei $|\boldsymbol{x_i x_j}|$ der geometrische Abstand der Punkte ist. Tab. 3.2 zeigt eine Übersicht der Invarianten der einzelnen räumlichen Transformationen.

Die Gesetzmäßigkeiten der perspektivischen Abbildung einer Zentralprojektion können mit der projektiven Geometrie mathematisch beschrieben werden, wobei die

homogenen Koordinaten wesentlicher Bestandteil des projektiven Raumes sind. Die projektive Geometrie führt dabei zu linearen Beziehungen und vereinfacht so die Schätzung und Fehlerfortpflanzung [Förstner 2004a].

Für die perspektivische Abbildung eines Objektpunktes $\boldsymbol{X}_k = (X, Y, Z, 1)^T$ auf die Bildebene (als Bildpunkt $\boldsymbol{x}'$) einer idealen Kamera (= Lochkamera, s. o.) gilt in homogenen Koordinaten:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.10}$$

oder kurz mit der homogenen 3×4 Projektionsmatrix $\boldsymbol{P}$

$$\boldsymbol{x'} = \boldsymbol{P} \cdot \boldsymbol{X}_k. \tag{3.11}$$

Mit der Einführung des Bildhauptpunktes $H(x'_0, y'_0)$ ergeben sich für die Bildkoordinaten $x' = c \cdot \frac{X}{Z} + x'_0$ bzw. $y' = c \cdot \frac{Y}{Z} + y'_0$. Ausgedrückt in homogenen Koordinaten:

$$\boldsymbol{x}' = \begin{bmatrix} c & 0 & x'_0 & 0 \\ 0 & c & y'_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.12}$$

und mit der Einführung der Kalibriermatrix $\boldsymbol{K}$

$$\boldsymbol{K} = \begin{bmatrix} c & 0 & x'_0 \\ 0 & c & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

ergibt sich die Abbildungsvorschrift zu

$$\boldsymbol{x}' = \boldsymbol{K}[\boldsymbol{I}|0]\boldsymbol{X}_k = \begin{bmatrix} c & 0 & x'_0 \\ 0 & c & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{X}_k, \tag{3.14}$$

wobei $\boldsymbol{I}$ = Einheitsmatrix.

Da der Raumpunkt üblicherweise in einem vom Kamerasystem unterschiedlichen Koordinatensystem vorliegt, kommt die äußere Orientierung in Form einer Translation mit $\boldsymbol{X_0}$ und einer Rotation mit $\boldsymbol{R}$ hinzu, so dass sich die Abbildungsvorschrift ergibt zu:

$$\boldsymbol{x}' = \begin{bmatrix} c & 0 & x'_0 \\ 0 & c & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & -\boldsymbol{X}_0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.15}$$

bzw.

$$\boldsymbol{x}' = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{X}_0]\boldsymbol{X}. \tag{3.16}$$

Die Abbildungsvorschrift der Lochkamera (3.16) hat neun Freiheitsgrade: drei (c, x'_0, y'_0) in der Kalibriermatrix $\boldsymbol{K}$ für die innere Orientierung, sechs für die äußere Orientierung ($\boldsymbol{R}$ und $\boldsymbol{X}_0$).

Der euklidische Abstand d zwischen beobachteten Bildkoordinaten $\boldsymbol{x}'_i$ und verbesserten Bildkoordinaten $\hat{\boldsymbol{x}}_i{}' = \hat{\boldsymbol{P}}\boldsymbol{X}_i$ wird als Maß für den Rückprojektionsfehler (rpe) genutzt:

$$\text{rpe} = \sum_i d(\hat{\boldsymbol{x}}_i{}', \boldsymbol{x}'_i)^2. \tag{3.17}$$

3.1.3 Zerlegung der P-Matrix

Die Projektionsmatrix $\boldsymbol{P}$ beschreibt den Abbildungsvorgang von Objektpunkten in den Bildraum. Aus ihr kann der Ort des Projektionszentrums $\boldsymbol{X_0}$, die Rotationsmatrix $\boldsymbol{R}$ und die Kameramatrix $\boldsymbol{K}$ abgeleitet werden [R. Hartley und Zisserman 2004], wenn nur $\boldsymbol{P}$ selbst bekannt ist.

Das Projektionszentrum $\boldsymbol{X_0} = (x, y, z, t)^T$ ergibt sich algebraisch aus

$$\boldsymbol{x} = \det([\boldsymbol{p_1}, \boldsymbol{p_3}, \boldsymbol{p_4}]) \quad \boldsymbol{y} = -\det([\boldsymbol{p_1}, \boldsymbol{p_3}, \boldsymbol{p_4}]) \tag{3.18}$$

$$\boldsymbol{z} = \det([\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_4}]) \quad \boldsymbol{t} = -\det([\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}]), \tag{3.19}$$

wobei $\boldsymbol{p_i}$ die Spaltenvektoren von $\boldsymbol{P}$ sind. Für das Projektionszentrum gilt $\boldsymbol{P}\boldsymbol{X_0} = 0$. Dieses lineare, homogene Gleichungssystem kann mit einer Singulärwertzerlegung (SVD) von $\boldsymbol{P}$ gelöst werden, wenn die $\boldsymbol{P}$-Matrix (3×4) um eine Zeile mit Nullen erweitert wird.

Um aus $\boldsymbol{P} = [\boldsymbol{M}|\boldsymbol{t}]$ die Rotationsmatrix $\boldsymbol{R}$ und die Kalibriermatrix $\boldsymbol{K}$ zu berechnen muss eine RQ-Zerlegung [R. Hartley und Zisserman 2004] der Teilmatrix $\boldsymbol{M} = \boldsymbol{K}\boldsymbol{R}$ durchgeführt werden. Eine nicht-singuläre Matrix kann als Produkt einer oberen Dreiecksmatrix $\boldsymbol{K}$ (Kalibriermatrix) und einer Orthogonalmatrix $\boldsymbol{R}$ (Rotationsmatrix) faktorisiert werden.

3.2 Aufnahmemodell eines Stereokamerasystems

Eine Stereokamera ist eine Kameraanordnung mit zwei Kameras, die über eine feste Basis verbunden sind. Dabei werden bezüglich der räumlichen Anordnung der Kameras zwei Klassen von Stereokamerasystemen unterschieden. Bei einem achsparallelen System sind die optischen Achsen der Kameras parallel ausgerichtet, bei einem allgemeinen Stereosystem schneiden sich die optischen Achsen in einem Konvergenzpunkt.

Die Epipolargeometrie beschreibt die allgemeine Beziehung zweier konvergenter, zueinander verschobener Aufnahmen, die nicht über eine feste Basis miteinander verbunden sein müssen. Die Epipolargeometrie ist allein von den inneren Orientierungen der Kameras und deren relativer Orientierung zueinander abhängig. Die

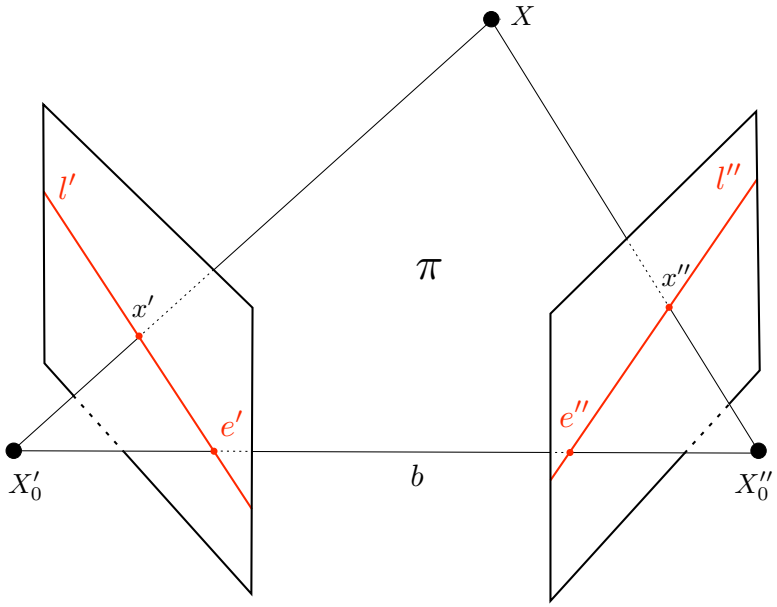


Abbildung 3.3: Epipolargeometrie eines Bildpaares

Epipolarebene $\boldsymbol{\pi}$ (auch Kernebene) wird durch den Objektpunkt $\boldsymbol{X}$ und die beiden Projektionszentren $\boldsymbol{X'_0}$ und $\boldsymbol{X''_0}$ definiert. In der Epipolarebene liegen dann auch die Bildpunkte $\boldsymbol{x'}$ und $\boldsymbol{x''}$ (siehe Abb. 3.3, S. 27). Die Basislinie $b \in \boldsymbol{\pi}$ verbindet die beiden Kamerazentren miteinander. Alle Epipolarebenen schneiden sich entlang der Basislinie (*pencil of planes*). Die Schnittpunkte der Basislinie mit den Bildebenen sind die Epipole (Kernpunkte) e' und e'' mit [Faugeras und Luong 2001]:

$$e' = \boldsymbol{K'} \boldsymbol{R}^T t, \quad e'' = \boldsymbol{K''} t. \tag{3.20}$$

Die Epipole sind die Abbildungen der Projektionszentren auf die Bildebene der jeweils anderen Kamera. Die Bildebenen werden jeweils durch die Epipolarebenen entlang der Epipolargeraden l', l'' geschnitten. Alle Epipolargeraden einer Bildebene gehören zum selben Strahlenbüschel und schneiden sich in deren Epipol.

Zur Berechnung der relativen Orientierung des Bildpaares nutzt man die Koplanaritätsbedingung (auch Komplanaritätsbedingung), die nur dann erfüllt ist, wenn sich die Raumstrahlen $\overrightarrow{\boldsymbol{X'_0 x}}$ und $\overrightarrow{\boldsymbol{X''_0 x}}$ im Objektpunkt $\boldsymbol{x}$ schneiden [McGlone 2004].

3.3 Relative Orientierung einer Bildsequenz

Die relative Anordnung zweier perspektivischer Abbildungen kann durch die Homographie, die Fundamental Matrix oder die Essential Matrix beschrieben werden. Dafür werden nur homologe 2D-Bildpunkte benötigt.

3.3.1 Die Fundamental Matrix

Die Fundamental Matrix $\boldsymbol{F}$ ist die algebraische Repräsentation der Epipolargeometrie in Bildkoordinaten. Sie beschreibt die Beziehung zwischen dem Bildpunkt x' und seiner Epipolargeraden l'' im zweiten Bild:

$$\boldsymbol{l}'' = \boldsymbol{F}\boldsymbol{x}' \quad \text{bzw.} \quad \boldsymbol{l}' = \boldsymbol{F}^T\boldsymbol{l}''. \tag{3.21}$$

Ein Punkt des einen Bildes muss also auf einer Epipolargeraden des anderen Bildes liegen. Mit dieser Beziehung kann der Suchbereich bei der automatischen Erkennung korrespondierender Bildpunkte eingeschränkt werden. Für die projektive Transformation $\boldsymbol{x}' \mapsto \boldsymbol{l}''$ existiert keine inverse Abbildung $\boldsymbol{l}' \mapsto \boldsymbol{x}''$. $\boldsymbol{F}$ ist daher singulär und es gilt die Singularitätsbedingung $\det(\boldsymbol{F}) = 0$. Das bedeutet, dass sich alle Epipolargeraden in genau einem Punkt schneiden.

Geometrisch betrachtet repräsentiert die homogene 3×3-Matrix $\boldsymbol{F}$ eine Abbildung von der zweidimensionalen projektiven Ebene des ersten Bildes auf das Strahlenbüschel der Epipolargeraden des zweiten Bildes, also eine Abbildung vom $\mathbb{P}^2$ nach $\mathbb{P}^1$. Daraus folgt, dass $\boldsymbol{F}$ den Rang zwei hat. $\boldsymbol{F}$ stellt die projektive Beziehung zwischen zwei Bildern her. In ihr sind die Projektionsmatrizen $\boldsymbol{P}$ der beiden Bilder mit jeweils 11 Freiheitsgraden (siehe Tab. 4.1) enthalten. Da $\boldsymbol{F}$ unter einer projektiven Transformation, die 15 Freiheitsgrade besitzt (siehe Tab. 3.2), invariant ist, ergibt sich die Anzahl der Freiheitsgrade zu 2×11-15=7 [Rodehorst 2004]. Mit ihren neun Elementen ist $\boldsymbol{F}$ damit überparametrisiert.

Für die Epipole gilt $\boldsymbol{F}e' = 0$ und $\boldsymbol{F}^Te'' = 0$. Damit lassen sich aus diesen linearen homogenen Gleichungssystemen beide Epipole direkt aus der $\boldsymbol{F}$-Matrix mit Hilfe einer Singulärwertzerlegung berechnen. Zur Bestimmung beider Epipole e', e'' genügt folgende Singulärwertzerlegung[1]:

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T = \begin{bmatrix} \cdots & e'_x \\ \cdots & e'_y \\ \cdots & e'_w \end{bmatrix} \boldsymbol{W} \begin{bmatrix} \cdots & e''_x \\ \cdots & e''_y \\ \cdots & e''_w \end{bmatrix}^T . \tag{3.22}$$

Die Epipolarpunkte entsprechen der Abbildung des Aufnahmezentrums der jeweils anderen Kamera. Da sie den Fluchtpunkten der Basislinie entsprechen, kann die Translationsrichtung mit ihnen bestimmt werden.

[1] Die homogenen Koordinaten der Epipole befinden sich jeweils in der letzten Spalte.

Für alle Punkte, die auf einer Epipolargeraden liegen, gilt die Inzidenzrelation $\boldsymbol{x}'^T\boldsymbol{l}' = \boldsymbol{x}''^T\boldsymbol{l}'' = 0$. Mit (3.21) folgt daraus, dass die Koplanaritätsbedingung

$$\boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x}'' = 0 \tag{3.23}$$

erfüllt ist.

Aus der Fundamental Matrix[2] können im unkalibrierten Fall die zwei Projektionsmatrizen zur Beschreibung der Abbildungsgeometrie abgeleitet werden. Die Bestimmung von $\boldsymbol{F}$ basiert allein auf einer Menge von homologen Bildpunkten ohne Kenntnis der Kameraparameter.

Bei den linearen Schätzverfahren müssen die Messwerte zur optimaleren Konditionierung der Gleichungssysteme und damit zur robusteren Berechnung normiert werden. Es sind zahlreiche Verfahren zur Berechnung der $\boldsymbol{F}$-Matrix entwickelt worden. Zu den bekanntesten Verfahren zählt der Acht-Punkt-Algorithmus [R. I. Hartley 1997].

3.3.2 Die Essential Matrix

Die Essential Matrix $\boldsymbol{E}$ wurde 1981 von Longuet-Higgins [Longuet-Higgins 1981] vor der Fundamental Matrix eingeführt[3], stellt aber eine spezialisierte Version der Fundamental Matrix mit fünf Freiheitsgraden, bestehend aus drei Rotationen und zwei Translationen dar. Der fehlende Freiheitsgrad, die Länge des Translationsvektors, kann wegen der Mehrdeutigkeit, die sich aus der unterschiedliche Tiefe der Raumpunkte ergibt, nicht bestimmt werden. $\boldsymbol{E}$ ist eine kalibrierte Version von $\boldsymbol{F}$ mit normalisierten Bildkoordinaten $\hat{\boldsymbol{x}}$, bei der der Einfluss der bekannten Kalibriermatrix $\boldsymbol{K}$ eliminiert wird ($\boldsymbol{K}^{-1}\boldsymbol{P} = [\boldsymbol{R}|\boldsymbol{t}]$). Eine Fundamental Matrix mit zwei normalisierten Kameras $\boldsymbol{P}' = [\boldsymbol{I}|0]$ und $\boldsymbol{P}'' = [\boldsymbol{R}|\boldsymbol{t}]$ ist somit eine Essential Matrix.

$\boldsymbol{E}$ ist wie $\boldsymbol{F}$ eine 3×3-Matrix mit Rang zwei. Daher gilt auch die Singularitätsbedingung $\det(\boldsymbol{E}) = 0$. Für die normalisierten Bildkoordinaten $\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}''$ gilt die Koplanaritätsbedingung (auch Longuet-Higgins Gleichung [Longuet-Higgins 1981])

$$\hat{\boldsymbol{x}}''\boldsymbol{E}\hat{\boldsymbol{x}}' = 0. \tag{3.24}$$

Zusammen mit der Rangbedingung stellen die folgenden Eigenschaften sicher, dass eine 3×3 Matrix mit $\det(\boldsymbol{E}) = 0$ eine Essential Matrix ist [Faugeras und Luong 2001]: Die beiden von Null verschiedenen Eigenwerte von $\boldsymbol{E}$ sind gleich. Diese Eigenschaft kann in der Singulärwertzerlegung von $\boldsymbol{E}$ genutzt werden [R. Hartley und Zisserman 2004]:

$$\boldsymbol{E} = \boldsymbol{U}\mathrm{diag}(d, d, 0)\boldsymbol{V}^T. \tag{3.25}$$

[2] Die $\boldsymbol{F}$-Matrix ist die einzige Größe in diesem Kapitel, der ein ganzes Lied gewidmet wurde: `http://danielwedge.com/fmatrix`

[3] Laut [R. Hartley und Zisserman 2004] wurde die Idee aber bereits 1908 in der Dissertation von H. von Sanden [Sanden 1908] verwendet.

	$\boldsymbol{F}$	$\boldsymbol{E}$
Raum	projektiv	euklidisch
Zerlegung	$\boldsymbol{H}$	$\boldsymbol{R}$, $\boldsymbol{t}$
Rang	2	2
Freiheitsgrade	7	5
Definition	$\boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x}'' = 0$	$\hat{\boldsymbol{x}}'^T\boldsymbol{E}\hat{\boldsymbol{x}}'' = 0$

Tabelle 3.3: Zusammenstellung der Eigenschaften von $\boldsymbol{F}$ und $\boldsymbol{E}$ Matrix

Die Beziehung zwischen $\boldsymbol{E}$ und $\boldsymbol{F}$ wird mit

$$\boldsymbol{E} = \boldsymbol{K}''^T\boldsymbol{F}\boldsymbol{K}' \tag{3.26}$$

bzw.

$$\boldsymbol{F} = \boldsymbol{K}''^{-T}\boldsymbol{E}\boldsymbol{K}'^{-1} \tag{3.27}$$

[Faugeras und Luong 2001] hergestellt, wobei $\hat{\boldsymbol{x}} = \boldsymbol{K}^{-1}\boldsymbol{x}$.

Mit $\boldsymbol{E} = [\vec{t}]_\times\boldsymbol{R}$ können Rotation $\boldsymbol{R}$ und Translationsrichtung $\vec{t}$ aus $\boldsymbol{E}$ über eine Singulärwertzerlegung bestimmt werden. Diese führt theoretisch zu vier verschiedenen Lösungen von denen aber nur diejenige Plausibel ist, bei der sich ein trianguliertер Punkt vor beiden Kameras befindet.

Das ursprüngliche Verfahren von Longuet-Higgins [Longuet-Higgins 1981] zur Berechnung von $\boldsymbol{E}$ benötigt acht 2D-Punktkorrespondenzen. Es gibt allerdings verschiedene Punktkonstellationen, bei denen $\boldsymbol{E}$ nicht lösbar ist:

- Vier Punkte liegen auf einer Geraden,
- sieben Punkte liegen in einer gemeinsamen Ebene,
- sechs Punkte sind die Eckpunkte eines regelmäßigen Sechsecks,
- acht Punkte sind die Eckpunkte eines Kubus.

Die beiden letzten Punkte sind eher unwahrscheinlich.

Minimal werden fünf 2D-Punktkorrespondenzen zur Berechnung benötigt. Der Fünf-Punkt Algorithmus von Nistér [Nistér 2004] zählt zu den bekanntesten Verfahren zur Schätzung der Essential Matrix.

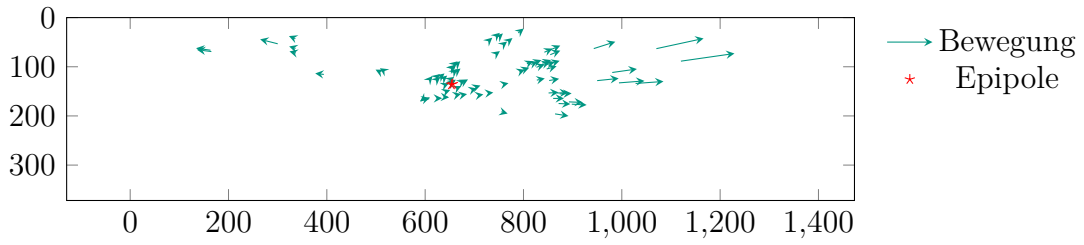


Abbildung 3.4: Bewegung der Bildmerkmale bei einer Translation der Kamera nach vorne [Pixel]

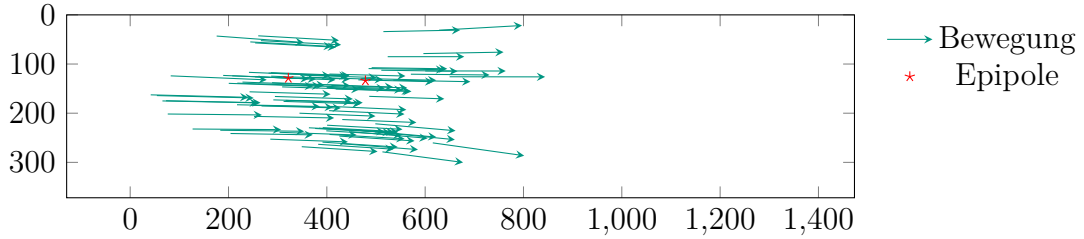


Abbildung 3.5: Bewegung der Bildmerkmale bei einer Kamerabewegung mit hohem Rotationsanteil [Pixel]

3.3.3 Spezialfälle für Bewegungen zwischen zwei Bildern

Bei einer reinen Translation entlang der optischen Achse einer Kamera auf einen Punkt hin haben beide Epipole identische Bildkoordinaten, falls die homologen Punkte fehlerfrei sind (siehe Abb. 3.4, S. 31). Ohne Rotation $\boldsymbol{R}$ wird

$$\boldsymbol{F} = \boldsymbol{K''}^{-T}[\vec{t}]_{\times}\boldsymbol{K'}^{-1}. \tag{3.28}$$

Bei nur einer Kamera, also $\boldsymbol{K'} = \boldsymbol{K''}$ wird $\boldsymbol{F} = [e']_x = [e'']_x$.

Im achsparallelen Fall sind die beiden Bildebenen parallel zueinander. Alle Schnittgeraden der Epipolarlinien mit den Bildebenen verlaufen parallel zueinander. Die Basislinie b schneidet sich mit den Bildebenen im Unendlichen. Daraus folgt, dass auch die Epipole $\boldsymbol{e'}, \boldsymbol{e''}$ im Unendlichen liegen. Dies entspricht dem Stereonormalfall.

Die Essential Matrix vereinfacht sich dann zu [McGlone 2004]

$$\boldsymbol{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -b' \\ 0 & b' & 0 \end{bmatrix}. \tag{3.29}$$

3.4 Absolute Orientierung einer Bildsequenz

Unter absoluter Orientierung versteht man allgemein die Bestimmung der Transformation zweier Koordinatensysteme mit Hilfe bekannter, korrespondierender Punkte

zweier starrer Punktwolken. Sie wird zur Registrierung von 3D-Punktwolken verschiedener Standpunkte aus z. B. 3D-Laserscanns genutzt, um eine Gesamtpunktwolke aus einzelnen Aufnahmen zu erhalten oder z. B. zur Berechnung der Eigenbewegung von Robotern, die ihre Umgebung mit verschiedenen Sensoren geometrisch erfassen.

Bei einer Stereobildsequenz liegen zusätzlich zu den korrespondierenden Bildmessungen nach erfolgter Triangulation auch 3D-Punktkorrespondenzen vor. Zur Ermittlung der Eigenbewegung eines Stereokamerasystems kann man die Verfahren deshalb in zwei Abschnitte einteilen: Berechnung der Orientierung nur aus den 3D-Punktkorrespondenzen und Berechnung mit 2D-3D-Punktkorrespondenzen.

3.4.1 Orientierung aus 3D-Punktkorrespondenzen

Der ICP Algorithmus (Iterative Closest Points) von [Besl und McKay 1992] bestimmt iterativ die Bewegung zweier Punktwolken zueinander. In jedem Iterationsschritt (*iterative*) werden dabei die i korrespondierenden 3D-Punkte ($X_{t,i}$, $X_{t+1,i}$) mit dem geringsten Abstand zueinander (*closest points*) ermittelt und daraus die Transformation $\boldsymbol{T}$ $(\boldsymbol{R}, \boldsymbol{t})$ bestimmt, die die Quadratsumme aller Abstände minimiert [Segal u. a. 2009]:

$$\boldsymbol{T} \leftarrow \arg\min_{\boldsymbol{T}} \{\sum_i w_i \|\boldsymbol{T} \cdot X_{t+1,i} - m_i\|^2\}. \tag{3.30}$$

Neben der initialen Transformation $\boldsymbol{T_0}$ wird ein Schwellwert $d_{\max}$ für einen maximalen Abstand benötigt, da nicht für alle Punkte aus $\boldsymbol{X_t}$ auch Punkte in $\boldsymbol{X_{t+1}}$ vorhanden sein müssen, wenn sich die Punktwolken z. B. nur in einem Bereich überlappen. In 3.30 sind m_i alle Punkte, die den kleinsten Abstand zu den Punkten $\boldsymbol{T} \cdot \boldsymbol{X_{t+1,i}}$ haben. Für das Gewicht w_i gilt:

$$w_i = \begin{cases} 1 & \|m_i - \boldsymbol{T} \cdot \boldsymbol{X_{t+1,i}}\| \leq d_{\max} \\ 0 & \|m_i - \boldsymbol{T} \cdot \boldsymbol{X_{t+1,i}}\| > d_{\max}. \end{cases} \tag{3.31}$$

Ein zu großes $d_{\max}$ kann zu falschen Punktkorrespondenzen und damit zu einem falschen Transformationsergebnis führen. Ein zu kleines $d_{\max}$ wiederum kann dazu führen, dass überhaupt keine Korrespondenzen gefunden werden [Segal u. a. 2009]. Ein schlechter Startwert $\boldsymbol{T_0}$ für die Iteration kann zusammen mit einem ungeeigneten $d_{\max}$ leicht dazu führen, dass der ICP-Algorithmus nicht konvergiert oder ein lokales Minimum annimmt und somit das Ergebnis falsch ist.

Allgemein können die verschiedenen ICP-Verfahren in sechs Abschnitte unterteilt werden [Rusinkiewicz und Levoy 2001] in denen die einzelnen Verfahren üblicher Weise variieren: die *Auswahl* der Punkte in einem oder beiden Punktwolken, das *Bestimmen der Korrespondenzen* (Matching) und deren *Gewichtung* sowie der *Ausschluss von Ausreißerpaaren*, die Auswahl des *Fehlermaßes* und der zugehörigen *Minimierung*.

Die Transformation selbst kann mit verschiedenen Minimierungsverfahren für alle korrekten Punktkorrespondenzen berechnet werden, wobei der Hauptschwerpunkt auf der Modellierung der Rotation liegt. So gibt es z. B. Verfahren die mit Hilfe einer Singulärwertzerlegung [Arun u. a. 1987] arbeiten, ein Verfahren mit Quaternionen [Horn 1987], ein weiteres Verfahren auf Basis der orthonormalen Eigenschaften der Rotationsmatrix [Horn, Hilden u. a. 1988] und ein Verfahren bei dem Translation und Rotation als Duale Quaternionen repräsentiert werden [Walker u. a. 1991]. Die genannten Verfahren wurden in [Eggert u. a. 1997] verglichen und haben eine ähnliche Leistung bezüglich ihrer Genauigkeit, der Robustheit in Bezug auf fehlerhafte Koordinaten und der Stabilität bei bestimmten geometrischen Konfigurationen, wie z. B. Punkte auf einer Ebene oder Linie.

[Ohta und Kanatani 1998] weisen darauf hin, das Kleinste-Quadrate Methoden wie [Arun u. a. 1987; Horn, Hilden u. a. 1988; Horn 1987] davon ausgehen, dass die Punkte fehlerfrei beobachtet werden, die transformierten Punkte jedoch der Normalverteilung unterliegen. Dieses Fehlermodell trifft unter realistischen Bedingungen nicht zu. Bei Stereoaufnahmen oder bei Tiefenbildmessungen sind die Fehler entlang der Tiefenachse groß im Vergleich zu den anderen Achsrichtungen. Nahe Punkte sind genauer als entfernte Punkte.

Punktwolken aus Stereotriangulationsverfahren oder Tiefenbildkameras haben keine wirkliche räumliche Verteilung im Dreidimensionalen, sondern bilden eine Oberfläche ab (2,5D statt 3D). Die Tiefenwerte können auch nur bestimmte numerische Werte annehmen, wodurch sich die Punkte auf diskreten Ebenen befinden (siehe Kap. 4, S. 39). Die Suche nach korrespondierenden Punkten kann dadurch erschwert oder gar unmöglich werden.

Das „Generalized ICP“-Verfahren [Segal u. a. 2009] löst dieses Problem mit einer „point-to-plane“ Variante des ICP-Algorithmus. Statt der Minimierung in Formel 3.30 wird der Abstand entlang der Oberflächennormalen n_i am Punkt m_i minimiert:

$$\boldsymbol{T} \leftarrow \arg\min_{\boldsymbol{T}} \{\sum_i w_i \| n_i \cdot (\boldsymbol{T} \cdot X_{t+1,i} - m_i) \|^2\}. \tag{3.32}$$

3.4.2 Orientierung aus 2D-3D-Punktkorrespondenzen

Wenn die 3D-Objektkoordinaten der gefundenen Bildmerkmale bekannt sind (z. B. aus der Triangulation mit einer Stereokamera), kann die Kameraposition und -Orientierung direkt aus diesen 3D-2D-Korrespondenzen mit einem räumlichen Rückwärtsschnitt abgeleitet werden. Die Verfahren sind auch als *perspective from n points* (PnP) Problem bekannt. Mehrere minimale Lösungen des PnP Problems wurden bisher entwickelt und bewertet [Haralick u. a. 1994], [Gao u. a. 2003].

Die Bewegungsschätzung mit 3D-2D-Korrespondenzen ist genauer als mit 3D-3D-Korrespondenzen da hier statt der 3D-Punktabstände der Rückprojektionsfehler

minimiert wird [Nistér u. a. 2004]. Wenn die 3D-Punkte aus Stereoverfahren entstanden sind, haben diese einen mit zunehmendem Abstand größer werdenden Fehler.

Bei vier Korrespondenzen ist die Lösung eindeutig. [Moreno-Noguer u. a. 2007; Moreno-Noguer u. a. 2008] haben einen PnP-Algorithmus entwickelt, der variabel mit $n \geq 4$ Punkten zurecht kommt und die dafür benötigte Rechenzeit nur linear mit der Anzahl n der Punkte ansteigt (Komplexität $\mathcal{O}(n)$). Das bestätigen auch eigene Tests (siehe Abb. 3.6, S. 34).

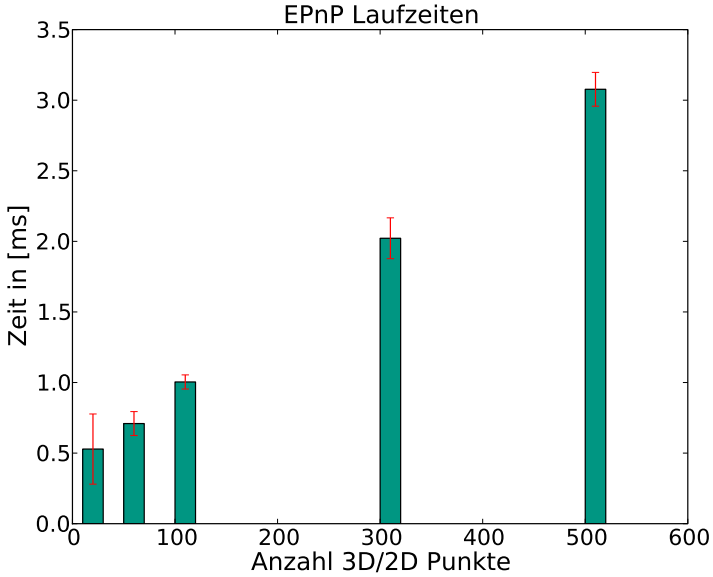


Abbildung 3.6: Vergleich der EPnP Laufzeiten bei unterschiedlicher Anzahl von 2D-3D-Punktkorrespondenzen

Im Folgenden wird der EPnP-Algorithmus (**e**nhanced **p**erspective-**n**-**p**oint) näher beschrieben. Als Quellen dienten [Moreno-Noguer u. a. 2007; Moreno-Noguer u. a. 2008] und die darin enthaltenen Quellen, sowie die von den Autoren zur Verfügung gestellte Implementierung in C++.

Die n 3D-Objektpunkte werden als eine gewichtete Summe vier virtueller, nichtkoplanarer Passpunkte im Weltkoordinatensystem (c_i^{Welt}) ausgedrückt. Es gilt für alle Punkte p_n:

$$p_n^{\text{Kamera}} = \sum_{i=1}^{4} \alpha_{i,n} c_i^{\text{Kamera}} \quad \text{und} \quad p_n^{\text{Welt}} = \sum_{i=1}^{4} \alpha_{i,n} c_i^{\text{Welt}}. \tag{3.33}$$

Die 12 Parameter dieser Passpunkte im Kamerakoordinatensystem $\boldsymbol{x} = [c_{i=1\ldots4}^{\text{Kamera}}]^T$ sind zunächst die Unbekannten. $\alpha_{i,n}$ und c_i^{Welt} sind bekannt.

Bei der Lösung des linearen Gleichungssystems $\boldsymbol{M}\boldsymbol{x} = 0$ mit der $2n \times 12$ Matrix $\boldsymbol{M}$ hat das Produkt $\boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{M}$ mit der Komplexität $\mathcal{O}(n)$ den größten Anteil an

der Berechnung. $\boldsymbol{M^T M}$ hat eine konstante Größe von 12×12. Die unbekannten $\boldsymbol{x}$ werden als Linearkombination der Null-Eigenvektoren ausgedrückt:

$$\boldsymbol{x} = \sum_{i=1}^{N} \beta_i \boldsymbol{v_i}. \tag{3.34}$$

wobei $\boldsymbol{v_i}$ die bekannten Eigenvektoren von $\boldsymbol{M^T M}$ sind. Die Dimension des Nullraumes N und die Koeffizienten β_i müssen berechnet werden. Im Falle fehlerfreier Punkte wäre $N = 1$. Aus Simulationen wurden die zu berücksichtigenden $N \in \{1, 2, 3, 4\}$ für fehlerbehaftete Punkte empirisch ermittelt. Für diese vier Fälle wird (3.34) berechnet und diejenige Lösung als die Beste bestimmt, deren Rückprojektionsfehler minimal ist. Damit sind die Gewichte β_i die eigentlichen vier Unbekannten. Da die räumliche Anordnung der Passpunkte zueinander in Welt- und Kamerakoordinatensystem fest ist, bleiben deren 6 Strecken d_{ij} zueinander auch fest. [Moreno-Noguer u. a. 2008] stellen mit den folgenden Bedingungen die quadratischen Gleichungen zur Lösung der β_i auf:

$$d_{ij}^2 = \|c_i^{\text{Kamera}} - c_j^{\text{Kamera}}\|^2 = \|c_i^{\text{Welt}} - c_j^{\text{Welt}}\|^2. \tag{3.35}$$

Die so ermittelten β_i werden anschließen noch mit der Methode der kleinsten Quadrate verbessert. In der verwendeten Implementierung wird dafür eine feste Anzahl von fünf Iterationen benötigt. Mit bekannten β_i können auch die p_n^{Kamera} berechnet werden. Abschließend wird die gesuchte äußere Orientierung $(\boldsymbol{R}, \boldsymbol{T})$ zwischen den p_n^{Kamera} und p_n^{Welt} mit Horns absoluten Orientierungsverfahren [Horn, Hilden u. a. 1988] auf Basis der orthonormalen Matrix berechnet.

Zum einen gibt es geometrische Konfigurationen, die zu keinen oder instabilen Lösungen führen und zum anderen kann es durch die Merkmalsextraktion zu vielen fehlerhaften 3D-2D-Zuordnungen kommen.

Auch EPnP ist nicht robust gegenüber solchen Fehlzuordnungen. In Abb. 3.7 ist der Rückprojektionsfehler des EPnP bei zunehmendem Ausreißeranteil dargestellt. Zuerst wurden 1000 zufällig verteilte 3D-Objektpunkte erzeugt, die dann auf die Bildfläche einer virtuellen Kamera projiziert wurden. Anschließend wurden die Bildpunkte mit einem zufälligen Fehler von max. 10 Pixel versehen (gestrichelte, rot Linie in Abb. 3.7), ein Ausreißeranteil von 0 % bis 50 % in Zehnerschritten hinzugefügt und diese Simulation 5000 mal durchgeführt. Man kann erkennen, dass der EPnP bereits bei 10 % Ausreißeranteil einen deutlichen Anstieg im mittleren Rückprojektionsfehler hat im Vergleich zum Rückprojektionsfehler ohne Ausreißer.

Robuste Verfahren wie RANSAC liefern noch bei einem Ausreißeranteil von 50 % Lösungen, wie es schon Fischler und Bolles [Fischler und Bolles 1981] am Beispiel eines minimalen PnP-Verfahrens demonstriert haben. In Abb. 3.7 zeigen die roten Balken den Rückprojektionsfehler mit RANSAC. Selbst bei einem Ausreißeranteil von 60 % bleibt dieser konstant deutlich unter 10 Pixel. Das EPnP-RANSAC-Verfahren ist robust.

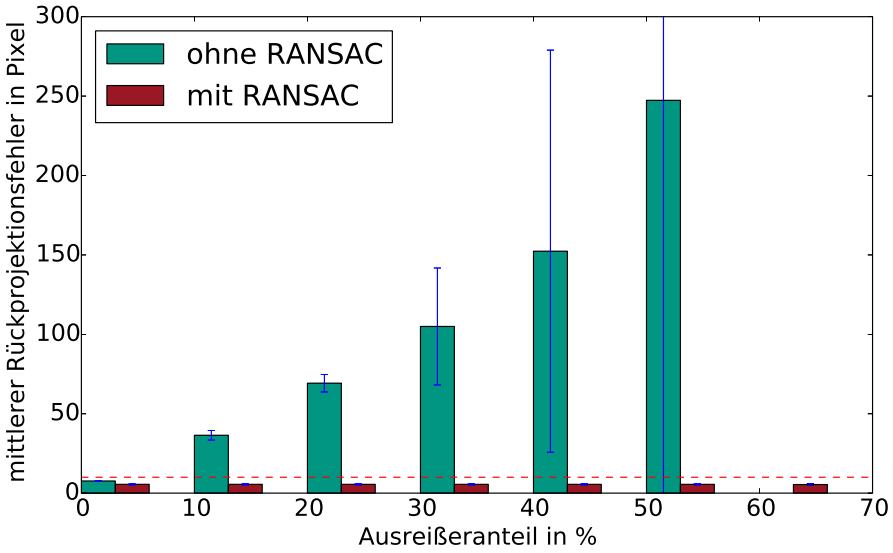


Abbildung 3.7: EPnP Rückprojektionsfehler bei zunehmendem Ausreißeranteil (Simulation mit 5000 Durchläufen. Die gestrichelte, rote Linie zeigt den max. Fehler der Bildmessung von 10 Pixel.)

3.5 Trajektorien aus Bildsequenzen

Der vom Stereokamerasystem zurückgelegte Weg kann als Trajektorie durch das fortlaufende Akkumulieren der relativen Bewegungen zwischen den Bildern erzeugt werden.

Der Fehler der dabei durch die Fehlerfortpflanzung mit akkumuliert wird, macht sich mit zunehmender Wegstrecke als Drift bemerkbar. Die Leistung eines VO-Verfahrens kann an seinem Driftverhalten gemessen werden.

Der VO-Ansatz muss allerdings nicht isoliert betrachtet werden. Gerade bei einem ARS kann die VO-Lösung Bewegungsschätzungen zu einem systemweiten Kalman Filter Kreislauf mit weiteren Sensoren beitragen.

Eine Möglichkeit zur Reduktion der Drift bieten Ansätze, die eine bestimmte Anzahl von Bewegungsschätzungen in der Zeit zurück mit einem Bündelblockansatz verfeinern [Scaramuzza und Fraundorfer 2011]. Diese Technik ist als *Sliding Window* Bundleadjustment bekannt [Sünderhauf u. a. 2006; Scaramuzza und Fraundorfer 2012], da die Anzahl der rückblickenden Zeitschritte entlang der Trajektorie konstant bleibt und mit der aktuellen Lösung weiter läuft. Allerdings müssen dafür auch die 3D-Punkte mitgeführt werden.

Zur Bewertung von Trajektorien gibt es aus dem SLAM Umfeld für Benchmarkdatensätze wie z. B. die „Rawseeds“-Sammlung [Ceriani u. a. 2009] zwei Maße: den

Absolute Trajectory Error (ATE) und den *Relative Pose Error* (RPE). Beide bewerten die Posen der Sensorplattform mit Hilfe von „Ground-Truth"-Posen.

Der ATE bewertet nur die Translation $\boldsymbol{t}$ der Bewegung. Es werden jeweils die Abstände d_i für alle Posen i zwischen der ermittelten Pose und der „Ground-Truth"-Pose ($\boldsymbol{t}^{\mathrm{GT}}$) ermittelt:

$$d_i = \|\boldsymbol{t}_i - \boldsymbol{t}_i^{\mathrm{GT}}\|. \tag{3.36}$$

Die Zuordnung der Messwerte zu den Referenzwerten erfolgt über Zeitstempel. Gefordert sind der Mittelwert $\bar{d}$, die Standardabweichung σ_d sowie Minimum und Maximum des 3σ-Intervalls aller Abstände d_i. Auf diese Weise erhält man ein Maß für den akkumulierten Fehler mit dem man die Leistung von VO oder SLAM Systemen bewerten kann.

Der *Relative Pose Error* (RPE) berechnet die Genauigkeiten der relativen Transformationen zwischen ermittelten Posen und Referenzwerten. Da nur relative Transformationen bewertet werden, wird der akkumulierte Fehler der Trajektorie nicht berücksichtigt. Der RPE eignet sich vielmehr zur Bewertung der Drift wobei Translationen (T-RPE) und Rotationen (R-RPE) getrennt betrachtet werden.

3.6 Zusammenfassung

Mit den vier Bildern zweier benachbarter Stereoaufnahmen stehen mehrere Verfahren zur Bewegungsberechnung zur Verfügung, die in Tabelle 3.4 bezüglich der benötigten Anzahl von Bilder bzw. Punktkorrespondenzen zusammengefasst sind. Speziell für Stereokamerasysteme kommen besonders die Verfahren mit 3D-3D- und 2D-3D-Korrespondenzen in Frage.

Unterschiedliche Arbeiten hierzu werden in Kapitel 6 besprochen und eine Auswahl in Kapitel 7 mit einem eigenen Ansatz verglichen.

Anzahl Bilder	Verfahren	Punktkorrespondenzen	
		Typ	min. Anzahl
2	Homographie	2D	4
2	E-Matrix	2D	5
4	ICP über Punktwolken	3D	(3)
4	PnP-Lösungen	2D+3D	4

Tabelle 3.4: Mögliche Verfahren zur Eigenbewegungsberechnung eines Stereokamerasystems mit Hilfe korrespondierender Bildpunkte (2D) und Objektpunkte (3D) aus einem Stereoverfahren

Die dafür verwendeten Aufnahmesysteme werden im folgenden Kapitel besprochen.

4 Aufnahmesysteme

In diesem Kapitel werden die Kamerasysteme besprochen, die in dieser Arbeit zum Einsatz kommen: ein Stereokamerasystem, konfiguriert aus zwei synchron aufnehmenden, monochromen FireWire Kameras und eine Kinect Tiefenbildkamera, die ein Farbbild und ein Tiefenbild liefert.

4.1 Die monokulare Kamera

Um das mathematische Modell der Zentralprojektion einer Kamera, die kollineare Abbildung, nutzen zu können, muss der Strahlengang vom Objektpunkt zum Bildpunkt entsprechend korrigiert und modelliert werden. Im Folgenden wird die Modellierung von typischen Videokameras mit einem zweidimensionalen Sensorfeld behandelt.

Heutzutage ist der CMOS Sensor am weitesten verbreitet und hat die CCD Sensoren verdrängt. In mobilen Endgeräten aller Art sind diese als SoC Kameras (*System on a Chip*) integriert, oft auch mehrfach. Die Nachteile dieser Kameras sind die kleine Sensorfläche, die zu einem schlechteren Signal-Rausch-Verhältnis führt und vor allem das zeilenweise Auslesen der Sensorwerte. Diese Ausleseart wird *Rolling Shutter* (auch Rollverfahren) genannt und kann zu geometrischen Verzerrungen bei bewegten Szenen führen, da nur alle Sensorelemente einer Zeile oder einer Spalte zur gleichen Zeit ausgelesen werden und somit für das bewegte Objekt keine kollineare Abbildung erzeugt wird. Eine CMOS Kamera mit einem *Global Shutter* liest alle Sensorelemente zeitgleich aus und umgeht damit dieses Problem. Dieser Kameratyp wird für Bildverarbeitungssysteme bevorzugt.

Als Bildelement (Pixel, von *picture element*) wird die Fläche, für die das Sensorelement repräsentativ ist, bezeichnet. Der Ursprung ist in der linken oberen Ecke und beginnt mit dem Index 0. Um einen Punkt $\boldsymbol{x}'_b = (x'_b, y'_b)^T$ im Sensorkoordinatensystem in einen Punkt in der Kamerabildebene $\boldsymbol{x}'$ umzurechnen, sind die geometrischen Parameter des Sensors wie folgt festgelegt:

- Verschiebung des Koordinatensystems in den Hauptpunkt mit den Koordinaten (x'_0, y'_0) im Sensorsystem. Der Maßstab der x'-Achse wird übernommen. Er entspricht dem Zeilenabstand der Sensorelemente.
- Maßstabskorrektur der y'-Achse um den Faktor $1 + m$. Dies entspricht dem Abstandstand der Sensorelemente in den Zeilen.
- Eine mögliche Scherung s des Sensorkoordinatensystems. Bei heutigen Sensoren ist ein Wert für $s \neq 0$ eher unwahrscheinlich.

Damit ergibt sich die Affinabbildung vom Sensorsystem in das Kamerasystem in homogenen Koordinaten zu

$$\boldsymbol{H} = \begin{bmatrix} 1 & s & x'_0 \\ 0 & 1+m & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

Aus dem Abbildungsvorgang einer digitalen Kamera vom Objektraum in den Bildraum folgt dann mit (3.16) die Kalibriermatrix $\boldsymbol{K}_C$

$$\boldsymbol{K}_C = \boldsymbol{H}\boldsymbol{K} = \begin{bmatrix} 1 & s & x'_0 \\ 0 & 1+m & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c & c \cdot s & x'_0 \\ 0 & c(1+m) & y'_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$\boldsymbol{K}_C$ ist eine obere Dreiecksmatrix mit fünf Freiheitsgraden deren Elemente die geometrischen Eigenschaften der Kamera mit Kamerakonstante c, Hauptpunktslage $\boldsymbol{x}'_0$ und $\boldsymbol{y}'_0$, Maßstabsunterschied m und Scherung s definieren. Ist $\boldsymbol{K}$ bekannt, ist die Kamera kalibriert.

Eine Übersicht von algebraischen Kameramodellen mit deren jeweiligen Kalibriermatrizen ist in Tab. 4.1 zusammengestellt. Die Kameramodelle unterscheiden sich in der Verwendung der Parameter der inneren Orientierung und modellieren eine reale Kamera. Die **Lochkamera** oder idealisierte Kamera hat c als einzigen Parameter mit dem Ursprung des Bildkoordinatensystems im Hauptpunkt. Die **normierte Kamera** ist ein Spezialfall einer Nadiraufnahme mit einer idealisierten Kamera und einer Kamerakonstanten $c = 1$. Die **euklidische Kamera** hat ein euklidisches Koordinatensystem in der Bildebene und Kamerakonstante c. Die **projektive Kamera** oder Kamera mit affinem Sensor kann ein affines Bildkoordinatensystem mit Maßstabsunterschied m und Scherung s in x- und y-Richtung haben. Die **Allgemeine Kamera** bildet nicht geradentreu ab. Es werden zusätzliche Parameter $\Delta x, \Delta y$ zur Wiederherstellung der Geradentreue benötigt. Die Projektionsmatrix $\boldsymbol{P} = \boldsymbol{K}\boldsymbol{R}[I|-\boldsymbol{X}_0]$ der projektiven Abbildung $\boldsymbol{x'} = \boldsymbol{P}\boldsymbol{X}$ einer digitalen Kamera hat insgesamt 11 Freiheitsgrade (siehe Tab. 4.2, S. 42). Die Zeilen $\boldsymbol{p^i}$ sind 4-Vektoren und beschreiben Ebenen im projektiven Raum $\mathbb{P}^3$. Die drei Ebenen schneiden sich im Projektionszentrum $\boldsymbol{X_0}$. Die Spalten $\boldsymbol{p_i}$ sind 3-Vektoren und beschreiben Punkte

Kameramodell	Kalibriermatrix $\boldsymbol{K}$
allgemein	$\boldsymbol{K} = \begin{bmatrix} c & c \cdot s & x'_0 + \Delta x' \\ 0 & c(1+m) & y'_0 + \Delta y' \\ 0 & 0 & 1 \end{bmatrix}$
projektiv	$\boldsymbol{K} = \begin{bmatrix} c & c \cdot s & x'_0 \\ 0 & c(1+m) & y'_0 \\ 0 & 0 & 1 \end{bmatrix}$
euklidisch	$\boldsymbol{K} = \begin{bmatrix} c & 0 & x'_0 \\ 0 & c & y'_0 \\ 0 & 0 & 1 \end{bmatrix}$
Lochkamera	$\boldsymbol{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$
normiert	$\boldsymbol{K} = \boldsymbol{I}$

Tabelle 4.1: Kameramodelle mit ihren Kameramatrizen ($\Delta x', \Delta y'$: nicht-lineare Bildfehler)

in der Projektionsebene. Die ersten drei Spalten $\boldsymbol{p_1}$, $\boldsymbol{p_2}$ und $\boldsymbol{p_3}$ beschreiben Abbildungen der Achsrichtungen des Weltkoordinatensystems. Damit entsprechen sie den Fluchtpunkten der Achsen X, Y und Z. Die $\boldsymbol{p_4}$-Spalte ist die Abbildung des Ursprungs vom Weltkoordinatensystem.

4.1.1 Modellierung der Linsenverzeichnung

Bisher ging die Modellierung des Kameraabbildungsvorgangs von einer geradentreuen Abbildung aus. Bei den nichtlinearen Bildfehlern $\Delta x', \Delta y'$ führt vor allem die Verzeichnung des Linsensystems zu Abweichungen der kollinearen Abbildung der oben modellierten idealisierten digitalen Kamera.

Die nicht geradentreuen Bildfehler können auf zwei unterschiedliche Weisen modelliert werden. Das *physikalische Modell* geht von den physikalischen Ursachen aus und modelliert diese entsprechend.

Das *mathematische Modell* modelliert die Wirkung der Effekte ohne deren physikalischen Ursachen zu hinterfragen. Die zusätzlichen Parameter unterdrücken die Wirkung dieser systematischen Bildfehler nur. In der Regel werden die Bildfehler als Polynome modelliert.

	Orientierung							
	äußere		innere					
Kameramodell	$\boldsymbol{X_0}$	$\boldsymbol{R}$	c	$(\boldsymbol{x_0}, \boldsymbol{y_0})$	m	s	$\Delta x', \Delta y'$	**DoF**
normiert	-	$\boldsymbol{I}$	1	$(0,0)$	1	0	0,0	3
Lochkamera	-	-	-	$(0,0)$	1	0	0,0	7
euklidisch	-	-	-	-	1	0	0,0	9
projektiv	-	-	-	-	-	-	0,0	11
allgemein	-	-	-	-	-	-	-	>11
	Transl.	Rot.	Affinität				nicht geradentreu	

Tabelle 4.2: Modellannahmen und Freiheitsgrade (dof: *degrees of freedom*) der verschiedenen Kameramodelle. Nach [Förstner 2004b]

Ein Beispiel für eine strenge Lösung ist das rotationssysmmetrische Modell für die Verzeichnung nach dem auch heute noch weit verbreiteten Ansatz von Brown [Brown 1971]. Danach ist die radialsymmetrische Verzeichnung

$$\Delta r'_{rad} = k_1 \cdot r'^3 + k_2 \cdot r'^5 + k_3 \cdot r'^7 + \dots \tag{4.3}$$

wobei für die meisten Objektive nach dem zweiten oder dritten Term abgebrochen werden kann. Zur Vermeidung der Korrelation mit der Kamerakonstanten wird ein linearer Anteil der Verzeichnungsfunktion abgespalten, wodurch die Verzeichnungskurve einen zweiten Nulldurchgang durch die r'-Achse bekommt, der so gewählt wird, dass maximale und minimale Verzeichnungswerte innerhalb des genutzten Bildformates etwa gleich groß werden.

Dezentrierungen der Linsen im Objektiv führen zu radial-asymmetrischen und tangentialen Verzeichnungen. Ihr Anteil kann nach Brown wie folgt berücksichtigt werden:

$$\Delta x'_{tan} = p_1 \cdot (r'^2 + 2x'^2) + 2p_2 \cdot x' \cdot y' \tag{4.4}$$

$$\Delta y'_{tan} = p_2 \cdot (r'^2 + 2y'^2) + 2p_1 \cdot x' \cdot y' \tag{4.5}$$

Der Anteil ist geringer als die radial-symmetrische Verzeichnung. Bei einfachen Objektiven und hohen Genauigkeitsansprüchen sollte die tangentiale Verzeichnung mitbestimmt werden. Der gleiche Ansatz [Heikkilä und Silvén 1997] wird sowohl in der „Camera Calibration Toolbox for Matlab“ [Bouguet 2014] als auch in der Open Computer Vision Bibliothek (OpenCV) verwendet. Die korrigierten Bildkoordinaten ergeben sich aus

$$x' = x + (k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_1 xy + p_2(r^2 + 2x^2)) \tag{4.6}$$

$$y' = y + (k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_2 xy + p_2(r^2 + 2y^2)) \tag{4.7}$$

wobei $r^2 = x^2 + y^2$. k_1, k_2 und k_3 die Parameter der radial-symmetrischen Verzeichnung, p_1 und p_2 die der tangential-asymmetrischen Verzeichnung sind.

4.1.2 Farbsensoren

Die Extraktion geometrisch relevanter Bildmerkmale erfolgt üblicherweise in Intensitätsbildern. Für ein VO System wählt man daher am besten monochrome Kameras. Wenn man allerdings bei mobilen Anwendungen auf die bereits in den Mobilgeräten integrierten Kameras zurückgreifen muss oder will, wird man dort immer Farbsensoren vorfinden. Auch in der Kinect Kamera ist u.a. eine Farbkamera integriert. Daher werden in diesem Abschnitt die Probleme von Farbsensoren besprochen.

Farbe nennt man den für Menschen sichtbaren Bereich des Lichts zwischen etwa 380 nm (UV) und 780 nm (IR) im elektromagnetischen Spektrum. Die spektrale Empfindlichkeit ist von Mensch zu Mensch selbst bei normaler Sehkraft leicht unterschiedlich, daher lässt sich dieser Bereich nicht exakt definieren. Die Netzhaut des Menschen enthält zwei Arten von Photorezeptorzellen: die Stäbchen, verantwortlich für das skotopische Sehen bei schwachem Licht (Dunkelsehen) und die Zapfen, die für das Farbsehen (auch Hellsehen, photopisches Sehen) verantwortlich sind. Der Mensch hat drei Zapfenarten, die jeweils unterschiedlich auf das einfallende Licht reagieren. L-Zapfen haben ihr maximales Antwortverhalten im langwelligen Licht, M-Zapfen im mittleren Wellenlängenbereich, und S-Zapfen (s = *short*) im kurzwelligen Bereich.

Die an den Rezeptoren in der Netzhaut absorbierte Energie wird in neurale, elektrochemische Signale umgewandelt, die dann an nachgeschaltete Neuronen, den Sehnerv und schließlich das Gehirn weitergeleitet werden. Aus dem Signalen der drei Zapfenarten entsteht im Gehirn dann ein Farbbild. Isaac Newton erkannte, dass Licht selbst keine Farbe hat, sondern nur in unserem Auge und im Gehirn existiert.

Der Trichromatismus des Menschen bedingt, dass mindestens drei Werte vorliegen müssen, um ein Farbbild numerisch beschreiben zu können. Die von drei Sensoren für die Primärfarben R, G, B erzeugten Antworten (Tripel) werden Tristimulus genannt [H.-C. Lee 2005; Jähne 2002].

Die internationale Lichtkommission CIE[1] hat bereits 1924 einen Standard für die spektrale Empfindlichkeit eines menschlichen Beobachters (Normalbetrachter) aus Untersuchungen an mehreren Probanden festgelegt, der mehrmals leicht angepasst wurde. Die spektralen Empfindlichkeiten sind immer noch Gegenstand der Forschung [H.-C. Lee 2005]. Beim skotopischen Sehen liegt das Maximum etwa bei einer Wellenlänge von 510 nm, beim photopischen Sehen bei etwa 555 nm.

4.1.2.1 Aufnahme von digitalen Farbbildern

Zur Aufnahme eines digitalen Farbbildes des sichtbaren Spektrums mit drei Kanälen (Rot, Grün und Blau) stehen folgende Möglichkeiten zur Verfügung:

[1] Commission Internationale de l'Éclairage

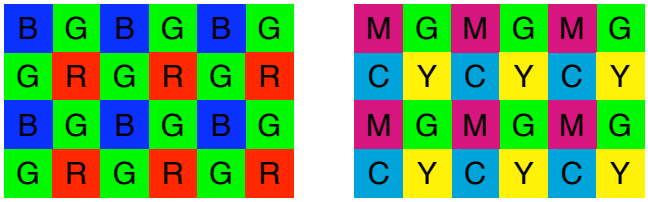
- Die Aufteilung des Lichtes über einen dichroitischen Prismenblock in Rot, Grün und Blau mit jeweils einem separaten CCD-Sensor pro Kanal. Damit stehen die drei Farbkanäle in voller Auflösung zur Verfügung.

- Der Foveon-X3-Sensor [Merrill 1998] verwendet drei übereinander liegende CMOS Sensoren und nutzt die unterschiedliche Eindringtiefe je nach Wellenlänge des Lichtes. Rotes, langwelliges, Licht hat eine größere Eindringtiefe in Silizium als blaues, kurzwelliges Licht.

- Das Farbfilterrad erlaubt zeitlich hintereinander folgende Aufnahmen mit jeweils entsprechender Filtereinstellung mit einem Ein-Chip-System. Für Farbbilder liegen die RGB-Kanäle jeweils in voller Auflösung vor. Farbfilterräder kommen z. B. in Kamerasystemen für astronomische Aufnahmen oder in Projektoren zum Einsatz.

- Farbfiltermuster über einem einzelnem Sensor sind am weitesten verbreitet, da sie gegenüber den anderen Verfahren am kostengünstigsten sind und deshalb auch in digitalen Farbkameras zum Einsatz kommen (z.B. die „DragonFly“ Kamera der Firma Point Grey).

  Vor jedem einzelnen Sensorelement befindet sich ein Farbfilter der jeweils nur eine Farbe durchlässt. Eine Bayer Filtermatrix (engl. *color filter array*, CFA) ist das am häufigsten angewandte Filtermuster. Es wurde von Bryce Bayer bei Eastman Kodak entwickelt [Bayer 1976]. Es filtert das einfallende Licht nach Rot, Grün und Blau. Eine mögliche Variante der Filteranordnung (BGGR) ist in Abb. 4.1a dargestellt. Das Mosaik aus 2×2 Filtern wiederholt sich über



(a) Bayer Filter Matrix **(b)** CMYG Filtermatrix

Abbildung 4.1: Bayer Farbfiltermatrix: (a) RGB Filter (BGGR), (b) CMYG Filter

  die gesamte Sensorfläche. Neben BGGR gibt es noch RGGB, GBGR, GRGB als mögliche Filteranordnungen. Grün bedeckt dabei immer 50 % in einem Schachbrettmuster, Rot und Blau bedecken jeweils 25 % der Sensorfläche. Die Bildhelligkeit (Luminanz) wird mit einer höheren Auflösung abgetastet als Farbton und Farbsättigung (Chrominanz). Grün dominiert, da es die Luminanz repräsentiert. Dies entspricht der Helligkeitsempfindlichkeit des mensch-

lichen Auges, die ein Maximum bei einer Wellenlänge von 550 nm hat, also im grünen Licht des sichtbaren Spektrums liegt [Ramanath u. a. 2002].

In [Gunturk, Glotzbach u. a. 2005] wird auf eine Alternative zum Bayer-RGB-Mosaik hingewiesen, die mit einem Cyan, Magenta, Gelb (Y) und Grün 2×2 Filtermosaik arbeitet (CMYG, siehe Abb. 4.1b). Das CMYG Mosaik ist in [Adams u. a. 1998] beschrieben. Der Vorteil gegenüber dem RGB-Filtermosaik ist eine höhere Lichtempfindlichkeit, da das Licht nur eine Filterschicht passieren muss. RGB-Filter basieren auf Kombinationen von CMY-Filtern. Ein Blau-Filter wird z. B. aus einer Kombination eines Cyan- und Magentafilters aufgebaut [Gunturk, Glotzbach u. a. 2005].

Da bei einem Farbfiltermuster-System an jedem Sensorelement nur ein spektraler Wert vorliegt, für ein Farbbild aber alle drei Kanäle benötigt werden, müssen die fehlenden Stellen in jedem Kanal interpoliert werden. Dieser Vorgang wird im Englischen *demosaicking* genannt.

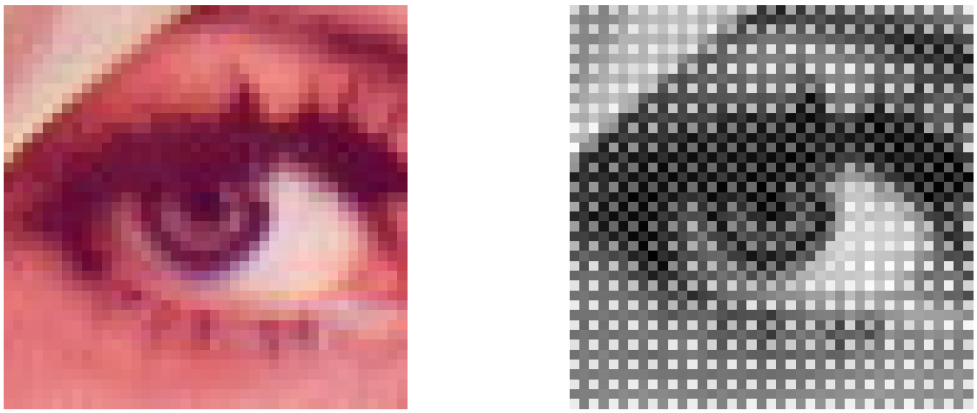
- „Pixelshift“ ist eine Variante , die aus zwei direkt hintereinander aufgenommenen Einzelbildern besteht. Vor der zweiten Aufnahme wird das durch die Farbfiltermatrix einfallende Licht mit einem Refraktionsblättchen um ein Sensorelement verschoben, so dass nach zwei Aufnahmen Grün in voller Auflösung vorliegt und Rot und Blau die Sensorfläche jeweils streifenförmig zu 50 % bedecken.

4.1.2.2 Bayer Demosaicking-Verfahren

Übersichten zu Bayer Demosaicking-Verfahren liefern [Gunturk, Glotzbach u. a. 2005] und [Ramanath u. a. 2002]. Oft liefern Farbkameras für die industrielle Bildverarbeitung nur das rohe, monochrome Bayermatrixbild (siehe Abb. 4.2, S. 46). Der Nutzer muss dann die Umwandlung in ein Farbbild auf dem Auswertesystem durchführen. Dies ist auch nötig, wenn nur Intensitäten verarbeitet werden sollen. Es muss immer der Umweg über ein Demosaicking-Verfahren genommen werden. Die Auswahl der hier untersuchten Verfahren wurde vor allem durch deren Verfügbarkeit in der Firewire-Kamera-API „libdc1394“ [Douxchamps 2014] beeinflusst.

Aus einem beliebigen digitalen RGB-Farbbild kann ein Bayermosaikbild generiert werden [Perko u. a. 2005]. Die Intensitäten aus den drei Farbkanälen müssen nur entsprechend dem gewünschten Bayermosaik in ein Grauwertbild übertragen werden (siehe Abb. 4.2, S. 46). Nach Anwendung eines Demosaicking-Verfahrens werden die Farbwerte des Originalbildes mit denen des berechneten Farbbildes verglichen, um die Leistung des jeweiligen Verfahren zu bewerten.

Die einzelnen Verfahren lassen lassen sich in adaptive und nicht-adaptive Verfahren einteilen [Chen 1999]. Zu den nicht-adaptiven Verfahren zählen z.B. die Nearest-Neighbor-, Bilineare- oder Bikubische-Interpolation. Jeder Kanal wird dabei für sich



(a) Original **(b)** Bayermosaik (BGGR)

Abbildung 4.2: Ausschnitt aus Originalfarbbild „Lena“ (Quelle: [SIPI 2015]) (a) und gleicher Ausschnitt des daraus generierten Bayermosaiks (b)

interpoliert, ohne die hohe Korrelation [Gunturk, Altunbasak u. a. 2002] der RGB-Kanäle untereinander zu berücksichtigen. Diese Verfahren arbeiten mit einer fest definierten Nachbarschaft um das jeweils zu interpolierende Pixel, während Adaptive Verfahren je nach lokalen Eigenschaften in der Nachbarschaft des interpolierenden Pixels zwischen mehreren angepassten Algorithmen auswählen können.

Beim Demosaicking kommt es in inhomogenen und hochfrequenten Bildteilen zu störenden Artefakten im interpolierten Farbbild, die sich als Moirémuster und als „Zipper“-Effekt [Lu und Tan 2003] bemerkbar machen. Der Zipper-Effekt beschreibt abrupte oder unnatürliche Änderungen in den Farben direkt benachbarter Pixel entlang einer Kante in einem abwechselnden Muster, das an einen „Reißverschluss“ (engl. *zipper*) erinnert (siehe Abb. 4.4, S. 48).

Die *Nearest-Neighbor*-Interpolation ist das einfachste Verfahren überhaupt. Durch das regelmäßige Farbfiltermuster sind die Nachbarschaften der zu interpolierenden Pixel fest. Die Grauwerte des Bayermatrixbildes müssen nur in die entsprechenden Kanäle übertragen werden. Bei diesem Verfahren ist der Zipper-Effekt am ausgeprägtesten (siehe Abb. 4.4, S. 48). In den hochfrequenten Bildanteilen wie z.B. im Bereich des Zauns neben dem Fernglas ergeben sich deutliche Farbartefakte. In der letzten Spalte und letzten Zeile liegen keine interpolierten Werte vor.

Die *Bilineare*-Interpolation erzeugt weniger Zipper-Effekte und Farbsäume. Die Kanten werden aber weniger scharf abgebildet. Das Bild hat einen ein Pixel breiten Rand, in dem keine interpolierten Werte vorliegen, bedingt durch die Nachbarschaft.

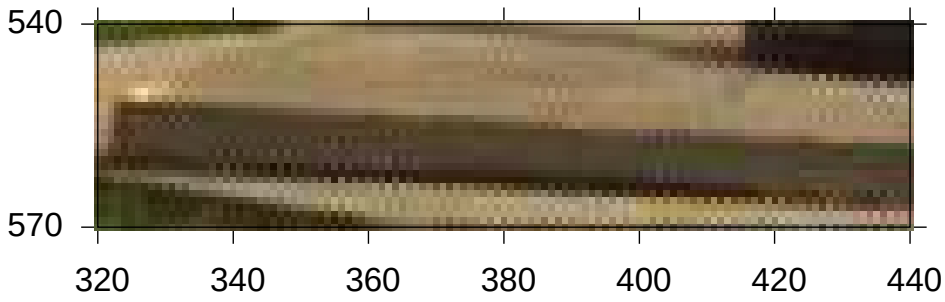
Das von [Malvar u. a. 2004] vorgestellte, lineare, adaptive Verfahren (*HQLinear*) berücksichtigt die Korrelation der RGB-Kanäle untereinander. Es wird davon ausgegangen, dass der Anteil der Bildhelligkeit entlang der Kanten stärker ausgeprägt ist als Farbwert und Sättigung.

Abbildung 4.3: Leuchtturmszene „kodim19“ aus [Kodak 1999]. Größe 512×768

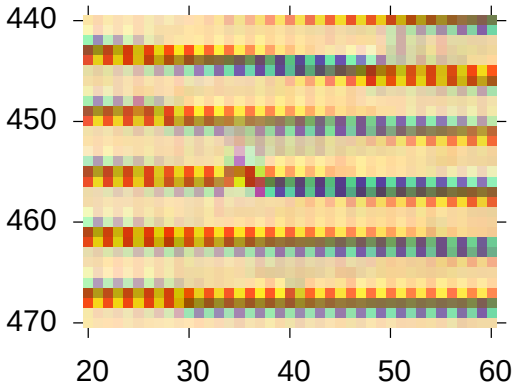
4.1.2.3 Verarbeitungszeit der Verfahren

Abb. 4.5 zeigt ein Vergleich einiger Verfahren, die in der frei verfügbaren Firewire-Kamera-API „libdc1394“ implementiert sind (in C). Die Rechenzeiten sind in der Grafik auf das Nearest-Neighbor Verfahren normiert. Auf dem getesteten System beträgt seine Rechenzeit im Mittel 0,1 ms für das Demosaicking des Leuchtturmbildes (Kodim19, 512×768). Es ist am schnellsten, da es keine arithmetischen Operationen enthält sondern nur Grauwerte in einer festen Vorschrift kopiert. Das adaptive Verfahren der variablen Anzahl von Gradienten (VNG) ist mit einer Rechenzeit von fast 20 ms das langsamste unter den getesteten. Die Verarbeitungszeit ist durch das adaptive Verfahren auch abhängig vom Bildinhalt. Bei einem Testdurchlauf mit allen 24 Bildern (900 mal je Bild) der „Kodak Lossless True Color Image Suite“ [Kodak 1999] ergaben sich Unterschiede in der Verarbeitungszeit von 6,3 ms ± 1 ms. Bei Originalbildern einer DragonFly Kamera (BGGR, 1024×768) beträgt die mittlere Rechenzeit bei VNG fast 40 ms. Bei einer Bildwiederholungsrate von 15 Bildern/s stehen nur 67 ms Rechenzeit insgesamt zwischen den Aufnahmen zur Verfügung. Mit einem Anteil von etwa 60 % Rechenzeit auf der CPU nur für das Demosaicking allein ist das VNG-Verfahren in der getesteten Implementierung für Echtzeitanwendungen nicht geeignet. Durch die Verwendung der GPU für das Demosaicking kann die Verarbeitungszeit jedoch reduziert werden [McGuire 2008].

[Perko u. a. 2005] zeigen, dass die deutlich sichtbaren Effekte unterschiedlicher Bayer-Verfahren keinen Einfluss auf die Geometrie der extrahierten Bildmerkmale haben.



(a) Ausschnitt 1: Fernglassockel



(b) Ausschnitt 2: Hauswand

Abbildung 4.4: „Zipper“-Effekt beim Nearest-Neighbor Verfahren. Ausschnitte aus Leuchtturmbild („kodim19“)

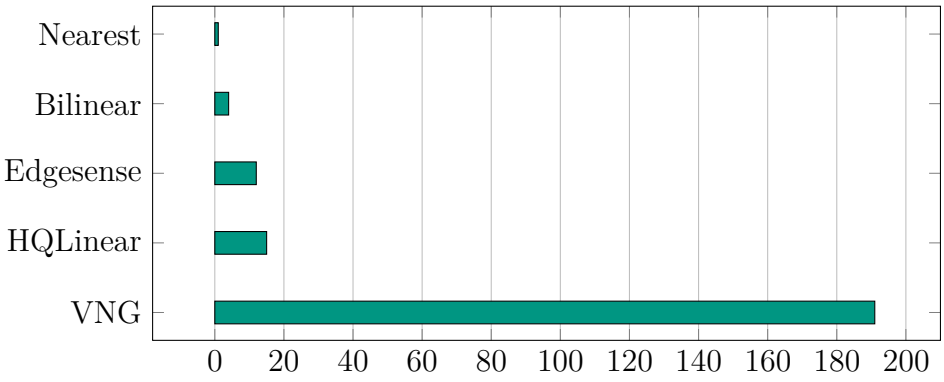


Abbildung 4.5: Gegenüberstellung der Rechenzeiten der einzelnen Demosaicking-Verfahren. Mittelwerte aus $4 \cdot 100$ Durchläufen pro Verfahren; auf Nearest-Neighbor normiert

4.2 Die Stereokamera

Eine Stereokamera ist ein System bestehend aus zwei Kameras mit idealerweise identischen inneren Orientierungen, die über eine feste, bekannte Basis b miteinander verbunden sind. Neben der bekannten inneren Orientierung der beiden Kameras wird bei einem Stereokamerasystem auch die relative Orientierung als bekannt vorausgesetzt.

Wenn Epipolarbilder vorliegen oder die Kameras im Stereonormalfall mit parallelen optischen Achsen zueinander orientiert sind, dann können die x-Parallaxen p_x direkt entlang des horizontalen Versatzes der Bildkoordinaten eines homologen Merkmals im linken (x') und rechten Bild (x'') bestimmt werden: $p_x = x' - x''$. Die Parallaxe p_x wird auch Disparität (eng. *disparity*, entsprechend hier $d == p_x$) oder Deviation genannt.

Für den Abstand z in [Pixel] von der Basis b des Stereokamerasystems gilt:

$$z = \frac{c \cdot b}{d} \quad \text{[Pixel]}. \tag{4.8}$$

Für den Abstand z in [mm] muss die Sensorelementbreite s bekannt sein:

$$z = \frac{c \cdot b}{d \cdot s} \quad \text{[mm]}. \tag{4.9}$$

Da alle anderen Größen konstant sind, ist die Disparität somit ein Maß für die Raumtiefe eines 3D-Punktes. Die Tiefe ist umgekehrt proportional zur Disparität. $1/z$ wird als inverse Tiefe bezeichnet.

Die vollständigen 3D-Objektkoordinaten $[X, Y, Z]^T$ können einfach aus den Epipolarbildmessungen berechnet werden:

$$X = \frac{b \cdot x'}{d} \qquad Y = \frac{b \cdot y'}{d} \qquad Z = \frac{b \cdot c}{d}. \tag{4.10}$$

Der Disparitätsbereich $d_{\max}$ beschränkt den minimal und maximal möglichen Tiefenbereich zwischen $\boldsymbol{Z}_{\min}$und $\boldsymbol{Z}_{\max}$ wie es in der vereinfachten Abb. 4.6 für den achsparallelen Stereofall dargestellt ist. Dieser Bereich ist abhängig von der Basis b und dem maximalen horizontalen Öffnungswinkel der Stereokameras. Der Öffnungswinkel ist wiederum abhängig von der Kamerakonstante c. Der mit Tiefenwerten erfasste Raum zwischen

$$\boldsymbol{Z}_{\min} = \frac{c \cdot b}{d_{\max}} \quad \text{und} \quad \boldsymbol{Z}_{\max} = \frac{c \cdot b}{d_{\min}} \tag{4.11}$$

wird Horopter genannt (siehe auch Abb. 4.8, S. 52).

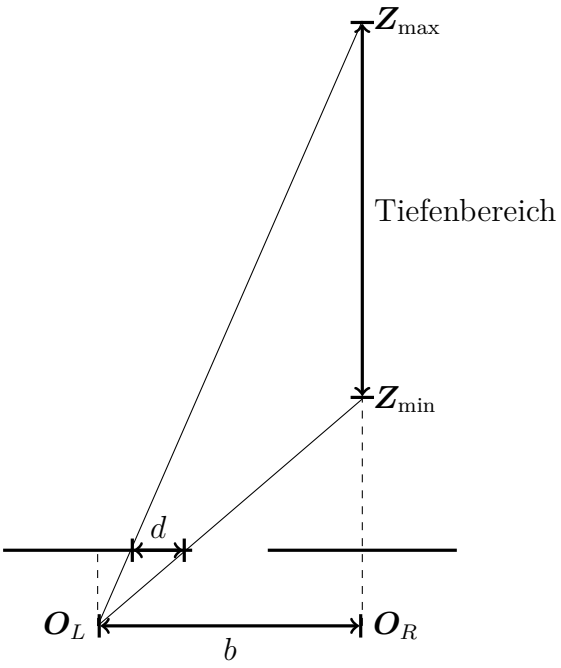


Abbildung 4.6: Beschränkung des Tiefenbereiches durch die Parallaxe. Nach [Schreer 2005]

4.2.1 Geometrie und Genauigkeitscharakteristiken der Entfernungsberechnung

Größere Disparitätswerte d führen zu kürzeren Abstanden Z. Liegt der 3D-Punkt im Unendlichen oder direkt auf dem Epipol ist die Disparität $d = 0$ und es können keine Tiefenwerte berechnet werden.

Fehler in der Disparität Δd führen zu Fehlern in den Tiefenwerten ΔZ:

$$\Delta Z = \frac{\partial Z}{\partial d}\Delta d = \frac{Z^2}{b \cdot c}\Delta d \tag{4.12}$$

Mit zunehmendem Objektabstand steigt die Unsicherheit beim Stereonormalfall quadratisch. In Abb. 4.7 ist die Tiefengenauigkeit ΔZ in Abhängigkeit zur Tiefe Z für eine typische Stereokamera mit Basis b =12 cm für zwei Brennweiten (4 mm und 2 mm) dargestellt. Die Merkmale werden mit einer hypothetischen horizontalen Genauigkeit von 0,1 Pixel gemessen. Die rechte Darstellung zeigt die drei markierten Abstände im Graphen (5 m, 10 m, 20 m) als konzentrische Kreise um einen Standpunkt an. Dies soll zeigen, wie viele Objekte sich innerhalb dieser Abstände noch in einer typischen Straßenszene befinden. In [Montiel u. a. 2006] und [Paz u. a. 2008] zeigen Simulationen mit verrauschten Bildbeobachtungen in einem synthetischen Stereonormalfall, dass die daraus resultierenden Objektpunkte in der XZ-Ebene keine elliptische Verteilung haben. In Z-Richtung gibt es deutliche Ausreißer.

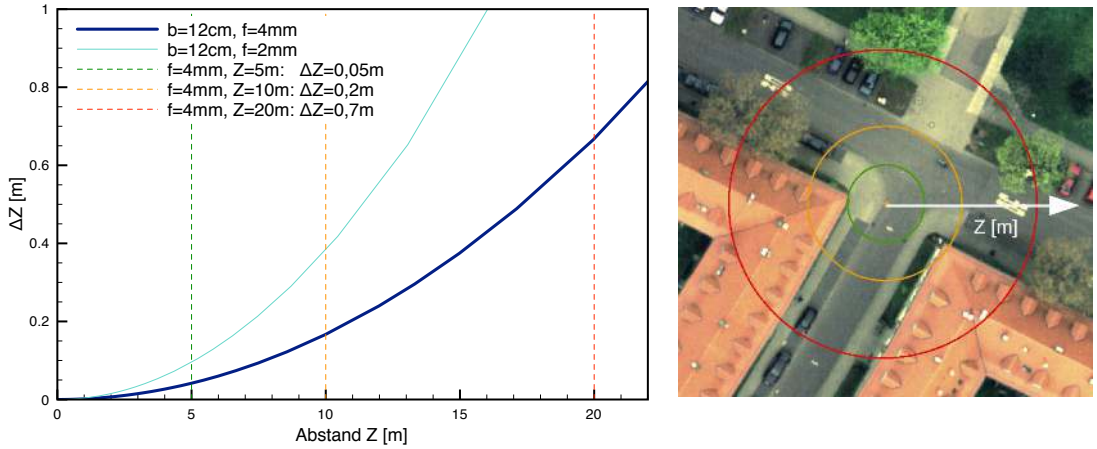


Abbildung 4.7: Tiefengenauigkeiten einer Stereokamera mit einer Basis $b = 12\,\text{cm}$ für zwei verschiedene Brennweiten. Drei ausgesuchte Tiefenwerte (5 m, 10 m, 20 m) sind in einer Straßenszene zum Vergleich dargestellt. Der gedachte Standpunkt befindet sich im Mittelpunkt der Kreise

Für die Umrechnung von Tiefenwerten $[x, y, d, 1]^T$ in homogene 3D-Punkte mit der Basis b, dem Hauptpunkt h_x, h_y und der Kamerakonstante c gilt:

$$\begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -h_x \\ 0 & 1 & 0 & -h_y \\ 0 & 0 & 0 & c \\ 0 & 0 & \frac{-1}{b} & \frac{h_x - h'_x}{b} \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} \tag{4.13}$$

wobei h'_x der x-Hauptpunkt des rechten Bildes ist. Schneiden sich die Hauptstrahlen im Unendlichen, dann wird $h_x = h'_x$ und somit der Term $\frac{h_x - h'_x}{b} = 0$.

4.2.2 Automatische Stereoanalyse

Die Stereoanalyse berechnet aus den automatisch gefundenen, homologen Bildmerkmalen zweier Epipolarbilder Tiefen- oder Disparitätsbilder. Hauptschwerpunkt bildet die Korrespondenzanalyse zum Auffinden möglichst vieler homologer Bildmerkmale im Überlappungsbereich des Stereobildpaares. Ziel ist ein möglichst dichtes Tiefenbild zu erhalten.

Scharstein und Szeliski haben 2002 die „Middlebury Stereo Evaluation“ Sammlung [Scharstein und Szeliski 2002] vorgestellt. Sie enthält Stereobildpaare mit Referenzbildern für die Disparitäten. Seither werden dort die zahlreichen dichten Stereoverfahren gesammelt und deren Ergebnisse sowohl mit den Referenzbildern, als auch die Verfahren untereinander, verglichen.

Probleme bei der dichten Tiefenbildberechnung entstehen durch

- Verdeckungen und Diskontinuitäten → keine Tiefenwerte
- Periodische Muster, die nicht eindeutig zugeordnet werden können → falsche Tiefenwerte
- Schwach texturierte Bereiche (glatte Flächen, Schatten, Spitzlichter)→ keine Tiefenwerte
- Perspektivische Verzerrungen.

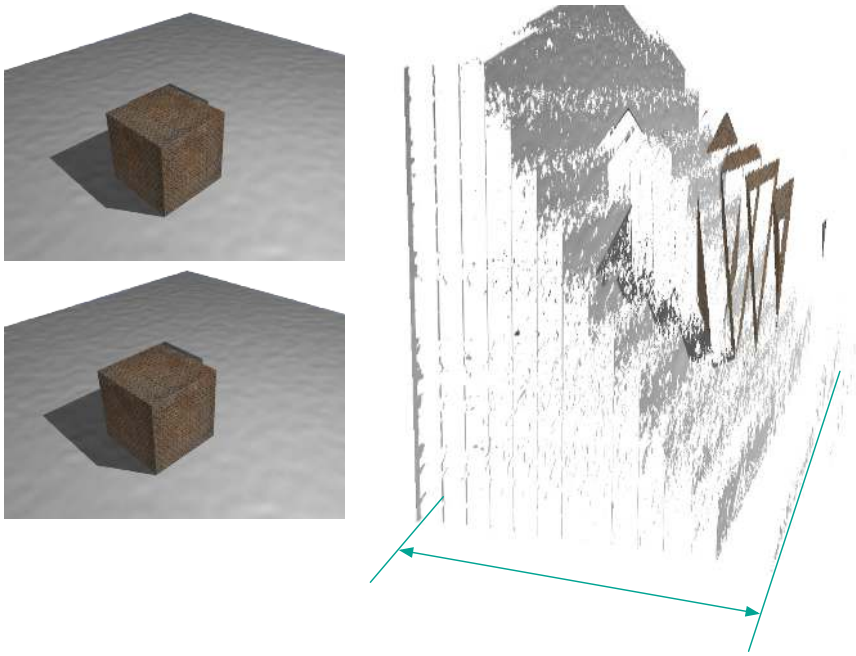


Abbildung 4.8: Horopter: Tiefenbereich (↔) zwischen $\boldsymbol{Z}_{\min}$und $\boldsymbol{Z}_{\max}$. (links: synthetisches Stereobildpaar)

In [Rodriguez und Aggarwal 1990] und [Blostein und T. S. Huang 1987] leiten die Autoren den Quantisierungsfehler bei Stereosystemen her. Dieser entsteht durch die Quantisierung der Disparitätswerte in ganzzahlige Integerwerte. Dieser Fehler kann maximal 0,5 Pixel betragen. Der Tiefenfehler hat nach [Rodriguez und Aggarwal 1990] folgenden Wertebereich:

$$0 \leqq |\Delta Z| \leqq \frac{Z_{\max}^2 \delta}{b \cdot c - Z_{\max} \delta} \tag{4.14}$$

wobei δ die effektive Tiefenauflösung ist.

4.3 Die Kinect Tiefenbildkamera

Die Firma PrimeSense hat ein Hardwarereferenzdesign („PrimeSensor 1.08“) und die Software (PrimeSensor NITE, NI=Natural Interaction) für das Pose-Tracking mehrerer Personen entwickelt. Der bekannteste Abkömmling des Referenzdesigns ist die Microsoft Kinect Kamera (siehe Abb. 4.9, S. 53) in der ersten Version. Sie ist eine Erweiterung zur Ganzkörpersteuerung für die Xbox Spielkonsole aus gleichem Hause. Der Sensor liefert Tiefenbilder (siehe Abb. 4.10b, S. 54) und RGB-Bilder (siehe Abb. 4.10a, S. 54) in VGA-Bildgröße mit einer Bildwiederholungsrate von 30 Hz. Für die Kombination beider Bilder hat sich der Name „RGB-D“ durchgesetzt.

Im Standfuß befindet sich ein steuerbarer Motor zum vertikalem Schwenken der Kamera. Ein dreiachsiger Beschleunigungssensor im Kameragehäuse dient zum Bestimmen des durch den Motor eingestellten Neigungswinkels.

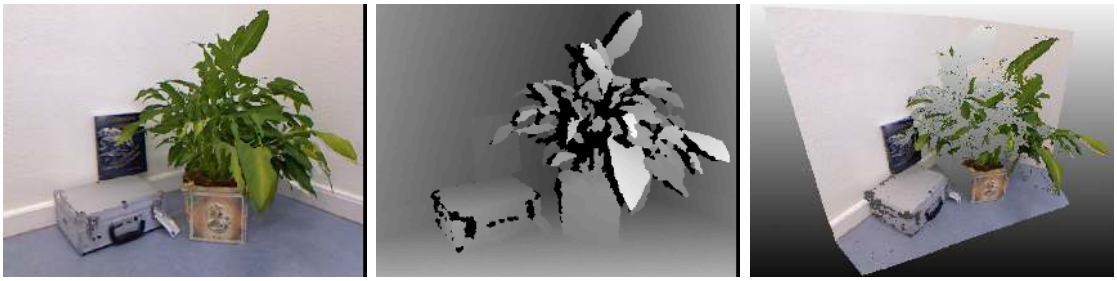
(a) Kinect Sensor

(b) Erkennung der Pose mehrerer Personen (NITE SDK)

Abbildung 4.9: Der Microsoft Kinect Sensor

Kurze Zeit nach dem Verkaufsstart des Kinect Sensors wurden erste Bibliotheken (Projekt *libfreenect*) zum Auslesen der Sensordaten an gewöhnlichen PCs durch Reverse Engineering [Fried 2010] der USB-Schnittstelle veröffentlicht. Mittlerweile gibt es auch offizielle Open-Source-Treiber von PrimeSense zusammen mit dem NITE-SDK (OpenNI) sowie ein offizielles SDK von Microsoft selbst.

Abb. 4.9b zeigt ein Tiefenbild eines Raumes mit drei Personen, die alle als solche erkannt worden und mit unterschiedlichen Farben hervorgehoben worden sind. Die blaue Person in der Mitte nimmt eine Kalibrierstellung ein, damit das NITE-System die Pose einpassen kann. Die Pose besteht aus einem einfachen „Skelett“ mit Kopf, Rumpf, Armen und Beinen. Die Raumkoordinaten der 15 Skelettpunkte einzelner Personen können mit 30 Hz direkt ausgelesen und zur interaktiven Programmsteuerung genutzt werden. Auch Handgesten wie z. B. „Hand heben“ oder „Hand zur Seite bewegen“ [*PrimeSense NITE Algorithms 1.5* 2011] können automatisch erkannt werden.



(a) Farbbild der RGB-Kamera (Verzeichnungs-korrigiert)

(b) Tiefenbild

(c) 3D-Punktwolke mit Farbwerten

Abbildung 4.10: Aus Farbbild (4.10a) und Tiefenbild (4.10b) wird eine 3D-Punktwolke mit Farbwerten erzeugt

4.3.1 Funktionsweise

Die folgenden technischen Details stammen von verschiedenen Veröffentlichungen [Konolige und Mihelich 2011; Smisek u. a. 2013; Fossati u. a. 2013; Kramer u. a. 2012; Mutto u. a. 2012; Jean 2012] zur Untersuchung der Kinect Eigenschaften. Neben einem PrimeSense-Patent [Freedmann u. a. 2010] sind keine näheren technischen Informationen über die Kinect Kamera offiziell erhältlich. Die ersten bekannten Details stammen aus einen Wiki-Beitrag [Konolige und Mihelich 2011] des ROS (Robot Operation System), der von Kurt Konolige und Patrick Mihelich verfasst wurde. Konolige ist u.a. für die Entwicklung von echtzeitfähigen Stereosystemen [Konolige 1997] für SLAM-basierte Robotersteuerung [Konolige, Agrawal u. a. 2007] bekannt.

Nicolas Burrus hat die im ROS-Wiki und ROS-Foren diskutierten Interna in einem quelloffenen Softwarepaket [Burrus 2014] der Allgemeinheit zur Verfügung gestellt. Das Auslesen der Kinect-Daten, (wie z. B. die Szene in Abb. 4.10) und die Kalibrierung wurden mit dieser Software durchgeführt.

Das Tiefenbild wird mit einer Kombination aus aktiver Lichtquelle im nahen Infrarot mit Musterprojektion und einer Kamera mit IR-Filter erzeugt. Der IR-Projektor (1 in Abb. 4.11) projiziert dazu mit ein festes Muster aus hellen und dunklen Punkten auf die Szene. Als Lichtquelle dient ein Laser mit einer Wellenlänge von 830 nm und einer Leistungsaufnahme von 70 mW [Kramer u. a. 2012]. Das Muster wurde in [Martinez und Stiefelhagen 2013] detailliert untersucht. Es soll für alle Kinect Sensoren gleich sein und wird durch spezielle optische Gitter erzeugt, die nur in einem Patent [Freedmann u. a. 2010] beschriebenen sind.

Ein erstes optisches Gitter teilt den Strahl des Projektors in eine Anordnung von 3×3 Strahlen. Diese neun Hauptstrahlen heben sich im Muster in Abb. 4.14 als helle Punkte ab. Die optische Verzeichnung der IR-Projektorlinse ist deutlich erkennbar. Das zweite optische Gitter erzeugt für jeden dieser Hauptstrahlen ein Muster von 211×165 Punkten. Die Ränder zwischen den einzelnen Gittern sind ebenfalls in 4.14

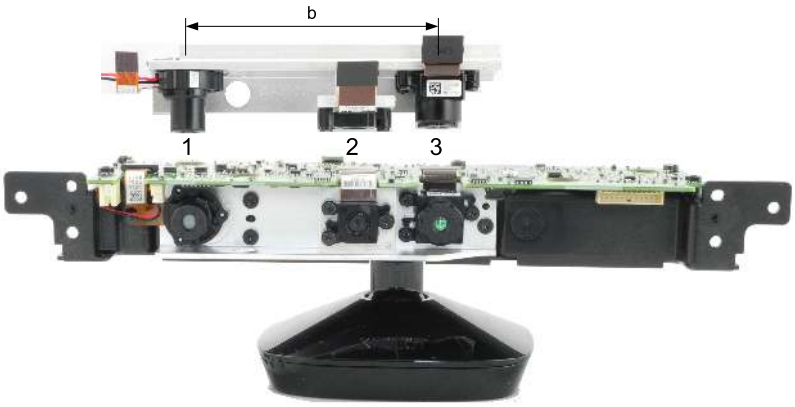


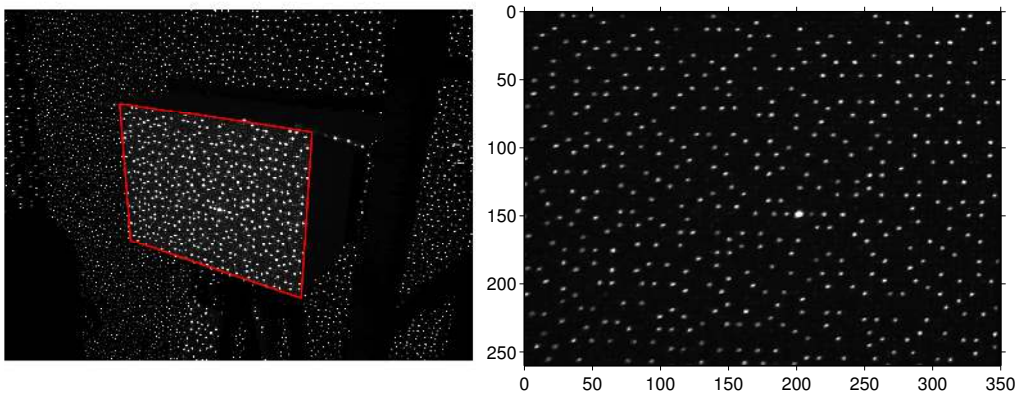
Abbildung 4.11: Microsoft Kinect Sensor geöffnet. 1: IR-Projektor, 2: RGB-Kamera, 3: IR-Kamera. Kameras und Projektor sind auf einer festen Aluminiumbasis montiert ($b = 7,52\,\text{cm}$)

erkennbar (siehe Pfeile am Rand). [Smisek u. a. 2013] haben etwa 800 Punkte entlang der horizontalen Mittellinie gezählt.

Zur Untersuchung des IR-Musters wurde dessen Projektion auf eine Schachtel in 2,77 m Entfernung mittig zur Kinect Kamera mit der IR-Kamera einer zweiten Kinect Kamera aufgenommen (siehe Abb. 4.12a, S. 56). Anschließend wurde die Vorderseite der Schachtel projektiv entzerrt (siehe Abb. 4.12b, S. 56). Auf dieser Fläche wurden insgesamt 457 Punkte detektiert. Der Hauptstrahl in der Mitte der Fläche hebt sich deutlich von den restlichen Punkten ab.

Die Punkte haben eine scheinbar zufällige Verteilung. Ihre Lage scheint in einem Raster von 79×59 mit 4,4 mm Kantenlänge (bzw. 0,09°) auf der untersuchten Fläche angeordnet (siehe Abb. 4.13, S. 56). Die rasterartige Struktur konnte durch Spektral- und Histogrammanalyse ermittelt werden. Unter der Annahme, dass die Öffnungswinkel von IR-Projektor und IR-Kamera mit 57° horizontal und 45° vertikal identisch sind, hätte das Gitter eine Ausdehnung von 626×500. Diese Beobachtung deckt sich einigermaßen mit [Martinez und Stiefelhagen 2013]. Wendet man das Verhältnis von hellen Punkten zu Gitterpunkten der Untersuchungsfläche (457/4661) auf die gesamte Fläche an, dann ergäbe das eine Gesamtpunktzahl von ≈30688 Punkten bei ≈313000 Gitterpunkten.

Die mit dem Muster beleuchtete Szene wird von der IR-Kamera (3 in Abb. 4.11) mit einer Bildgröße von 1280×1024 Bildpunkten mit 30 Hz aufgenommen. IR-Projektor und IR-Kamera haben einen Basisabstand von b=7,52 cm, gemessen am inneren Aufbau (siehe Abb. 4.11, S. 55). Die RGB-Kamera (2 in Abb. 4.11) ist zusammen mit IR-Projektor und -Kamera auf einer einer Aluminiumleiste montiert und befindet sich etwa 5 cm vom IR-Projektor und etwa 2,5 cm vom der IR-Kamera entfernt. Die technischen Daten der Kameras sind in Tabelle 4.4 zusammengefasst.



(a) IR-Muster auf Untersuchungsfläche. Der Hauptstrahl hebt sich deutlich von den restlichen Punkten ab

(b) Projektiv entzerrter Ausschnitt des IR-Musters auf der Untersuchungsfläche (350×260 mm)

Abbildung 4.12: IR-Muster

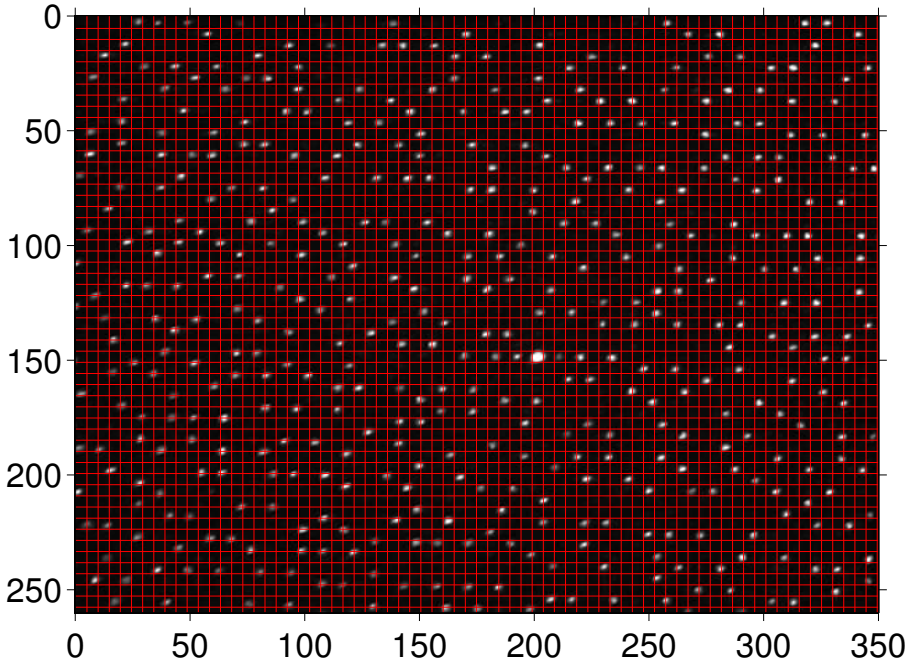


Abbildung 4.13: Ausschnitt des IR-Musters auf der Untersuchungsfläche: Anordnung der hellen Punkte in einem regelmäßigen Raster

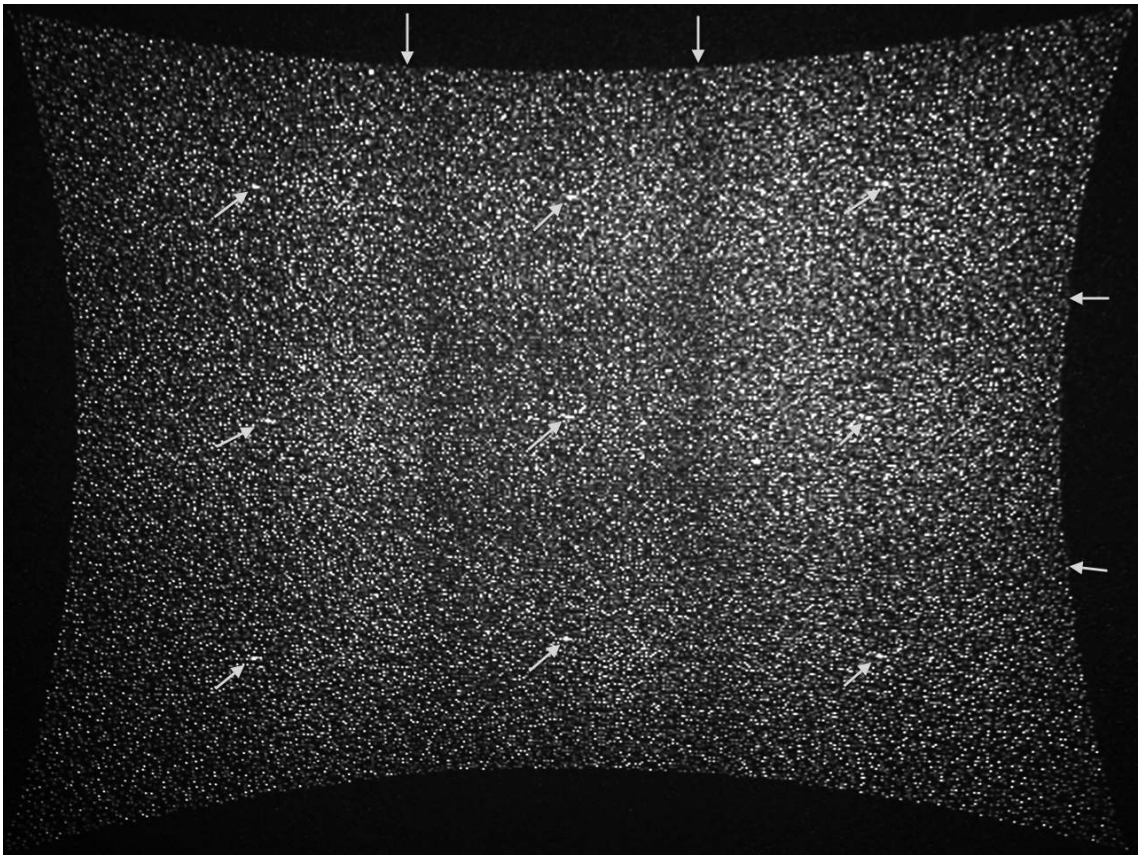


Abbildung 4.14: Aufnahme des projizierten IR-Musters auf eine glatte Wand. Man kann die neun Hauptstrahlen und die 3×3-Aufteilung erkennen. Das IR-Muster wurde mit einer zweiten Kinect IR-Kamera aufgenommen

Die Tiefenwerte werden über ein Korrelationsfenster aus dem IR-Bild berechnet. Wegen des acht Pixel breiten rechten Randes wird eine Fensterbreite von neun Pixel angenommen [Konolige und Mihelich 2011]. Bei einem 9×9 großen Korrelationsfenster würde das erste Pixel mit einem Disparitätswert an der Stelle $(u, v) = (5, 5)$ beginnen. Die ersten und die letzten vier Spalten und Zeilen wären nicht mit Werten besetzt. Der rechte Rand des Tiefenbildes stützt diese Theorie, wenn man davon ausgeht, dass die Tiefenwerte direkt in die erste Spalte geschrieben werden. Die Zeilen sind jedoch voll besetzt.

Dies kann mit der verwendeten IR-Kamera (Aptina MT9M001, siehe Tab. 4.4) erklärt werden: Bei einer Bildgröße von 1280×1024 und einem Zusammenfassen benachbarter Pixel (2×2 binning) blieben 640×512 Pixel übrig. In der Breite führt dies zu dem beschriebenen Rand. In der Höhe sind aber mit 512-8>480 noch mehr Zeilen mit Tiefenwerten vorhanden als ausgelesen werden. Aus diesem Grund kann auch keine Aussage über die Höhe des Korrelationsfensters gemacht werden. [Khos-

helham und Elberink 2012] haben einen Versatz von vier Pixel zwischen IR- und Disparitätsbild beobachtet, der diese These unterstützt.

Das Muster ist intern für einen Referenzabstand gespeichert. Der Versatz der Parallaxe für die Referenzfläche ist $d_{\text{offset}} \approx 1090$. Die Berechnung erfolgt intern auf dem von PrimeSense entwickelten Soc-System (System on a Chip) in der Kamera und wird über die USB-Schnittstelle mit 11 bit ausgelesen. Für die Tiefenwerte die Tiefenbilds stehen damit maximal 2048 Werte zur Verfügung. Aus Tiefen- und RGB-Bild erzeugte 3D-Farbpunktwolken (siehe Abb. 4.10c, S. 54) haben somit bis zu $(640 - 8) \times 480 = 303360$ Punkte bei einer Aufnahmefrequenz von 30 Hz.

4.3.2 Berechnung der Tiefenbilder

Die Tiefenwerte werden wie beim Stereonormalfall berechnet:

$$z = \frac{b \cdot f}{d} \tag{4.15}$$

mit z = Tiefe in $[m]$, f = Brennweite, d = Parallaxe (Disparität). Die Beziehung zwischen normalisierter Disparität und „rohen“ Disparitätswerten der Kinect ($\frac{1}{8}$ Pixel) wird über den gerätespezifischen Offset d_{offset} hergestellt:

$$d = \frac{1}{8} \cdot (d_{\text{offset}} - d_{\text{kinect}}). \tag{4.16}$$

Damit folgt für z:

$$z = \frac{b \cdot f \cdot 8}{d_{\text{offset}} - d_{\text{kinect}}}. \tag{4.17}$$

Auch bei der Kinect Kamera nimmt der Fehler der Tiefenwerte mit dem Abstand quadratisch zu.

4.3.3 Das Tiefenbild

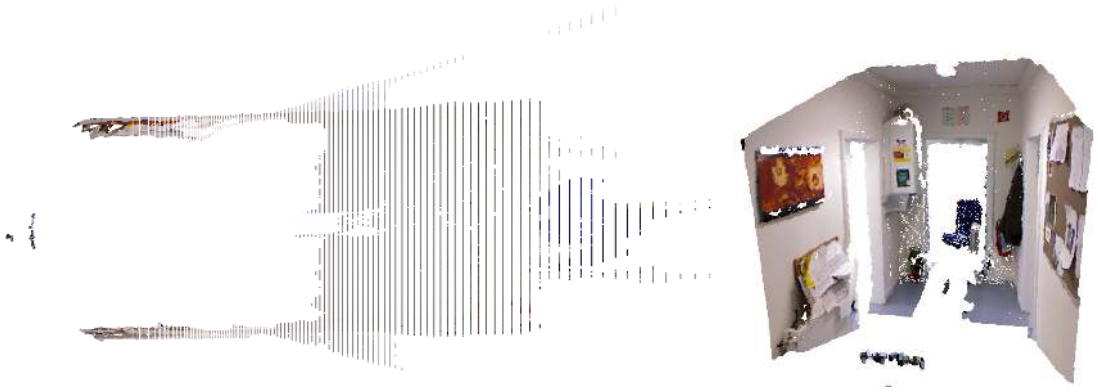
Der Tiefenbereich ist vom Hersteller zwischen 0,8 m bis 4 m definiert. Die Auflösung soll in 2 m Abstand bei 3 mm für x, y und 1 cm für z liegen.

In [Smisek u. a. 2013; Khoshelham und Elberink 2012] wurde die Tiefenauflösung q in Abhängigkeit zur Tiefe z, die aufgrund der Quantisierung der Tiefwerte mit 11 bit entsteht (2048 Werte) ermittelt. Dabei wurden in [Smisek u. a. 2013] die Tiefenwerte einer Fläche im Abstand von 0,5 m bis 15 m von der Kinect Kamera gemessen. Durch die so gemessenen Tiefenwerte wurde ein Polynom zweiten Grades gelegt:

$$q(z) = 2,73z^2 + 0,74z - 0,58 \tag{4.18}$$

z [m]	$q(z)$ [m]	Δz [m]	Präzision
0,5	0,007	-	0.0001
2,0	0,012	-	0.0005
5,0	0,070	0,04	-

Tabelle 4.3: Tiefenauflösung $q(z)$ und Genauigkeiten dz für verschiedene Tiefen z [Konolige und Mihelich 2011; Smisek u. a. 2013; Khoshelham und Elberink 2012]



(a) Ansicht von oben **(b)** Schrägansicht

Abbildung 4.15: Eine Kinect Punktwolke eines Ganges in zwei Ansichten. In (a) kann man die Quantisierung der Disparitäten anhand des zunehmenden Abstandes der Punktebenen von links nach rechts erkennen

Den Effekt der Quantisierung kann man deutlich in den Kinect Punktwolken erkennen (siehe auch Abb. 4.15a).

Unterschiedliche Eigenschaften der Oberfläche und der Umgebung beeinflussen das Ergebnis des Kinect Tiefenbildes. Folgende Situationen führen zu einem höheren Rauschen oder gar zu fehlenden Messwerten:

- Tageslicht; im Freien selbst im Schattenbereich Probleme
- Glänzende bzw. spiegelnde Oberflächen
- Kleine Strukturen, die nicht genügend Oberfläche für die Musterprojektion bieten
- Flächen mit zu spitzen Winkeln zur Kamera

Im Folgenden wurden verschiedene statische Szenen mit mindestens 100 Aufnahmen beobachtet (siehe Abb. 4.16, S. 60). Für die Tiefenbildstapel wurden jeweils Mittelwert, Standardabweichung und die maximale Anzahl gültiger Pixel bestimmt.

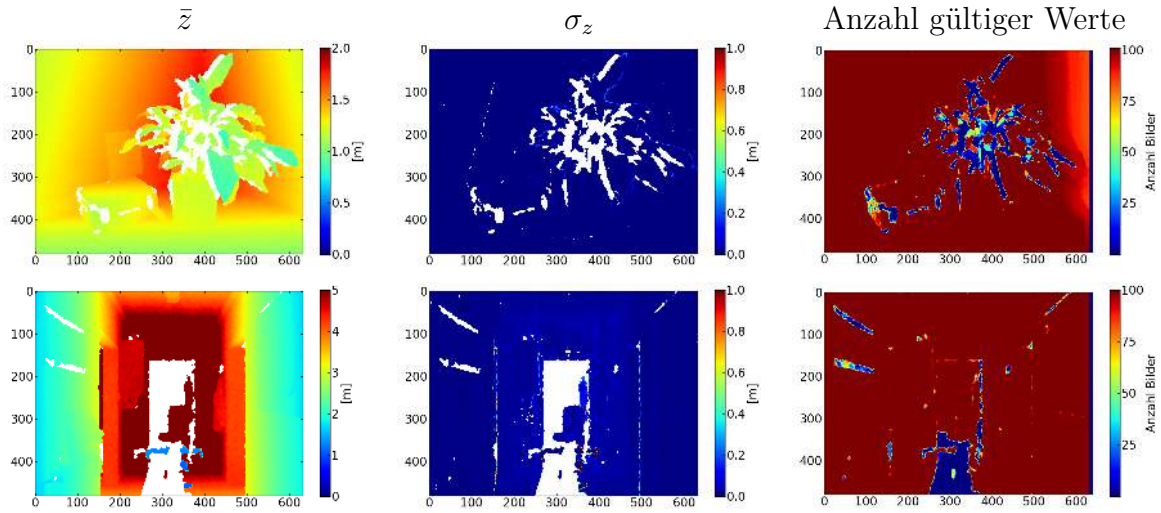
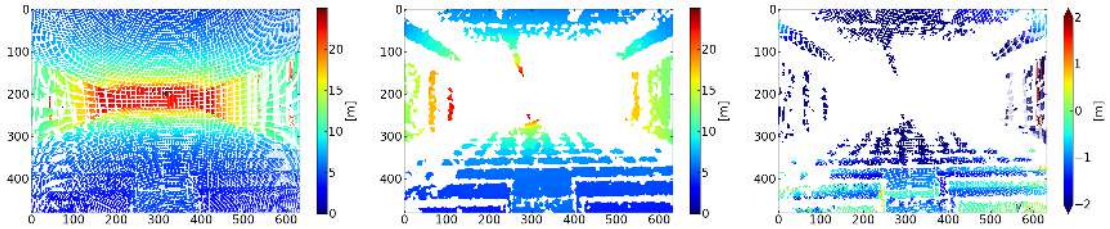


Abbildung 4.16: Auswertung mit je 100 Aufnahmen einer statischen Szene. (obere Reihe: siehe Abb. 4.10, untere Reihe: 4.21a)

Ungültige Pixel haben keinen Tiefenwert. Die Tiefenbilder zeigen Bereiche ohne Tiefenwerte, die durch Verdeckungen entstehen. Diese entstehen durch den Basisabstand von IR-Projektor und IR-Kamera. An Tiefenkanten haben die Werte ein höheres Rauschen. Auf bestimmten Oberflächen kann das projizierte Muster schlecht oder gar nicht erkannt werden, wie z. B. die spiegelnde Schnallen des Koffers (siehe Abb. 4.10 zum Vergleich).

In [Weinmann u. a. 2011] wurde die maximale Reichweite der Tiefenwerte in einem Hörsaal gemessen und mit Werten eines Laserscanners verglichen (siehe Abb. 4.17, S. 60). Die Kinect Tiefenwerte liegen zwischen 3,61 m und 23,86 m - weit über den empfohlenen Bereich hinaus.



(a) z Laserscanner **(b)** $\bar{z}$ Kinect **(c)** Differenzen

Abbildung 4.17: Auswertung mit über 100 Aufnahmen eines Hörsaals [Weinmann u. a. 2011]

Zu Kalibrierung und Genauigkeit der Kinect Kamera gibt es mittlerweile mehrere Arbeiten [Wujanz u. a. 2011; Khoshelham 2011; Smisek u. a. 2013; Mutto u. a. 2012; Gonzalez-Jorge u. a. 2013].

Kamera	Eigenschaft	Wert
RGB-Kamera	Hersteller, Typ[3]	Aptina, MT9M112214STM
	Blickfeld[2] (H×V)	63°×50°
	Brennweite[2]	2,9 mm
	Sensorelementgröße[2]	2,8 µm
	Sensortyp[1]	CMOS, Bayer-Muster, Rolling Shutter[5]
IR-Kamera	Hersteller, Typ[3]	Aptina, MT9M001C12STM
	Blickfeld[2]	57°×45°
	Bildgröße[2]	1280×1024
	Brennweite[2]	6,1 mm
	Sensortyp[1]	CMOS
	Sensorelementgröße[4]	5,2 µm
Tiefenbild	Größe[1]	VGA (640×480)
	Pixelgröße in 2 m Entfernung[1]	3 mm
	Auflösung der Tiefe in 2 m Entfernung[1]	1 cm
	Max. Bildwiederholungsrate[1]	30 Hz
	Spezifizierter Tiefenbereich[1]	0,8 m-3,5 m
	Bit-Tiefe	11 bit

Tabelle 4.4: Technische Spezifikationen des Kinect Sensors. Werte aus [1]PrimeSense 1.08 Referenzdesign, [2][Smisek u. a. 2013], [3]Suche nach „Kinect“ bei `https://chipworks.secure.force.com`, [4] Kameramodulspezifikation, [5][Sturm, Engelhard u. a. 2012]

Die Kinect hat, wie alle elektronischen Messgeräte einen Temperaturgang. In [Fiedler und Müller 2012] wurde die Temperaturabhängigkeit der Kinect Kamera untersucht. Erst nach etwa zwei Stunden Betrieb bleiben die Tiefenwerte konstant.

[Sturm, Engelhard u. a. 2012] haben einen Zeitversatz zwischen RGB- und Tiefenbild analysiert. Dieser ist mit 20 ms recht hoch, wenn man die Bildwiederholrate von 30 Hz in Betracht zieht.

Abschließend sind die aus verschiedenen Quellen ermittelten technischen Daten der einzelnen Kinect-Sensoren in Tabelle 4.4 zusammen gestellt.

4.4 Andere Tiefenbildkameras

Mit den Time-of-Flight-Kameras (ToF) gibt es einen weiteren Ansatz für Tiefenbild- bzw. 3D-Kameras (siehe Tab. 4.5). Die Szene wird mit einem monochromatischen, sinusförmigen Licht konstanter Wellenlänge beleuchtet. Pro Sensorelement wird die Laufzeit gemessen, die das Licht von der Lichtquelle zum Objekt und wieder zurück zum Sensorelement benötigt. Als Sensor kommt ein CCD oder CMOS zum Einsatz.

Die Distanz R ist abhängig von der gemessenen Phasendifferenz $\Delta\varphi$ und der Modulationsfrequenz f_m. $\Delta\varphi$ wird aus vier benachbarten Subpixeln bestimmt, die vier unterschiedliche Intensitäten der Amplitude A mit einer Phasendifferenz von 0°,90°, 180° und 270° messen [Jutzi 2010]. Mit

$$\Delta\varphi = \arctan(\frac{A_{270} - A_{90}}{A_0 - A_{180}}) \tag{4.19}$$

wird die Distanz zu

$$R = \frac{c}{2f_m}\frac{\Delta\varphi}{2\pi}, \tag{4.20}$$

wobei c die Lichtgeschwindigkeit ist.

Die Phasendifferenz ist nur bis 2π eindeutig. Bei einer PMD CamCube 2.0 Kamera ist die maximale, eindeutige Reichweite bei einer Modulationsfrequenz von $f_m = 18$ auf $R = 8,33\,\text{m}$ begrenzt. Mit Hilfe verschiedener Modulationsfrequenzen kann die Mehrdeutigkeit aufgelöst werden und somit die Reichweite erhöht werden [Jutzi 2009; Jutzi 2010]. In [Weinmann u. a. 2011] wurde gezeigt, das man auf diese Weise mit einer CamCube Kamera Entfernungen von 25 m messen kann.

Kameramodel	Bildgröße [Pixel]	Framerate [Hz]	Tiefenbereich [m]	Outdoor
Kinect	632×480	30	0,8 - 3,5	nein
PMD CamCube 2.0[1]	204×204	25	8,33	ja
SwissRanger SR 4000[2]	176×144	54	5	ja

Tabelle 4.5: Vergleich zweier ToF-Kameras mit der Kinect Kamera, [1][Jutzi 2010], [2][Jutzi 2009],

4.5 Kamerakalibrierung

Die Elemente der inneren Orientierung der am jeweiligen Aufnahmesystem beteiligten Kameras werden über die Kamerakalibrierung ermittelt. Das sind bei der

Stereokamera die inneren Orientierungen der linken und rechten Kamera sowie die relative Orientierung der Kameras zueinander. Für die aus der Kalibrierung resultierenden Epipolarbilder stehen Projektionsmatrizen $\boldsymbol{P'}$ und $\boldsymbol{P''}$ für beide kalibrierten Kameras zur Verfügung, wobei nach der Zerlegung (siehe Abschn. 3.1.3) in $\boldsymbol{R}, \boldsymbol{t}$ und $\boldsymbol{K}$ gilt: $\boldsymbol{K'} = \boldsymbol{K''}$, $\boldsymbol{R'} = \boldsymbol{I}$ und $\boldsymbol{t'} = (0, 0, 0)^T$. Mit $\boldsymbol{R''}$ und $\boldsymbol{t''}$ ist die relative Orientierung beschrieben.

Die Kinect Kamera benötigt die inneren Orientierungen von RGB-Kamera und IR-Kamera sowie deren relative Orientierung. Zudem wird die Basislinie zwischen IR-Kamera und IR-Projektor benötigt. Die Kamera ist ab Werk kalibriert. Bei Verwendung der OpenNi-Schnittstelle zum Auslesen der Kamera benötigt man keine Kalibrierdaten.

Neben der Berechnung des Tiefenbildes werden die Kalibrierdaten auch für die folgenden Abbildungen benötigt:

- Abbildung des RGB-Bildes auf das Tiefenbild um eine RGB-Punktwolke zu erhalten.
- Abbildung des Tiefenbildes auf das RGB-Bild, um direkt Tiefenwerte im RGB-Bild abgreifen zu können. In Abb. 4.18 ist so eine Transformation dargestellt.

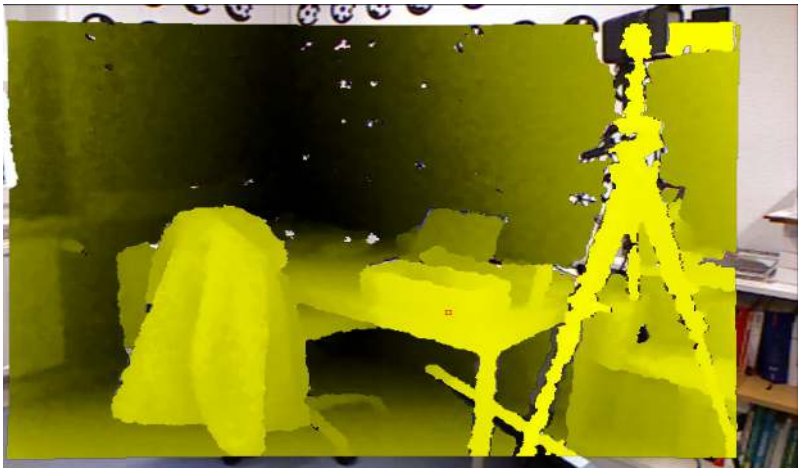
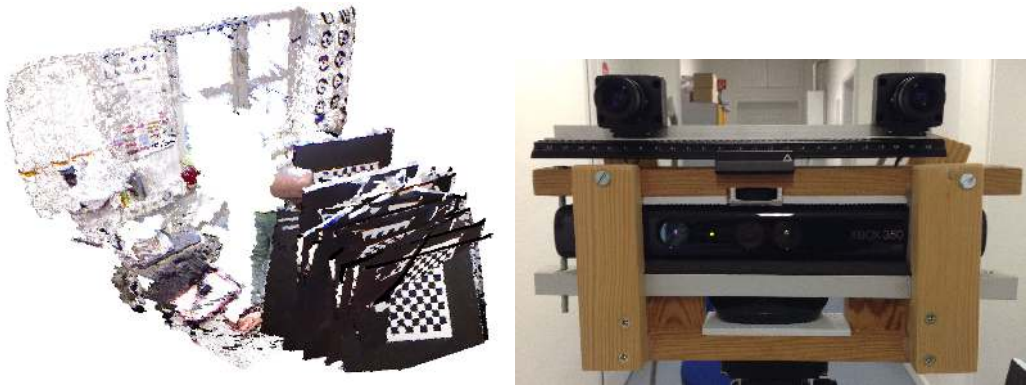


Abbildung 4.18: Kinect: Tiefenbild auf RGB-Bild registriert. Erzeugt mit OpenNi

Für die Kalibrierung beider Systeme wurde OpenCV genutzt. Als Kalibrierfeld kam das übliche planare Schachbrettmuster zum Einsatz (siehe Abb. 4.19a, S. 64).

Die beiden Aufnahmesysteme sind fest miteinander verbunden (siehe Abb. 4.19b). Das Stereokamerasystem besteht aus zwei monochromen Firewire 400 Kameras mit VGA-Auflösung (640×480). Die Kameras lösen zeitsynchron aus, wenn sie am gleichen FireWire Bus angeschlossen sind. Die Aufnahmen beider Systeme (linkes, rechtes Stereobild, Kinect-RGB und Kinect Tiefenbild) können mit bis zu 12 Hz zusam-



(a) Kinect: Lage der Kalibriertafeln **(b)** Aufnahmesystem bestehend aus Stereokamera und Kinect

Abbildung 4.19: Lage der Kalibriertafeln für das Aufnahmesystem

men synchron aufgenommen werden. Das synchrone Auslesen der Kamerabilder wurde mit Aufnahmen einer hochzählenden Vierfach-Siebensegmentanzeige kontrolliert. Manuelle Vergleiche der Zählwerte in den gespeicherten Aufnahmen des Stereokamerasystems und der Kinect RGB-Kamera dienten als Beweis. Das zeitsynchrone Aufnehmen der Kinect Bilder (RGB-Bild und Tiefenbild) wurde nicht getestet. Die Kinect bietet ohne Kenntnisse des inneren Aufbaus hierfür auch keine Einflussmöglichkeit.

Die Kameras sind so ausgerichtet, dass sich die Aufnahmen überlappen (siehe Abb. 4.20 und 4.21a).

(a) linkes Stereobild **(b)** rechtes Stereobild **(c)** Kinect RGB **(d)** Kinect Tiefenbild

Abbildung 4.20: Die vier Aufnahmen des Stereo-Kinect-Systems

Die Stereokameras sind als Normalfall ausgerichtet und haben eine Basis von 18,1 cm. Die Öffnungswinkel der Stereokamera-Objektive ist kleiner als die der Kinect. Die Lage der Epipolarbilder im Kinect RGB-Bild sind in Abb. 4.21a gezeigt.

4.6 Vergleich von Tiefenbildern aus Stereokameras und Kinect

In Abb. 4.21 sind Tiefenbilder einer Gangszene 4.21(a) zu sehen. Das Tiefenbild 4.21(b) stammt von der Kinect Kamera, die unteren beiden Tiefenbilder sind mit den Stereoanalyseverfahren Block Matching (c) und Semi-Global Block Matching (d) erzeugt worden.

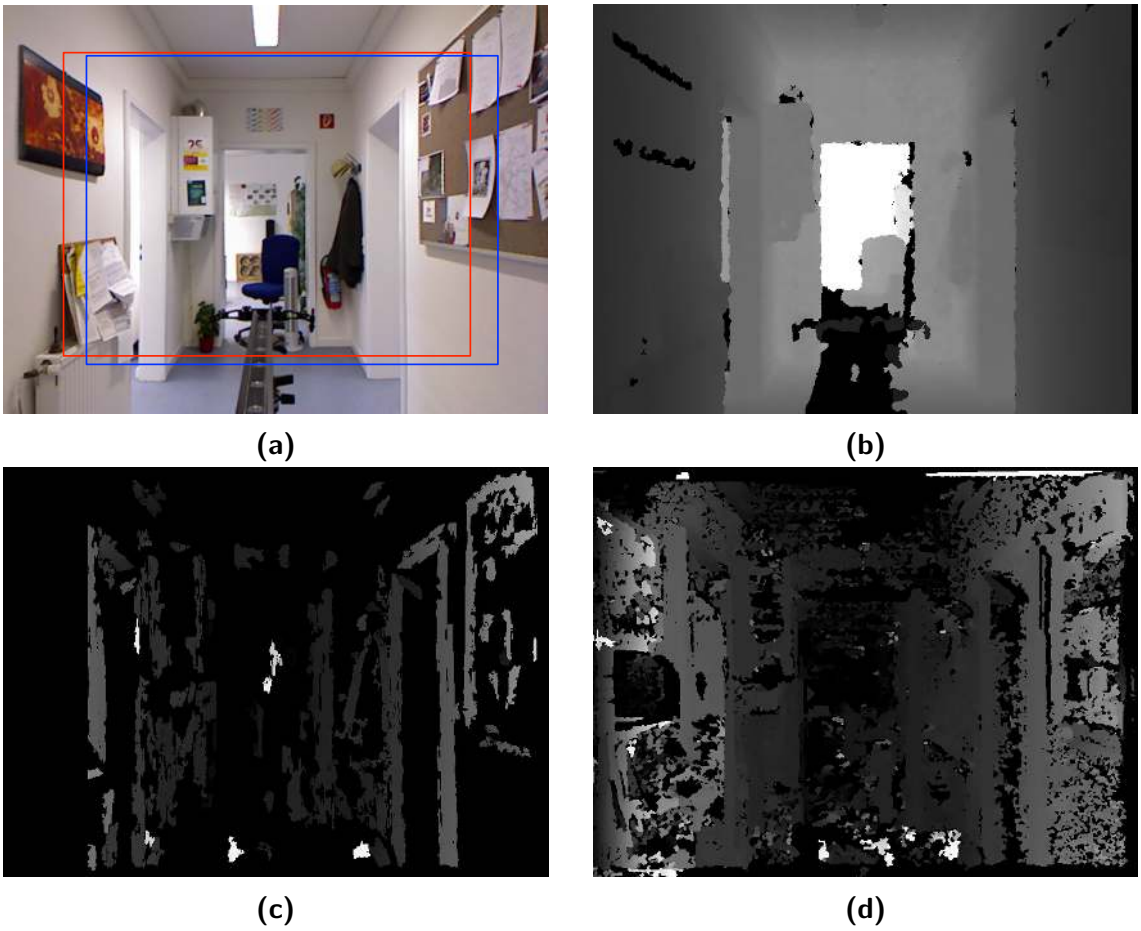


Abbildung 4.21: Gang Szene (a) mit Lage des linken und rechten Stereobildes: Ergebnisse zweier unterschiedlicher Stereoverfahren: Block Matching (c), Semi-Global Block Matching (d) im Vergleich zum Kinect Tiefenbild (b)

Die Szene ist wegen der weißen, texturschwachen Wandflächen schlecht für die Stereoanalyse geeignet. Die Kinect Kamera ist hier als aktives System nicht auf natürliche Merkmale angewiesen. Tabelle 4.6 zeigt den prozentualen Anteil gültiger Tiefenwerte pro Verfahren für diese Szene.

Für die Berechnung der Eigenbewegung wird die Menge natürlicher Merkmale, die mit dem Stereokamerasystem gemessen werden kann, ausreichen. Für eine zuverlässige dichte 3D-Punktwolke ist die Tiefenbildkamera besser geeignet.

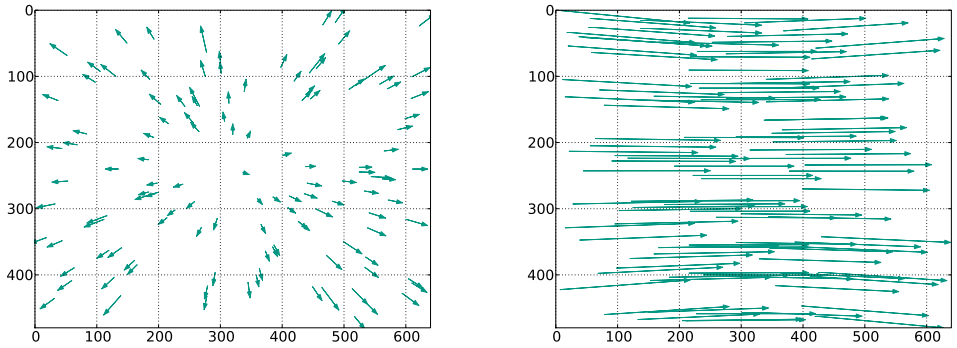
Verfahren	Abb.	% gültige Werte	Berechnungszeit [ms]
Kinect	4.21b	94	-
Block Matching	4.21c	19	27 ± 6
Semi-Global Block Matching	4.21d	66	355 ± 96

Tabelle 4.6: Laufzeiten der Tiefenbildberechnungen

Im folgenden Kapitel wird die Extraktion und Verfolgung von Bildmerkmalen behandelt.

5 Merkmalsextraktion und Merkmalsverfolgung

Abbildung 5.1 zeigt die Bewegungen von einzelnen Merkmalen zwischen zwei Bildern. Hierfür wurden zufällig verteilte, synthetische 3D-Punkte erzeugt und mit je zwei simulierten Projektionsmatrizen für die entsprechenden Bewegungen in die Bildebene projiziert. Die Pfeile repräsentieren die Bewegung der Merkmale vom ersten zum zweiten Bild. In der linken Abbildung wurde die synthetische Kamera in Richtung ihrer optischen Achse nach vorne bewegt. In der rechten Abbildung stand die Kamera fest und wurde nach links gedreht. Beide Kamerabewegungen verursachen deutliche Verschiebungen bei den Bildkoordinaten der abgebildeten Objektpunkte.



(a) Translation einer Kamera um $t_x = 1$ **(b)** Rotation einer Kamera um $r_y = 15°$

Abbildung 5.1: Durch Kameraeigenbewegungen verursachte Bewegung von Merkmalen im Bild (640×480 Pixel)

Die Bildkoordinaten der Merkmale fließen in die Berechnung der Kameraposition ein. Das bildbasierte Tracking gliedert sich daher in die drei Teilbereiche Merkmalsextraktion, Matching der extrahierten Merkmale zu zwei Aufnahmezeitpunkten und

Parameterschätzung der Bewegung, die in vielen Systemen verschränkt oder iterativ ablaufen und somit als Einheit betrachtet werden sollten.

Es gilt, die statischen Merkmale einer Szene zu erfassen, damit die relative Bewegung der Kamera in dieser Szene berechnet werden kann. Aus den Bewegungen der Bildmerkmale allein kann nicht ohne Weiteres auf deren eigentliche Bewegung im Raum geschlossen werden.

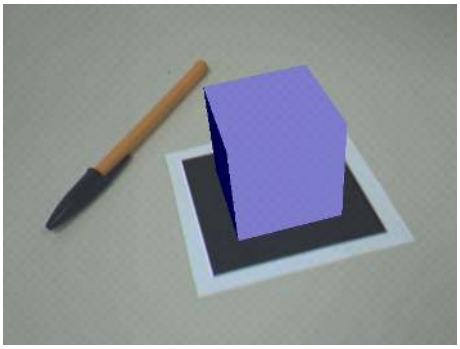
Die Verfahren werden für den weiteren Verlauf dieses Kapitels in vier Bereiche unterteilt:

- Tracking mit Zielmarken
- Tracking natürlicher Bildmerkmale mit Hilfe des optischen Flusses (z.B. Kanade-Lucas-Tracker), siehe Abschnitt 5.2.3
- Tracking natürlicher Bildmerkmale über die Zuordnung von Merkmalsvektoren (z.B. Keypoint Verfahren), siehe Abschnitt 5.2.4
- Modellbasiertes Tracking, siehe Abschnitt 5.3

Die Erscheinung bestimmter Objektklassen im Bild kann auch vorab gelernt werden. Das Tracking solcher Objekte erfolgt dann über die fortlaufende Erkennung mit dem trainierten Detektor. Diese *Tracking-by-Detection* Methoden, wie z. B. beim Personen- [Schmidt und Hinz 2011] oder Fahrzeugtracking [Hinz u. a. 2007], werden hier nicht behandelt.

5.1 Tracking mit Zielmarken

Schwerpunkt beim bildbasierten Tracking bleibt die Definition und Zuordnung geeigneter Merkmale im Bild. Besonders gekennzeichnete Zielmarken vereinfachen die Detektion erheblich und sind im AR-Bereich weit verbreitet. Es gibt kreisförmige, kodierte Zielmarken aus dem Bereich des Echtzeittrackings und der Photogrammetrie, deren Lage in Subpixelgenauigkeit ermittelt werden kann. Allerdings müssen entsprechend viele dieser Passpunkte in der Systemumgebung angebracht werden, damit eine minimale Anzahl von Punkten in einer sinnvollen geometrischen Verteilung im Bild erkennbar ist. [Kato und Billinghurst 1999; Szalavari u. a. 1998] nutzen quadratische, planare Zielmarken, deren innerer Bereich eine Kodierung zur eindeutigen Unterscheidung mehrerer Zielmarken hat. Allein durch die vier Eckpunkte kann die Kameraposition relativ zu einer Zielmarke ermittelt werden. Damit genügt schon eine Zielmarke zum Überlagern des Kamerabildes mit Zusatzinformationen. Die einfache Anwendbarkeit dieses Prinzips durch die Verfügbarkeit der ARToolKit-Bibliothek [Lamb 2010] (siehe Abb. 5.2, S. 69) haben zu einer weiten Verbreitung dieses Ansatzes geführt. Der Schwerpunkt der ARToolKit-basierten Anwendungen



(a) Einfaches Beispiel mit virtuellem Würfel

(b) Typische Zielmarke

Abbildung 5.2: Markerbasiertes Tracking mit ARToolKit [Kato und Billinghurst 1999] in Echtzeit

liegt in der Nutzerinteraktion und im Nahbereich, wie z. B. das Tracking der Handposition [Piekarski und Thomas 2002b] oder in der Leberchirurgie [Suthau 2003]. Es wurde aber auch für AR-basierte Gebäudeinformationssysteme genutzt [M. Wagner 2002]. Die Trackinggenauigkeit nimmt mit zunehmendem Abstand zur Zielmarke schnell ab [Piekarski und Thomas 2002a].

Der größte Nachteil kodierter Zielmarken ist die notwendige Vorbereitung der Systemumgebung durch das Anbringen einer ausreichend großen Anzahl an Marken und die Messung der zugehörigen 3D-Koordinaten. In manchen Umgebungen wird das Anbringen von Zielmarken nicht erwünscht sein und bei vielen Outdooranwendungen ist es nicht möglich.

5.2 Tracking mit natürlichen Bildmerkmalen

Das Tracking natürlicher Merkmale im Bild wie Eckpunkte, Kanten oder anderer, speziell für die Systemumgebung typischer Merkmale, ermöglicht den Verzicht auf Zielmarken. [Georgel u. a. 2007] extrahieren z. B. in der Wand befestigte Ankerplatten im Bild und nutzen diese als Passpunkte, da deren Objektkoordinaten aus der Vermessung bekannt sind. Die Zuordnung der einzelnen Platten ist durch deren eindeutige Lage und Verteilung an der Wand möglich.

Generell muss unterschieden werden, ob die korrespondierenden, natürlichen Merkmale zwischen den einzelnen Kamerabildern lediglich Informationen für die relative Bewegung von Bild zu Bild liefern sollen oder ob diese natürlichen Merkmale Passpunkte bzw. Landmarken mit bekannter Lage im Raum darstellen. [Weinmann 2013] bietet eine Übersicht möglicher Merkmale im Bild. Diese können allgemein nach Intensität im Bild, Form, Textur oder lokale Merkmale unterteilt werden.

Markante Ecken im Bild eignen sich gut für punktförmige Merkmale. Dies sind Punkte im Bild, deren Signal I sich in zwei Richtungen ändert. Änderungen im Bildsignal werden durch dessen Ableitungen modelliert. Werden die Ableitungen in zwei zueinander orthogonalen Richtungen beobachtet, liegt wahrscheinlich ein Eckpunkt vor.

Harris hat einen bekannten Ecken-Detektor [Harris und Stephens 1988] auf Basis der partiellen Ableitungen $I_x(x, y)$, $I_y(x, y)$ des Bildes I geschaffen, der später von Shi und Tomasi [Shi und Tomasi 1994] durch die direkte Verwendung der Eigenwerte anstelle der Harris *Corner Response Funktion* vereinfacht wurde.

Die Strukturmatrix S (auch Autokorrelationsmatrix oder Strukturtensor) beschreibt die Verteilungen der Gradienten in einer lokalen Nachbarschaft N im Bild:

$$S = \sum_{(x,y)\in N} w(x, y) \begin{bmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{bmatrix}, \tag{5.1}$$

wobei $w(x, y)$ eine Fensterfunktion ist.

Die Analyse der beiden Eigenwerte λ_1 und λ_2 der Strukturmatrix S liefert die Art des Merkmals. Bei zwei kleinen Eigenwerten ist die lokale Nachbarschaft strukturarm. Bei einem großen Eigenwert und einem kleinen Eigenwert liegt eine Kante vor. Ein Eckmerkmal ist durch zwei großen Eigenwerte definiert. Als Schwellwert c für Eckmerkmale nach [Shi und Tomasi 1994] gilt:

$$c = \min(\lambda_1, \lambda_2). \tag{5.2}$$

5.2.1 Merkmalsverfolgung mit dem optischen Fluss

Der optische Fluss hat die Erfassung des Bewegungsfeldes zum Ziel. Dazu bedarf es einer hinreichend räumlich und zeitlich aufgelösten Bildfolge. Dabei kann nur das Resultat der Abbildungen von Objektbewegungen in die Bildebene beobachtet werden. Es kann nicht unterschieden werden, ob sich der Sensor bewegt, ob sich nur die Objekte bewegen, ob es zu einer Überlagerung von beiden Bewegungsarten kommt oder ob sich nur die Beleuchtung ändert.

Durch das Beobachten von lokalen Nachbarschaften kann Richtung oder Betrag einer über die Nachbarschaft hinaus gehenden Bewegung womöglich nicht ermittelt werden. Dieser Effekt ist auch als Blendenproblem bekannt.

Zudem ist das Problem des optischen Flusses schlecht konditioniert: zu jedem Pixel werden zwei Komponenten für die Bewegung (u, v) gesucht. Es müssen daher mehr Bedingungen aufgestellt werden. Mit Hilfe unterschiedlicher Regularisierungsansätze, die entweder den zeitlichen Fluss von Bild zu Bild oder räumliche Eigenschäften des Bewegungsfeldes benachbarter Merkmale modellieren, kann das „ill-posed problem“ des optischen Flusses gelöst werden.

Horn und Schunck [Horn und Schunck 1981] haben dafür zwei Bedingungen für den optischen Fluss aufgestellt:

- Flussbedingung: konstante Helligkeit zwischen den Bildern

$$I(x, y, t) \equiv I(x + u, y + v, t + 1) \tag{5.3}$$

  Dies führt zur Flussgleichung

$$I_x u + I_y v + I_t = 0 \tag{5.4}$$

  wobei I_x, I_y und I_t die partiellen Ableitungen nach x, y, t sind.

- Glattheitsbedingung: nur kleine Bewegungen von Bild zu Bild .

Diese Bedingungen haben sie zu einem Energiefunktional E_{HS} zusammengefasst, das für das Bild (Bildraum Ω) minimiert werden muss. Allgemein wird das Energiefunktional bestehend aus Ähnlichkeitsterm (d für *data term*) E_{d} und Glattheitsterm E_{s} (s für *smoothness*)

$$E(u, v) = \int_{\Omega} (E_{\mathrm{d}} + \alpha E_{\mathrm{s}}) \mathrm{d}x \ \mathrm{d}y \tag{5.5}$$

bei Horn-Schunck zu:

$$E_{\mathrm{HS}} = \int_{\Omega} (I_x u + I_y v + I_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) \mathrm{d}x \ \mathrm{d}y, \tag{5.6}$$

wobei α der Regularisierungsparameter ist.

Im Laufe der Zeit wurden unzählige Varianten von Minimierungsstrategien, Ähnlichkeitstermen und Regularisierern für den optischen Fluss entwickelt [Baker u. a. 2011].

5.2.2 Szenenfluss

Neben dem Bildfluss kann auch der Disparitätsfluss bei Stereobildsequenzen regularisiert werden [Wedel und Cremers 2011] (siehe auch Kap. 6). Mit der Disparität wird die Raumtiefe Bestandteil des Flusses zwischen zwei aufeinander folgenden Epipolarbildpaaren. Laut [Wedel und Cremers 2011] wurde der Szenenfluss zum ersten Mal 1996 in [Patras u. a. 1996] vorgestellt.

Wegen der gleichzeitigen Berechnung von 3D-Geometrie und 3D-Bewegung wird dies als Szenenfluss bezeichnet. Die Entwicklung von echtzeitfähigen Systemen wurde wegen der hohen Dynamik in Verkehrsszenen von der Automobilbranche vorangetrieben [Franke, Gehrig u. a. 2008].

Der dichte optische Bildfluss (u, v) wird mit dem dichten Fluss der Disparität p zu einem Flussfeld $[u, v, p]^T$ erweitert. Die Intensität wird im rechten und linken Stereobildpaar I^L, I^R für beide aufeinander folgende Zeitpunkte als konstant angenommen. Die dritte Flussbedingung wird dann zu [Wedel und Cremers 2011]:

$$I(x + u, y + v, t)^L = I(x + d + p + u, y + v, t)^R \tag{5.7}$$

Die x-Bildkoordinate des rechten Bildes ergibt sich aus Disparität d, Änderung der Disparität p und dem Bildfluss u in x-Richtung.

Das zu minimierende, erweiterte Energiefunktional $E(u, v, p)$ wird zu [Wedel, Brox u. a. 2011]:

$$E(u, v, p) = \int_\Omega (E_\mathrm{D}(u, v, p) + E_\mathrm{S}(u, v, p))\mathrm{d}x\ \mathrm{d}y \tag{5.8}$$

mit der Glattheitsbedingung E_S

$$E_\mathrm{S} = \lambda\Psi(|\nabla u|^2 + |\nabla v|^2) + \gamma\Psi(|\nabla p|^2). \tag{5.9}$$

wobei λ und γ die unterschiedlichen Regularisierungsparameter für Bildfluss und Dispritätsfluss sind. Als Regularisierungsfunktion Ψ wird in [Wedel, Brox u. a. 2011] eine Näherung der L^2-Norm durch einem L^1-Norm Total-Variation-Ansatz verwendet. Diese ist robust gegenüber Ausreißern. In [Wedel und Cremers 2011] werden unterschiedliche Regularisierungsfunktionen evaluiert.

Pixel, die wegen Verdeckungen keine Disparitätswerte haben, werden für den Szenenfluss ignoriert.

5.2.3 Der Kanade-Lucas Feature Tracker

Der Kanade-Lucas Feature Tracker [Lucas und T. Kanade 1981] (kurz: KLT, oft auch LKT) hat die Erfassung der Bewegung markanter Punkte zum Ziel. Das Ergebnis ist ein „dünn besetzter“ optischer Fluss (*sparse optical flow*) bei dem die Auswahl der Merkmale nicht vom Tracking getrennt ist.

Zusätzlich zu den Bedingungen des optischen Flusses kommt noch die Forderung nach einer räumlichen Kohärenz der Punkte. Diese wird durch die Nachbarschaft N eines Pixels definiert. Es gilt:

$$E_\mathrm{LK} = \sum_N (I_x u + I_y v + I_t)^2 \longrightarrow \min. \tag{5.10}$$

Bei der Wahl der Nachbarschaftsgröße muss zwischen Genauigkeit und Robustheit abgewägt werden. Ein zu großes Fenster glättet die darin enthaltenen Details und sorgt für die geforderte räumliche Kohärenz. Gleichzeitig reduziert es aber auch die Genauigkeit. Zudem könnten benachbarte Bildbereiche, die in unterschiedliche Richtungen fließen, mit zu großen Fenstern nicht mehr unterschiedenen werden.

Durch eine Taylor-Reihenentwicklung ergibt sich ein lineares Gleichungssystem, das bei einer typischen Nachbarschaft von $N = 5 \times 5$ zu 25 Gleichungen führt. Für den Flussvektor gilt:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b \tag{5.11}$$

mit

$$A^T A = \sum_N \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad \text{und} \quad b = \sum_N \begin{bmatrix} \partial I I_x \\ \partial I I_y \end{bmatrix}. \tag{5.12}$$

Es ist dann lösbar, wenn das Normalgleichungssystem $A^T A$ invertierbar ist d.h. einen Rang 2 hat und zwei große Eigenvektoren hat. $A^T A$ entspricht der Strukturmatrix. Die Flussvektoren zeigen immer zum markantesten Eckmerkmal in N. Der resultierende Flussvektor (u, v) hat nach einer iterativen Anpassung Subpixel-Genauigkeit.

Die Fenstergröße für die lokale Nachbarschaft schränkt die maximal mögliche Bildbewegung ein. Eine Vergrößerung des Fensters würde zu weniger räumlicher Kohärenz führen. Ansätze mit Bildpyramiden vergrößern dabei den maximal verfolgbaren Pixelabstand der Merkmalspunkte von Bild zu Bild [Thormählen 2006]. Drei Pyramidenstufen erlauben z. B. eine 15-fache Vergrößerung des Suchbereiches. Dies würde bei einem 5×5 Fenster ein Suchbereich von 75 Pixel bedeuten. Mehr Pyramidenstufen ergeben bei kleinen Bildern wenig Sinn. Für ein Bild in VGA-Größe (640×480) ergibt dies Stufen von 320×240, 160×120, 80×60 und für die vierte Stufe 40×30.

Bei diesen Werten für Fenstergröße und Anzahl von Pyramidenstufen ergäbe sich ein Randbereich von 40 Pixel im Originalbild. Aus diesem Grund muss in der Implementierung darauf geachtet werden, nur zulässige Pixel innerhalb der Nachbarschaft in die Berechnung mit einzubeziehen [Bouguet 2000].

Die Merkmale sind verloren, wenn sie sich außerhalb dieses Suchabstandes befinden, und es müssen neue Merkmale aus dem Bild extrahiert werden. Beim KLT kann es auch leicht passieren, dass einzelne Punkte entlang einer stärkeren Ecke, aber eigentlich falschen Ecke, abdriften.

Abhilfe kann hier ein Vorwärts- und Rückwärtstracking mit einer anschließenden Konsistenzkontrolle bieten [Kalal u. a. 2010].

5.2.4 Merkmalsverfolgung mit Keypointverfahren

Mit dem SIFT (Scale Invariant Feature Transform) [Lowe 2004] oder SURF (Speed Up Robust Features) [Bay u. a. 2006] Verfahren werden markante Merkmale im Bildraum extrahiert und zusätzlich ein Merkmalsvektor (Deskriptor) zu jedem Punkt ermittelt, der diesen mit Hilfe seiner lokalen Umgebung möglichst eindeutig beschreibt. Allgemein wird ein Bildmerkmal mit einem zusätzlichen Merkmalsvektor Keypoint genannt.

Keypoints können über den Vergleich ihrer Merkmalsvektoren zugeordnet werden. Damit entfällt die Beschränkung auf einen bestimmten Suchbereich und die Bewegung zwischen zwei Bildern kann entsprechend größer sein. Die Bezeichnung *wide-baseline matching* soll verdeutlichen, dass größere Änderungen zwischen den Ansichten eines Merkmals erlaubt sind [Tuytelaars und Mikolajczyk 2008] und die Merkmale nicht innerhalb eines Suchfensters liegen müssen.

Unterschiedliche deskriptorbasierte Verfahren sind dabei gegenüber Änderungen der Betrachtungsperspektive, Rotation, Skalierung, Unschärfe oder Beleuchtung recht robust [Mikolajczyk und Schmid 2005]. Die Anwendung der Keypointverfahren ist generisch und kann in drei Schritte unterteilt werden:

- Merkmalsextraktion mit dem Detektor
- Erzeugung eines Keypointdeskriptors
- Matching des Keypointdescriptors

Beim SIFT Verfahren wird der Deskriptor aus einem Richtungshistogramm aus normierten, lokalen Gradientenrichtungen erzeugt.

Beim Berechnen der Relativbewegungen zwischen den Bildern erzeugen die mitgeführten Fehler eine Drift, die z. B. über bekannte Stützpunkte korrigiert werden muss. [Vacchetti u. a. 2004b] nutzen dazu Bilder, die zuvor in einer Trainingsphase von der Systemumgebung aufgenommen werden müssen. Die Orientierung und Lage dieser Referenzbilder (Keyframes) muss über 3D-2D Korrespondenzen ermittelt werden. Eine Gruppe von Keypoints einer bestimmten Textur am Objekt kann wiedererkannt und entsprechend zugeordnet werden und dient somit als Stützpunkt.

Diese Methode ähnelt der Verwendung kodierter Zielmarken und ist daher auch im AR-Bereich verbreitet. Die Texturmerkmale müssen dabei nicht planar sein. Es sind auch Verformungen möglich [Lepetit, Lagger u. a. 2005]. Obwohl die natürlichen Texturen vom Nutzer nicht als Marken erkannt werden, müssen sie dennoch in der Systemumgebung vorhanden sein und deren Postionen zuvor vermessen werden.

Auf mobilen Systemen wie Smartphones ist es möglich, Keyframe-basierte Ansätze zu nutzen [Klein und Murray 2009; D. Wagner u. a. 2008; Arth u. a. 2009]. Mit Keyframe-Datenbanken soll auch eine größere Systemumgebung (outdoor) möglich sein [Arth u. a. 2009; Schindler u. a. 2007; Irschara u. a. 2009]. Die zunehmende Verfügbarkeit von öffentlich zugänglichen, georeferenzierten Bildern im Internet rechtfertigen diese Ansätze [Zheng u. a. 2009]. An weniger bekannten Orten müssten zuerst ausreichend viele Keyframes erzeugt werden und diese bei Änderungen stets aktuell gehalten werden.

5.2.5 Vergleich der Strategien

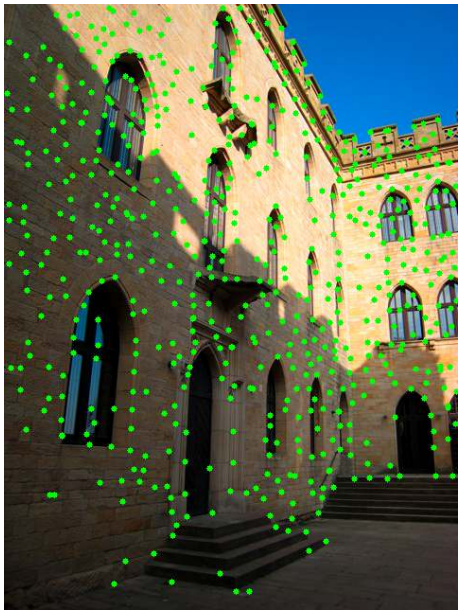
Grundsätzlich sind KLT und Keypointverfahren fähig Merkmale in Echtzeit zu verfolgen. Für beide Verfahrensarten gibt es GPU-Versionen, die dies ermöglichen. In [Sinha u. a. 2006] können mit einer GPU-KLT-Version 1000 Merkmale in einem 1024×768 Pixel großen Bild bei 30 FPS verfolgt werden. Eine SIFT-GPU-Version erreicht für 1000 Merkmale in 640×480 Pixel großen Bildern maximal 10 FPS.

Die Zahlen der GPU-basierten Verfahren sind beim Blick auf mobile Anwendungen immer mit Vorsicht zu genießen. Bei der verwendeten Hardware handelt es

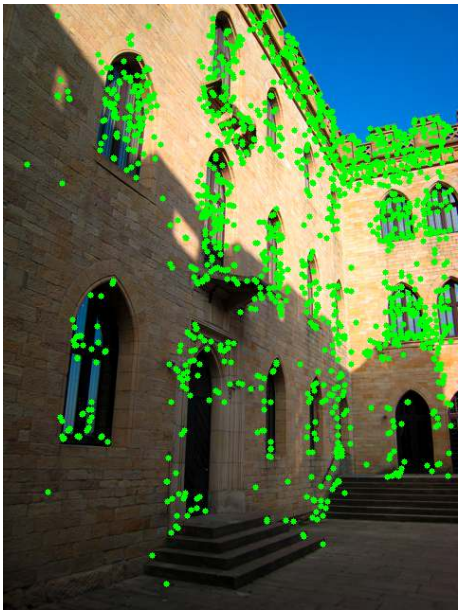
sich oft um dedizierte Grafikkarten, die ohnehin nur in wenigen Notebooks verbaut werden. In einem mobilen Szenario für Fußgänger spielen diese Geräteklassen eine untergeordnete Rolle.

Eigene Zeitmessungen haben ergeben, dass der SIFT Detektor etwa doppelt so viel Zeit pro Merkmalsextraktion benötigt, wie die Detektion eines Harris Merkmals mit anschließender Subpixelverfeinerung (diese hatte einen Zeitanteil von 40 %).

In Abb. 5.3 ist das Ergebnis der Merkmalsextraktion mit Harris- und SURF-Detektor gegenübergestellt. SURF ist ein Skaleninvarianter Merkmalsdetektor basierend auf der Hessematrix. Man sieht, dass an der Fassade des Gebäudes die meisten Merkmale extrahiert worden sind. SURF hat annähernd doppelt so viele Punkte extrahiert. Deren Lage lässt sich aber nicht so leicht nachvollziehen wie beim Harris Ergebnis.



(a) Harris Operator (703 Merkmale)



(b) SURF Operator (1380 Merkmale)

Abbildung 5.3: Unterschiedliche Anzahl und Verteilung der gefundenen Merkmale bei Harris und SURF Operator

Grundsätzlich sind Keypointverfahren für Bildsammlungen besser geeignet. Bei einer losen Sammlung von Bildern ohne bekannte zeitliche Abfolge oder zumindest ohne eine konstante zeitliche Abfolge ist es schwieriger oder gar unmöglich den Suchbereich eines homologen Merkmals von einem zum nächsten Bild einzuschränken. In einer Bildsammlung ist die Frage nach dem nächsten Bild ohne zusätzliche Informationen nicht so leicht zu klären. Hier ist es vorteilhafter einzelne Deskriptoren oder sogar lokale Gruppen von Deskriptoren zwischen einzelnen Bildern zu matchen.

Der optische Fluss passt dagegen besser zum Tracking in typischen Bildsequenzen mit Wiederholraten von z. B. 30 Hz. Hier kann man kleine Bewegungen zwischen zwei Bildern erwarten. Dies wird in [Klippenstein und Zhang 2007] bestätigt, wo KLT u.a. mit SIFT verglichen wurde. Bedingungen wie die räumliche Kohärenz benachbarter Merkmale sind bereits ein Teil des Bewegungsmodells. Die Kenntnis des konstanten Zeitabstandes zwischen den Bildern ermöglicht auch eine Vorhersage über den Suchbereich von Merkmalen, wenn das aktuelle Bewegungsmodell durch fortlaufende Schätzung zur Verfügung steht.

5.3 Modellbasiertes Tracking starrer Körper

Da bei AR ein 3D-Modell mit der Umgebung überlagert wird, liegt es nahe, dieses vorhandene 3D-Modell auch für das Tracking zu nutzen und so die Bewegungen des Kamerasystems innerhalb einer absoluten Referenz zu erhalten. Hätte man zu jedem Zeitpunkt ausreichend Passpunkte, wäre die Drift beim Tracking kein Problem mehr.

Mit [Lepetit und Fua 2005] gibt es eine ausführliche Übersicht zum modellbasierten Tracking. Diese wurde als Ausgangspunkt für die eigenen Ausführungen genommen.

Kantenbasierte Verfahren vergleichen die in den Bildraum projizierten Kanten eines 3D-Modells mit den Bildregionen die große Gradienten in der Bildintensität aufweisen [David u. a. 2003; David u. a. 2004; Wuest, Vial u. a. 2005]. In [Drummond und Cipolla 2002] werden 3D-Punkte in die Bildebene projiziert und mit den Bildkanten abgeglichen. Aus den neuen 2D-Positionen der korrespondierenden Punkte wird die Bewegung der Kamera relativ zum 3D-Modell ermittelt. Das Verfahren ist echtzeitfähig, benötigt aber eine Initialisierung. [Behringer u. a. 2002] und [Reitmayr und Drummond 2006] nutzen zusätzliche Sensoren wie GPS und IMU zur Initialisierung des modellbasierten Trackings mit 3D-Gebäudemodellen.

In [Vacchetti u. a. 2004a] wird ein echtzeitfähiges 3D-Trackingverfahren vorgestellt, das Kanten und Merkmalspunkte kombiniert. Für jedes neue Bild werden Merkmalspunkte mit Referenzbildern abgeglichen, die zuvor erstellt werden müssen. Da die Merkmalspunkte der Referenzbilder bekannte 3D-Koordinaten haben, stehen 3D-2D Korrespondenzen zur Berechnung der Kameraposition zur Verfügung. Um zu starke Schwankungen in der Kameratrajektorie zu vermeiden und das Verfahren bei schwach texturierten Regionen zu stützen, wird diese erste Kameraposition mit den Kantenmessungen und Merkmalspunkten des vorhergehenden Bildes kombiniert. Anschließend wird auf die aktuelle Kameraposition, die vorhergehende Kameraposition und auf die 3D-Koordinaten aus den korrespondierenden Merkmalspunkten ein Optimierungsverfahren angewandt.

[Rosten und Drummond 2005] kombinieren auch Kanten und Punktmerkmale. Hier werden jedoch keine Referenzbilder genutzt, sondern das 3D-Modell muss offline zur Verfügung stehen. Eckpunkte werden mit einem eigens entwickelten Algorith-

mus extrahiert und jeweils nur die aktuellen Punkte mit den vorherigen abgeglichen. Die daraus abgeleitete Schätzung der Kamerabewegung zwischen den Bildern wird für das kantenbasierte Tracking verwendet. Das Verfahren soll relativ große Bewegungen zwischen den Bildern bewältigen können: bis zu 15° Rotation und 200 Pixel Translation.

[Reitmayr und Drummond 2006] nutzen neben GPS und IMU ein texturiertes, grobes 3D-Modell zum modellbasierten Tracking. Zur aktuellen Positionsschätzung aus GPS und IMU wird das Bild des 3D-Modells gerendert und daraus Kantenstücke extrahiert. Die 3D-Koordinaten der Kantenstücke werden durch eine Projektion zurück auf das 3D-Modell ermittelt. Durch die Nutzung der Texturen werden in entfernteren 3D-Objekten automatisch weniger Kantenstücke aufgrund ihrer Skalierung gefunden. Das Rendering mit Verdeckungsberechnung und Texturskalierung wird direkt von der Grafikkarte übernommen und belastet die CPU nicht. Der hybride Ansatz erlaubt eine kurzzeitige, vollständige Verdeckung des Kamerabildes durch Fußgänger oder Fahrzeuge im Vordergrund.

[Wuest, Wientapper u. a. 2007] erzeugen das 3D-Kantenmodell zum Tracken direkt aus dem zur aktuellen Positionsschätzung der Kamera gerenderten Bild eines Oberflächenmodells. Dieses untexturierte Oberflächenmodell ist in einem VRML-Modell gespeichert. Ähnlich wie in [Reitmayr und Drummond 2006] führt dies automatisch zu einer verringerten Komplexität der Kanten in entfernteren Bereichen. Auch werden auf diese Weise keine verdeckten Kanten eingeführt. Mit Hilfe des Z-Buffers des 3D-Modells werden die Kanten der Silhouette erfasst. Damit ist es auch möglich, die Kanten runder Objekte wie Rohrleitungen in Industrieanlagen für das Tracking zu berücksichtigen.

Eine Möglichkeit, die Anzahl sichtbarer Merkmale in den Kamerabildern zu erhöhen, um somit die Wahrscheinlichkeit einer eindeutigen Merkmalskonfiguration zur Selbstlokalisierung zu erhalten, ist die Vergrößerung des Sichtfeldes mit Omnivisionkameras [J. W. Lee u. a. 2002; Goedemé u. a. 2007] oder mit Kameras, die unterschiedliche Blickrichtungen abdecken [Zhu u. a. 2008]. Stereokamerasysteme ermöglichen es, die räumliche Struktur zu erfassen und Verdeckungen in den Bildern beim modellbasierten Tracking zu berücksichtigen [Najafi und Klinker 2003].

Abgesehen von [Reitmayr und Drummond 2006], die eine initiale Postions- und Lageschätzung von GPS und IMU implementiert haben und [Ottlik und Nagel 2007], die speziell die das Initialisierungsproblem untersuchen, wird bei den anderen modellbasierten Ansätzen nicht weiter auf die Automation der Initialisierung eingegangen. Es ist davon auszugehen, dass eine initiale Kameraposition- und Lage in einer Laborumgebung manuell erzeugt wird, wie im Beispiel [Najafi und Klinker 2003]: hier muss der Anwender zuerst mehrere Bildmerkmale manuell auswählen. Da [Behringer u. a. 2002] das modellbasierte Trackingverfahren als Ergänzung ihres mobilen AR-Systems [Livingston u. a. 2002] mit GPS und IMU vorstellen, ist anzunehmen, dass die Initialisierung nicht mit dem Kameratracking durchgeführt wird.

[Wuest, Wientapper u. a. 2007] verweist im Ausblick auf eine mögliche zukünftige Verwendung einer IMU zur besseren Positions- und Lageschätzung der Kamera.

Die für das Tracking verwendeten Modelle sind meist so klein, dass sie vollständig im Kamerabild sichtbar sind und nur auf die Robustheit bezüglich Verdeckungen von Teilbereichen eingegangen wird. In [Ulrich u. a. 2009] werden Werkstücke erkannt und deren Lage- und Orientierung im Raum ermittelt. [Dahlkamp u. a. 2007; Ottlik und Nagel 2007] beschäftigen sich mit dem modellbasiertem Tracking von Fahrzeugen auf einer Straßenkreuzung. [Wuest, Wientapper u. a. 2007; Drummond und Cipolla 2002; Rosten und Drummond 2005] untersuchen das Tracking von Geräten und Strukturen innerhalb eines Raumes, wobei nur [Wuest, Wientapper u. a. 2007] ein Verfahren vorstellen, dass mit kleinen Detailansichten funktionieren kann.

In [Urban u. a. 2013] wird eine Lösung des Initialisierungsproblemes vorgestellt. Dort wird ein Helmsystem mit drei montierten Fisheyekameras gezeigt, das die initiale Position innerhalb eines Gebäudebereiches mit Hilfe eines Abgleiches der Kanten in Fisheyebildern und Modell schätzen kann.

Tabelle 5.1 stellt die untersuchten Verfahren gegenüber. In der letzten Spalte wird bewertet, ob die Verfahren für AR geeignet sind.

Arbeit	Zusätzliche Sensoren	Initialisierung	Daten /Modell	max. Verdeckung	AR?
[Rosten und Drummond 2005]	-	-	Punkte auf Kanten	teilweise, kein Wert	✓
[Wuest, Vial u. a. 2005; Wuest, Wientapper u. a. 2007]	-	Zukunft: IMU	Polygon, daraus Oberflächen, sichtbare Kanten	teilweise, kein Wert	✓
[Reitmayr und Drummond 2006]	GPS, IMU	GPS, IMU	Polygone, Texturen	ganz, kein Wert	✓
[Behringer u. a. 2002]	GPS, IMU	GPS, IMU	Punkte auf Kanten	keine Angabe	✓
[Najafi und Klinker 2003]	-	manuell	-	teilweise, kein Wert	✓
[Drummond und Cipolla 2002]	-	-	-	teilweise, kein Wert	-
[David u. a. 2003; David u. a. 2004]	-	Robustheit erhöht	Kanten	$< 50\,\%$	-
[Ulrich u. a. 2009]	-	offline Training	Polygone, gerenderte 2D Ansicht	„robust“, kein Wert	-
[Ottlik und Nagel 2007]	-		Kanten		-
[Marchand und Chaumette 2002]	-	-	Kanten	teilweise, kein Wert	✓
[Urban u. a. 2013]	-	offline Training	Kanten	„robust“	✓

Tabelle 5.1: Gegenüberstellung untersuchter Verfahren zum modellbasierten Tracking. In der letzten Spalte „AR“ wird bewertet, ob das jeweilige Verfahren für AR geeignet ist

6 Eigenbewegungen aus Stereobildsequenzen

Nachdem in Kapitel 3 auf mögliche geometrische Verfahren zur Bewegungsberechnung eingegangen wurde und in Kapitel 5 die Detektion und Verfolgung von Merkmalen im Bild behandelt wurde, werden in diesem Kapitel komplette Abläufe zur Eigenbewegungsanalyse mit Hilfe von Stereobildsequenzen und Kinect RGB-D-Sequenzen untersucht.

Die Verfahren haben folgende Eingangsdaten für die Bewegungsschätzung gemeinsam (siehe Abb. 6.1, S. 82):

- Stereoverfahren: mindestens zwei Epipolarbildpaare einer Stereosequenz
- RGB-D Verfahren: zu mindestens zwei Aufnahmezeitpunkten jeweils ein Farbbild (RGB) und ein Tiefenbild (D)

Während die Stereoverfahren 3D-Merkmale selbst aus den korrespondierenden Bildmerkmalen im linken und rechten Stereobild erzeugen, werden diese bei RGB-D Kameras aus dem Tiefenbild abgeleitet, das auf der Kamerahardware erzeugt wird.

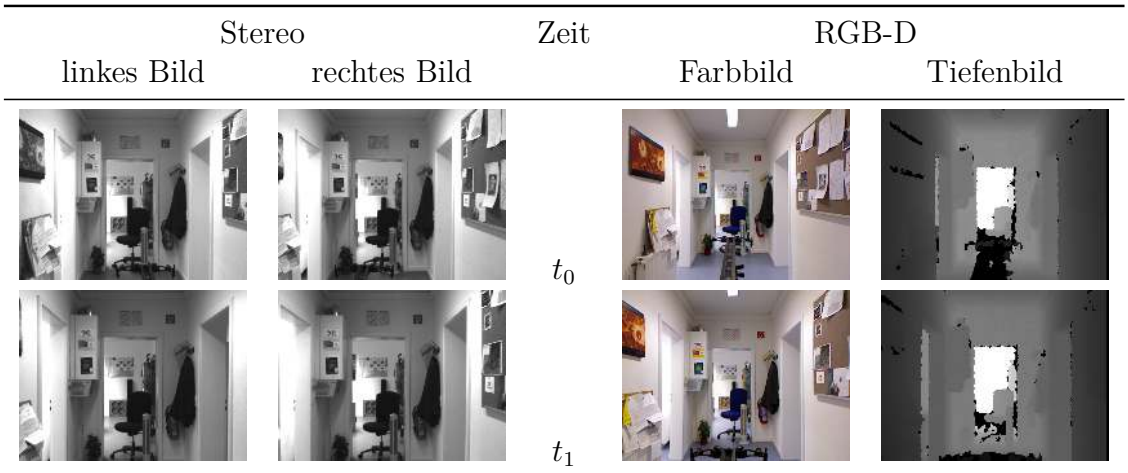


Abbildung 6.1: Eingangsdaten für die Bewegungsschätzung bei Stereo- und RGB-D-Kamera

6.1 Plattformen und Anwendungsbereiche

Typische Anwendungsgebiete von Stereo-Egomotion-Systemen sind im Bereich der autonomen Fahrzeuge und Fahrerassistenzsysteme [Geiger, Lenz u. a. 2012; Franke, Gehrig u. a. 2008], in der Robotik [Nistér u. a. 2004], in UAV [A. Huang u. a. 2011] oder in der Fußgängernavigation [Hernán Badino und Takeo Kanade 2011; Pradeep u. a. 2010; Molton 1998] zu finden. Die vier „Hauptplattformen“ sind in Abb. 6.2 dargestellt. In [Scaramuzza und Fraundorfer 2012] werden auch Unterwasserroboter als Anwendung erwähnt, diese werden hier aber nicht weiter behandelt.

Die Geschichte der Stereo Odometrie beginnt bereits 1980 durch Arbeiten von Moravec zur Hindernisdetektion und Navigation unbemannter Rover zur Erkundung des Mars [Scaramuzza und Fraundorfer 2011]. Solche Erkundungsfahrzeuge müssen nicht zwingend in Echtzeit navigieren können. Man kann davon ausgehen, dass keine bewegten Objekte die Eigenbewegungsanalyse stören. Ein großes Problem ist die robuste Merkmalsverfolgung in der wüstenähnlichen Umgebung [Maimone u. a. 2007]. Es muss mehr Aufwand betrieben werden, um die Drift möglichst klein zu halten, da außer der Lokalisierung durch das Matching der Rover Panoramen mit Ansichten aus Bilddaten der MARS Fernerkundungssatelliten (mit HiRISE Bilddaten[1] und DHM) keine weitere Möglichkeit einer Positionskorrektur, wie z. B. mit GNSS, besteht [Parker u. a. 2013].

Fahrerassistenzsysteme und autonome Fahrzeuge, die am öffentlichen Straßenverkehr teilnehmen, müssen in Echtzeit arbeiten. Zudem müssen neben der kamerabasierten Selbstlokalisation und Hinderniserkennung oft noch weitere Aufgaben

[1] High Resolution Imaging Science Experiment

Stereo Egomotion Plattformen			
Autonome Fahrzeuge	Robotik	UAV	Fußgänger
			

Abbildung 6.2: Typische Plattformen für Stereo Egomotion Systeme. Quellen von links nach rechts: [Geiger, Lenz u. a. 2012], [Nistér u. a. 2006], [A. Huang u. a. 2011], [Hernán Badino und Takeo Kanade 2011]

gelöst werden. Dazu gehören z. B. die Klassifizierung der anderen Verkehrsteilnehmer (Fahrzeug, Fahrradfahrer, Fußgänger), Geschwindigkeit und Größe der Hindernisse, die Art ihrer Bewegung und die Erkennung der befahrbaren Spur. Die Leistungsfähigkeit eines solchen Systems hat Daimler 2013 mit der vollständig autonomen Fahrt der ca. 100 km langen historischen „Bertha Benz Memorial Route“ von Mannheim nach Pforzheim über Landstraßen und durch Ortschaften demonstriert [Franke, Pfeiffer u. a. 2013].

Unbemannte Fluggeräte (UAV, *unmanned* oder *unpiloted aerial vehicle*) können in zwei Unterklassen unterteilt werden: Fluggeräte, bei denen Parameter wie Flughöhe und Aktionsradius denen eines bemannten Fluggerätes gleichen und Fluggeräte, mit denen man eher kleinräumige Umgebungen untersucht. Kleine Multicopter, mit geringer Flughöhe, die in urbanen Umgebungen oder innerhalb geschlossener Gebäude betrieben werden, eignen sich gut für Stereo-Egomotion-Systeme. Merkmale, die am Horizont extrahiert werden, erhöhen die Stabilität der Rotation bei der Eigenbewegungsberechnung. Allerdings müssen diese Punkte im Unendlichen gesondert betrachtet werden, da die daraus resultierenden, sehr kleinen Schnittwinkel zu numerischen Instabilitäten führen können [Schneider u. a. 2012]. [A. Huang u. a. 2011] verwenden eine gewichtsreduzierte Kinect Kamera für die Indoornavigation eines Quadrocopters.

Bei Fußgängern werden Stereokamerasysteme oft zur Unterstützung Blinder eingesetzt. Neben der Navigation ist ähnlich wie bei den Fahrerassistenzsystemen auch die Hinderniserkennung zusammen mit deren Bewegungen und Geschwindigkeiten wichtig. Das Stereo Kamerasystem kann entweder am Kopf [Hernán Badino und Takeo Kanade 2011; Pradeep u. a. 2010], auf der Brust [Jirawimut u. a. 2003] oder auf den Schultern [Molton 1998] befestigt sein (siehe Abb. 6.3, S. 84). Für die Unterstützung eines Augmented-Reality-Systems kommt nur eine Montage am Kopf bzw. an einer Brille in Frage, wenn ein HMD zum Einsatz kommen soll. Das Kamerasystem

kann aber auch einfach in einem Tablet eingebaut sein, das in der Hand getragen wird.

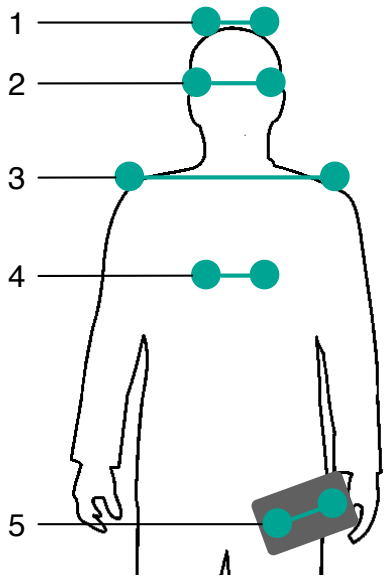


Abbildung 6.3: Unterschiedliche Befestigungsmöglichkeiten eines Stereokamerasystems beim Fußgänger. Kopf: Helm (1) oder Brille (2), Schultern (3), Brust (4) oder mobiles Gerät in der Hand (5)

Beim Fußgänger kann es je nach Befestigungsort der Kameras zu unterschiedlichen Bewegungsmustern kommen. Bei Systemen, die an Brust und Schulter befestigt sind, blicken die Kameras in die Bewegungsrichtung des Fußgängers. Die Vorgehensweise zur Bewegungsberechnung und Hinderniserkennung kann ähnlich wie bei den Fahrzeugsystemen gestaltet werden.

Bei der Befestigung am Kopf muss beachtet werden, dass die Blickrichtung des Kopfes nicht mit der Bewegungsrichtung übereinstimmen muss. Für den Menschen typische, schnelle Kopfbewegungen können im Mittel bis zu 50°/s annehmen [Holloway 1995], wodurch es zu unscharfen Bildern oder gar Abbruch des Merkmalstrackings kommen kann.

Gerade der Bedarf von AR an aktueller 3D-Information in der unmittelbaren Umgebung des Nutzers wird zu einer zunehmenden Verbreitung von Stereokameras oder Tiefensensoren in mobilen Endgeräten sorgen. PrimeSense bietet z. B. ein kleines Modul seines Tiefensensors für Tablets oder Smartphones an. Google wirbt mit einem Smartphone Prototyp „Project Tango“ mit integriertem Tiefensensor und Fisheye-Trackingkamera um Entwickler. Derartige Entwicklungen werden dafür sorgen, dass der Fußgänger als Stereo-Egomotion „Plattform“ in Zukunft an Bedeutung gewinnt.

Bei den fahrbaren, bodengebundenen Plattformen können die Freiheitsgrade der Bewegung eingeschränkt werden um die Stabilität zu erhöhen bzw. die Drift in den „unnötigen“ Parametern zu unterbinden. Bei einer auf zweidimensionalen Karten basierenden Navigationslösung könnten nur Lagekoordinaten und der Gierwinkel als freie Parameter gewählt werden. Auf den Roll- oder Nickwinkel bzw. die Höhe über der Karte könnte verzichtet werden.

6.2 Eigenbewegungsanalyse mit Stereokameras

In diesem Abschnitt werden einzelne Verfahren aus den oben genannten Anwendungsbereichen näher analysiert.

In der Arbeit von [Nistér u. a. 2004], die den Begriff *Visual Odometry* geprägt hat, kommt ein autonomes Offroad-Roboterfahrzeug zum Einsatz. In [Nistér u. a. 2006] wird das Verfahren auch mit Fußgängern (Kameras an Helm montiert) und im Straßenverkehr verwendet. Es werden maximal 5000 Harris Merkmale in 10×10 Bildregionen in allen vier Bildern detektiert. Die homologe Merkmale werden innerhalb eines festgelegten Disparitätsbereiches mit dem normalisierten Kreuzkorrelationskoeffizienten paarweise bestimmt. Aus den homologen Merkmalen eines Stereobildpaares werden jeweils 3D-Punkte trianguliert. Ein RANSAC-basiertes Drei-Punkt-Verfahren im linken Bild liefert die Pose der Stereokamera.

Dabei werden die Bildmerkmale zur Vermeidung von Drift möglichst lange über mehrere Aufnahmezeitpunkte hinweg verfolgt. [Nistér u. a. 2006] unterstreichen die Bedeutung von 2D-3D-Punktkorrespondenzen. Tests haben gezeigt, das dieses System bei einer Bewegungsberechnung nur mit den 3D-Punkten allein zu schlechteren Ergebnissen führt. Dies ist durch die, mit dem Abstand quadratisch zunehmende Unsicherheit der Tiefe von 3D-Punkten aus Stereosystemen zu erklären.

[Comport u. a. 2007] berechnen dichte Tiefenbilder aus den aufeinander folgenden Epipolarbildern. Die Beziehungen der vier Bilder der beiden Stereobildpaare werden mit dem quadrofokalen Tensor beschrieben. Damit können die Bilder des zweiten Stereobildpaares auf das erste Referenzbildpaar abgebildet werden. Da innere und äußere Orientierung bekannt sind, ist diese Abbildung nur noch von der relativen Bewegung der Stereokamera abhängig. Diese wird mit Hilfe der Minimierung einer nichtlinearen Kostenfunktion berechnet. Mit einem Schwellwert für den Median der absoluten Abweichungen (MAD) wird bestimmt, wann ein neues Referenzbildpaar festgelegt werden soll.

Der Szenenfluss, bei Daimler auch „6D-Vision“[2] genannt [Franke, Rabe u. a. 2005], basiert auf der Idee des optischen Flusses für alle vier Bilder der aufeinander folgenden Epipolarbildpaare [Franke, Gehrig u. a. 2008; Wedel, Brox u. a. 2011; Wedel und Cremers 2011]. Details zum Szenenfluss sind in Abschnitt 5.2.2 beschrieben.

[2] `http://www.6d-vision.de`, besucht im April 2014

Mit dem Szenenfluss können die für den Straßenverkehr typischen dynamischen Szenen mit mehreren Objekten unterschiedlicher Bewegungsrichtung und Geschwindigkeit segmentiert werden [Wedel, Brox u. a. 2011; Pfeiffer und Franke 2011]. Es können auch statische und bewegte Merkmale voneinander getrennt werden. Dies erhöht die Robustheit der Eigenbewegungsschätzung, welche auf einem Ansatz von [Hernan Badino 2004] basiert. Die Bewegung wird mit Hilfe der statischen 3D-Merkmale zwischen zwei Aufnahmezeitpunkten mit Horns Verfahren zur Berechnung der absoluten Orientierung mit Quaternionen [Horn 1987] geschätzt. Vier Zeitpunkte werden genutzt, um eine robuste, glatte Bewegungsschätzung zu erhalten („Smoothness Motion Constraint“).

Das dichte Stereomatching ist in einem FPGA (Field Programmable Gate Array) implementiert und der Szenenfluss mit Hilfe der GPU beschleunigt. Damit wird laut [Wedel, Brox u. a. 2011] eine Bildwiederholrate von 20 Hz bei einer Bildgröße von 320x240 Pixel erreicht.

Der Szenenfluss ist das leistungsfähigste Verfahren. Allerdings ist er auch sehr rechenintensiv. Typische autonome Fahrzeuge können aber auch Rechenleistung mitführen, die in einen Kofferraum passt. Bei einem mobilen ARS ist das anders. Hier ist die Rechenleistung begrenzt.

LibViso2 [Geiger 2014] ist eine Visual Odometry Bibliothek. Sie ist in C++ geschrieben und wurde am Institut Mess- und Regelungstechnik des KIT für die autonomen Fahrzeuge des „Team AnnieWay“[3] entwickelt.

Das Stereo-Egomotion-Verfahren ist in [Geiger, Ziegler u. a. 2011; Geiger 2013] und durch den Quellcode beschrieben. In allen vier Bildern werden 2D-Merkmale durch Filterung mit einem 5×5 Blob- und Cornerfilter und anschließender Nonmaximum und Non-minimum-suppression erzeugt. Blob min/max-Werte sowie Corner min/max-Werte liefern Merkmalskandidaten, die für das paarweise Matching genutzt werden. Das Matching selbst erfolgt über ein jeweils 11×11 großes Fenster mit horizontalen und vertikalen Sobelfilterantworten. Der Match wird mit SAD bewertet.

Die Reihenfolge des Matching erfolgt „im Kreis“ vom aktuellen linken Bild zum vorherigen linken Bild danach über das vorherige rechte Bild zum aktuellen rechten Bild. Dabei müssen die Merkmale innerhalb eines festgelegten Fensters liegen. Matches zwischen den Epipolarbildpaaren dürfen einen Schwellwert für den vertikalen Abstand von einem Pixel nicht überschreiten.

Statt einer Regularisierung der Flussparameter (u, v, p) wie im Szenenfluss werden bei [Geiger 2013] Konsistenztests durchgeführt. Zwischen den Aufnahmezeitpunkten wird eine Delauny Triangulation mit den 2D-Merkmalen des aktuellen linken Bildes durchgeführt. Für jedes Dreieck aus der Delauny Triangulation werden die Abstände der Bildkoordinaten (u, v) und Abstände der Disparitäten (p) der Eckpunkte (bzw. direkten Nachbarn) zwischen aktuellem und vorherigem Bild bestimmt und

[3] `http://www.mrt.kit.edu/annieway`, besucht im April 2014

mit Schwellwerten verglichen. Die Schwellwerte für Fluss und für Disparität haben einen Standardwert von fünf Pixel. Erst, wenn zwei dieser Abstände unterhalb der Schwellwerte liegen ist ein Punkt gültig.

Die Reduktion der Anzahl der Merkmale wird durch eine Unterteilung in 50×50 Regionen des Bildes erreicht. Für die Eigenbewegungsschätzung verbleiben somit etwa 200 bis 500 Merkmale. Die Bewegung $\boldsymbol{M}(R,t)$ wird über die Minimierung des Rückprojektionsfehlers in einem RANSAC-Schema berechnet [Geiger 2013]:

$$\hat{\boldsymbol{M}} = \arg\min_{\boldsymbol{M}} \sum_i \|x_i' - \boldsymbol{P'}\boldsymbol{X_i}\|^2 + \|x_i'' - \boldsymbol{P''}\boldsymbol{X_i}\|^2, \tag{6.1}$$

wobei $\boldsymbol{P'}$ und $\boldsymbol{P''}$ die Projektionsmatrizen des linken und rechten Bildes sind.

Als initiale Startwerte werden die sechs Parameter für Rotationen und Translation auf null gesetzt. Maximal 50 Samples, bestehend aus je drei Bildpunkten, die einen Mindestabstand überschreiten müssen, werden für einem ersten Durchlauf zufällig ausgewählt. Weitere Ausreißer werden mit einen Schwellwert für den maximal erlaubten Abstand der rückprojizierten Punkte bestimmt. Mit den verbleibenden Punkten erfolgt durch eine zweite Berechnung eine abschließende Verfeinerung der Bewegungsschätzung.

6.3 Eigenbewegungsanalyse mit RGB-D-Sequenzen

Ähnlich wie [Comport u. a. 2007] im Stereofall, nutzen [Steinbrücker u. a. 2011] eine Energieminimierung zur Schätzung der Bewegung. Dafür wird das Farbbild zunächst in ein Intensitätsbild I umgewandelt und auf die Oberfläche S des Tiefenbildes abgebildet. Die unbekannte Bewegung zwischen den Aufnahmezeitpunkten t_0 und t_1 ermöglicht zusammen mit der Oberfläche $S(t_0)$ zum Zeitpunkt t_0 eine eindeutige Abbildung (*warp* ω) der Intensitätswerte $I(t_1)$ der zweiten Aufnahme auf die Intensitätswerte der ersten Aufnahme $I(t_0)$.

Es wird angenommen, dass die beobachtete Szene statisch ist und die Intensität der Oberfläche zwischen allen beobachteten Aufnahmen konstant bleibt, ähnlich wie beim optischen Fluss. Die Bewegung wird als Schraube erster Art (*twist*) ξ modelliert. Für das Energiefunktional $E(\xi)$ gilt unter der Annahme, das zum Zeitpunkt t_0 keine Bewegung stattfindet [Steinbrücker u. a. 2011]:

$$E(\xi) = \int_\Omega (I(\omega(x,t_1),t_1) - I(x,t_0))^2 \, \mathrm{d}x. \tag{6.2}$$

Die Kamerabewegung, die diese Bedingung mit dem kleinsten Fehler über alle Pixel im gesamten Bild (Ω) erfüllt, ist die gesuchte Lösung.

Da ξ nur für kleine Bewegungen gilt, wird das Ergebnis iterativ mit Bildpyramiden für Intensitätsbilder, Tiefenbilder und Gradientenbilder vom Groben ins Feine berechnet. Die Anzahl der Iterationen in den einzelnen Pyramidenstufen ist mit

Werten zwischen minimal 7 und maximal 10 recht klein. Die Autoren erreichen mit ihrer Implementierung auf einer Intel Xeon CPU mit 2,27 GHz eine Framerate von 12.5 Hz für die kontinuierliche Bewegungsschätzung. In OpenCV [*Open Source Computer Vision Library (OpenCV)* 2015] ist eine Version dieses Verfahrens in der Methode `RGBDOdometry` integriert.

Der bisher beschriebene Ansatz wird in [Kerl u. a. 2013] für dynamische Szenen erweitert. Mit Hilfe einer robusten Gewichtsfunktion, die von der t-Verteilung abgeleitet ist, werden bewegte Objekte für die Eigenbewegungsschätzung ignoriert. Sprünge in der resultierenden Kamera-Trajektorie können mit der Einführung eines konstanten, normalverteilten Geschwindigkeitsmodells vermieden werden.

[A. Huang u. a. 2011] haben eine Kinect Kamera auf ein Gewicht von 115 g reduziert und diese auf einem autonom fliegenden Quadrocopter montiert (siehe auch UAV Beispiel in Abb. 6.2). Zusätzlich ist noch eine IMU montiert. Da das „micro air vehicle“ (MAV) für Flüge innerhalb von Gebäuden konzipiert ist, kann die Kinect Kamera hier problemlos eingesetzt werden. Für den Gebrauch im Freien ist der Musterprojektor der Kinect Kamera bei Tageslicht selbst im Schatten nicht hell genug.

Das Verfahren ist Teil eines entkoppelten SLAM Systems. Die Bewegungsschätzung erfolgt dabei mit Hilfe der VO. Um die Drift zu reduzieren, werden in periodischen Abständen Positionskorrekturen vom SLAM System geliefert. Alle Berechnungen finden auf dem Navigationscomputer des MAV statt. Dieser verfügt über eine 1,86 GHz „Core2 Duo“ CPU und soll in der Lage sein, alle Berechnungs- und Steueraufgaben in Echzeit durchzuführen.

2D-Merkmale werden in einer Gauß-Pyramide mit dem FAST Detektor [Rosten und Drummond 2006] (Features from Accelerated Segment Test) bestimmt. Jede Pyramidenstufe ist in 80×80 Pixel große Regionen unterteilt, um eine gute Verteilung der Merkmale im Bild zu erzielen. Aus jeder Region werden 25 Merkmale mit der besten Eckenbewertung des FAST-Detektors gewählt. Das Matching der Merkmale wird mit einem Deskriptor durchgeführt. Die Bewertung erfolgt über SAD, da diese mit Intel SSE2 Befehlen schnell berechnet werden können.

Rotationen erzeugen auf diesem System die größten Bewegungen zwischen den Bildern. Die Schätzung von Rotationen ohne Translation hilft, das Suchfenster für das Matching zwischen den Bildern einzuschränken. Damit kann die Zahl der Ausreißer verringert werden. Scheinbar ist das graphenbasierte Deskriptor Matching dafür nicht robust genug. Es ist bemerkenswert, dass die vorhandene IMU nicht für die initiale Rotationsschätzung herangezogen wird. Vielmehr wird diese durch eine Minimierung des quadratischen Pixelfehlers zwischen den Bildern benachbarter Aufnahmezeitpunkte geschätzt.

Zur eigentlichen Bewegungsschätzung wird zuerst Horns Orientierungsverfahren [Horn 1987] für initiale Startwerte verwendet. Danach kommt ein nichtlineares Verfahren zur Minimierung des Rückprojektionsfehlers zum Einsatz. In einem ersten Durchlauf werden Ausreißer entfernt und im nächsten, abschließenden Schritt die

finale Bewegung mit den verbleibenden Inliern bestimmt. Die Drift wird mit Schleifenschluss auf Basis einer Keyframetechnik reduziert, in dem die Bewegung zwischen dem aktuellen Bild und dem letzen Keyframe berechnet wird. Für das beschriebene Verfahren ist eine freie C++-Implementierung erhältlich: „libfovis - Fast Odometry from VISion“ [A. Huang 2014].

[Lui u. a. 2012] und [Izadi u. a. 2011] nutzen nur das Tiefenbild um die Eigenbewegung mit einem ICP Verfahren zu berechnen. Dabei werden die Tiefenwerte direkt genutzt (inverse Tiefe), ohne diese zuvor in euklidische 3D-Punkte umzuwandeln.

[Dominguez Quijada u. a. 2013] nutzen das Farbbild um mit dem KLT Bildmerkmale zu verfolgen und das Tiefenbild um 3D-Punkte aus den Merkmalen zu generieren. Die Tiefe wird in einer Umgebung des 2D-Merkmals mit einem Radius von drei Pixel um den Punkt gemittelt. Die bedeutet, das die Tiefenwerte in einem Fenster von 7×7 gemittelt werden.

Der Kern in [Dominguez Quijada u. a. 2013] ist die Filterung der Ausreißer. 3D-Merkmale werden über mehrere Bilder verfolgt. Dabei sollte die 3D-Position eines Merkmals sich nicht zu sehr ändern. Innerhalb der lokalen Nachbarschaft befindet sich ein 3D-Punkt zu Beginn seiner Verfolgung in einem Gleichgewicht. Die „Kräfte“, die den 3D-Punkt während der Merkmalsverfolgung aus diesem Gleichgewicht bringen, dürfen einen Schwellwert nicht überschreiten, sonst wird der Punkt für die Bewegungsschätzung verworfen.

In [Israël und Plyer 2013] werden 3D-Kontouren mit Hilfe des Sobelfilters aus dem Tiefenbild extrahiert. Mit dem Übergang von 3D-Punktwolken zu 3D-Kontouren kann die Datenmenge reduziert werden. Lange Kanten werden dabei mit einem Gewichtsfaktor bevorzugt. Das Matching erfolgt über einer Maximum-Likelihood Funktion. Die Echtzeitfähigkeit des Verfahrens wird mit einer GPU-basierten Implementierung erreicht.

6.4 Zusammenfassung der Ansätze zur Eigenbewegungsberechnung

Stereokamera und Tiefenbildkamera liefern theoretisch die gleichen Daten. Bei beiden kann man über die inverse Tiefe zu 3D-Punkten gelangen. Die Tiefenkamera liefert grundsätzlich ein nahezu dichtes Tiefenbild, wenn der maximale Messbereich in der Szene nicht überschritten wird. Die Erfassung einer dichten Oberfläche ist bei Stereoverfahren abhängig von der Textur. Aus diesem Grund wird bei den Stereoverfahren öfter auf Lösungen mit relativ wenig Merkmalen zurückgegriffen, während bei RGB-D-Sequenzen häufiger mit 3D-Punktwolken gearbeitet wird und die Verfahren zum Vergleich oft eine ICP-Variante heranziehen.

Der Szenenfluss ist bei beiden Varianten zu finden. Auch KLT-basierte Varianten findet man bei beiden Aufnahmesystemen.

Wichtiger als das eigentliche Verfahren zur Schätzung der Bewegung ist jedoch die sorgfältige Filterung nach Ausreißern. Dabei kann man die Lösungen bezüglich der Annahme über die Umgebung in zwei Gruppen unterteilen: statische Szenen und dynamische Szenen. Die Segmentierung der Tiefenszene nach Bereichen unterschiedlicher Bewegungsrichtungen und -Geschwindigkeiten bringt für die Eigenbewegungsschätzung den Vorteil, auf rein statische Merkmale zugreifen zu können. Eine geeignete Filterung nach Bewegungsmustern benachbarter Merkmale kann diesen Nachteil bei einer als statisch angenommen Szene aber ausgleichen.

Allgemein bieten die verschiedenen Ansätze der verfügbaren Veröffentlichungen zur Eigenbewegungsanalyse bei Stereo- und Tiefenbildkameras Lösungen zu folgenden Bereichen:

- Anzahl verwendeter Bilder zwischen zwei Zeitpunkten
- Anzahl der Zeitpunkte; meist zur Vermeidung von Drift
- Art der Korrespondenzen: 3D oder 2D-3D
- Extraktion von Merkmalen
- Strategien, die für eine gute Verteilung von Merkmalen sorgen
- Matching der Merkmale
- Detektion von Ausreißern
- Modellierung der Bewegung
- Art der Regularisierung
- Lineare bzw. nichtlineare Lösungsansätze für die Bewegungsschätzung
- Echtzeitfähige Implementierung (GPU, FPGA)

6.5 Eigener Ansatz auf Basis des EPnP

Wenn nach geeigneter Filterung korrespondierende 2D-Merkmalspunkte mit ihren berechneten oder gemessenen 3D-Objektkoordinaten vorliegen, sollte ein robustes Rückwärtsschnittverfahren wie EPnP (siehe 3.4.2) für Stereobildsequenzen und RGB-D-Sequenzen gleichermaßen für die Bewegungsschätzung einsetzbar sein. Dies führt zu einem eigenen Ansatz zur Berechnung der Eigenbewegung, der im Folgenden beschrieben wird.

Eingangsdaten sind die Bilder der jeweils aufeinander folgenden Zeitpunkte t und $t+1$, wie sie zu Anfang des Kapitels beschrieben wurden (siehe Abb. 6.1, S. 82). Die inneren Orientierungen der Bilder sind nach erfolgter Kamerakalibrierung bekannt. Beim Stereofall sind dies die zwei zeitlich benachbarten Epipolarbilder, d.h. auch die relative Orientierung ist nach der Kalibrierung des Stereokamerasystems bekannt und wird für die Aufnahme und Auswertung einer Bildsequenz als fest angesehen. Bei der Kinect Kamera ist zusätzlich zur Kamerakalibrierung die Überlagerung von RGB- und Tiefenbild zu beachten (siehe Abb. 4.18, S. 63). Das RGB-Bild wird in ein Intensitätsbild umgewandelt. Die Farbinformation wird nicht verwendet.

Der Ablauf ist in Abb. 6.4 skizziert. Im Stereofall werden 2D-Eckmerkmale im linken Epipolarbild, bei der Kinect im RGB-Bild mit Shi-Tomasi („Good Features to Track“ [Shi und Tomasi 1994]) extrahiert. Eine Unterteilung der Bilder in gleich große, rechteckige Regionen („Bucketing“) soll für eine gute Verteilung der Punkte im Bild sorgen. In Abb. 6.5b ist diese Unterteilung durch die grünen Linien dargestellt. Mit einem Pyramidenansatz des Kanade-Lucas Trackers [Bouguet 2000] werden diese von t nach $t+1$ verfolgt. Bei der Bildgröße von 640×480 für Stereokameras und Kinect RGB-Kamera erreicht der Algorithmus mit einer dreistufigen Bildpyramide einen Suchbereich von 75 Pixel für die Merkmalsverfolgung (siehe 5.2.3).

Dabei wird ein Vorwärts-Rückwärts-Schema ähnlich wie in [Kalal u. a. 2010] angewandt: Die 2D-Merkmale in $t+1$ werden auch nach t verfolgt. Der euklidische Abstand zwischen Ausgangsmerkmal und Vorwärts-Rückwärts-Merkmal darf den Schwellwert für einen maximalen Abstand von einem Pixel nicht überschreiten. Die gültigen Punkte werden anschließend noch mit einem NCC-Schwellwert gefiltert. Damit stehen korrespondierende Merkmale für beide Aufnahmezeitpunkte t und $t+1$ zur Verfügung (siehe rote Linien in Abb. 6.5b).

Für die so verbleibenden 2D-Merkmale werden im Stereofall im rechten Bild korrespondierende Merkmale gesucht und trianguliert (Schema siehe Abb. 6.5a). Für die Korrespondenzanalyse wird ebenfalls das KLT-basierte Vorwärts-Rückwärts-Schema verwendet, da die Bildpyramiden für das linke Epipolarbild bereits vorhanden sind. Der Abstandstest der Stereokorrespondenzen wird nur auf die y-Koordinaten angewandt.

Im RGB-D-Fall werden Tiefenwerte zu den 2D-Merkmalen über einen bilinearen Abgriff aus dem Tiefenbild ermittelt. Wenn einer der hierfür benötigen vier Werte

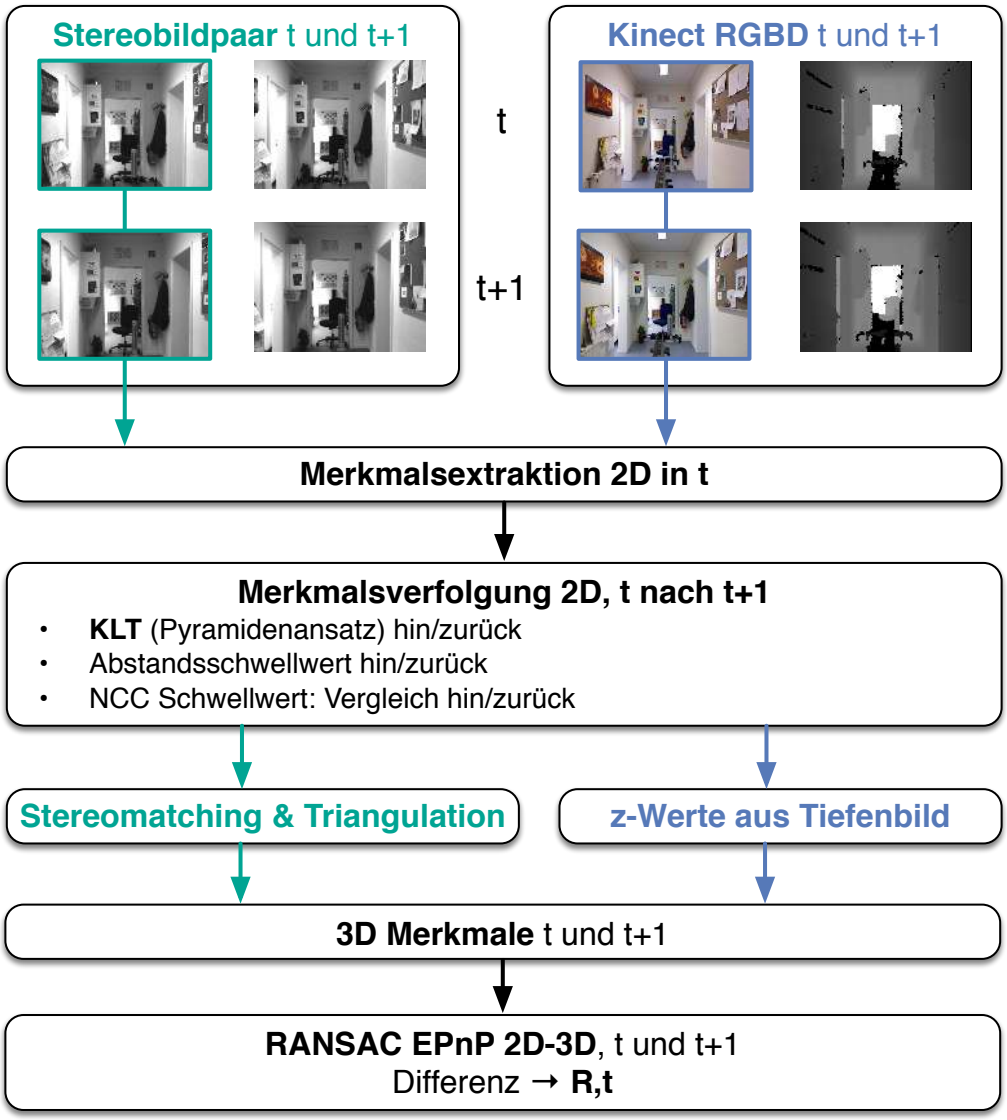
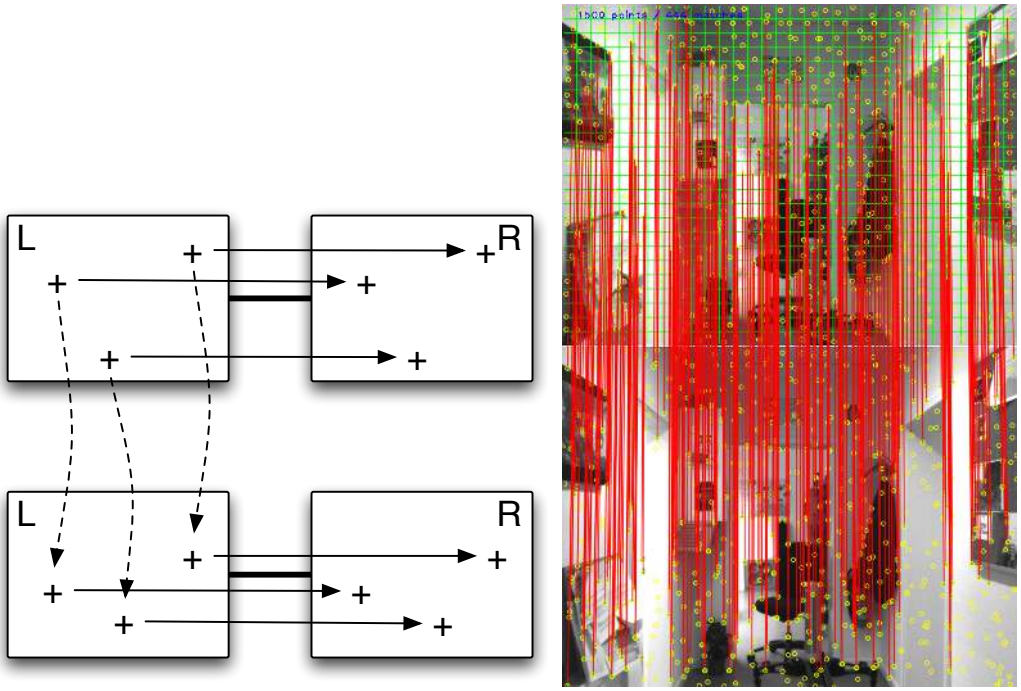


Abbildung 6.4: Ablauf der Eigenbewegungsanalyse mit EPnP für Stereobildpaar und Kinect RGB-D. Die beiden Varianten unterscheiden sich nur in der Berechnung der 3D-Objektpunkte zu den extrahierten 2D-Bildmerkmalen



(a) Ablauf der 2D-3D-Merkmalsverfolgung beim Stereokamerasystem

(b) 2D-Merkmalsverfolgung

Abbildung 6.5: Ablauf der Merkmalsverfolgung am Beispiel des Stereokamerasystems (linke Kamera, Zeitpunkte t, $t+1$). Für die Merkmalsextraktion ist das Bild in gleich große Regionen unterteilt (grüne Linien). Die gefundenen Zuordnungen sind durch rote Verbindungen dargestellt

null ist, also keine Tiefe besitzt, wird das 2D-Merkmal verworfen. Die so entstandenen 3D-Merkmale werden nach gültigem Tiefenbereich und maximal zulässiger Entfernung gefiltert. Dabei müssen die 3D-Merkmale in beiden Zeitpunkten t und $t+1$ gültig sein. In der Regel sind zu diesem Zeitpunkt noch etwa 300-500 gültige Merkmale vorhanden.

Mit den verbleibenden 2D-3D-Punktkorrespondenzen wird jeweils eine RANSAC-basierte äußere Orientierung mit dem EPnP-Algorithmus berechnet (siehe 3.4.2). Die Differenz dieser beiden Lösungen ist die Bewegung zwischen beiden Zeitpunkten t und $t+1$.

Falls die Anzahl der 2D-3D-Punktkorrespondenzen unterhalb eines Schwellwertes liegt oder die Bewegung nach Translation und Rotation getrennte Schwellwerte für eine maximal zulässige Bewegung überschreitet, wird keine Lösung an den Kalman Filter weitergegeben.

Im folgenden Kapitel werden beide Varianten dieses Verfahrens mit einer Auswahl der hier vorgestellten Ansätze vergleichen.

7 Vergleich der Verfahren

Die im vorangehenden Kapitel vorgestellten Verfahren haben alle die Schätzung der relativen Eigenbewegungen entweder aus Stereobildsequenzen oder RGB-D-Bildsequenzen zum Ziel.

Im Folgenden werden die in Tabelle 7.1 aufgeführten Verfahren miteinander verglichen. Die Verfahren Stereo EPnP und RGB-D EPnP sind die eigenen Ansätze, die in Abschnitt 6.5 beschrieben wurden. Das verwendete Aufnahmesystem be-

Verfahren	Aufnahmesystem	Datentyp	Merkmale
Stereo EPnP	Stereo	Epipolarbildpaar	2D+3D
RGB-D EPnP	RGB-D	RGB + Tiefenbild	2D+3D
RGB-D Odometry [Steinbrücker u. a. 2011]	RGB-D	RGB + Tiefenbild	2D+3D, dicht
GICP [Segal u. a. 2009]	RGB-D	Punktwolke aus Tiefenbild	3D, dicht
Stereo LibViso [Kitt u. a. 2010]	Stereo	Epipolarbildpaar	2D+3D

Tabelle 7.1: Untersuchte Verfahren. Von jeden Datentyp werden jeweils zwei aufeinanderfolgende Zeitpunkte benötigt

steht aus einem Stereokamerasystem und einer Kinect Kamera, die fest miteinander verbunden sind (siehe Abschnitt 4.5). Es werden vier Aufnahmen synchron bei einer Wiederholrate von max. 12 Hz gespeichert: jeweils das linke und rechte Bild des

monochromen Stereokamerasystems, das Kinect RGB-Bild und das Kinect Tiefenbild. Alle Aufnahmen haben eine Bildgröße von 640×480 Bildpunkten. Die schwache Leuchtstärke des Kinect Musterprojektors beschränkt die Aufnahme der Testsequenzen auf Indoorszenen.

7.1 Methodik

Für die Evaluation der Systeme werden Referenzwerte für die Bewegungen (relative Posen) benötigt. Im Idealfall sind Referenzwerte für die Bewegungen in einer höheren Genauigkeit für jeden Aufnahmezeitpunkt vorhanden.

In der Regel stammen diese Referenzwerte von einem Trackingsystem, das sechs Freiheitsgrade sowohl in einer höheren Frequenz und als auch in einer höheren Genauigkeit als die zu testenden Systeme liefert. Mit einem anschließenden Abgleich der Aufnahmezeiten kann man Referenzwerte und Messwerte zusammenführen und vergleichen.

[Dominguez Quijada u. a. 2013] verwenden Bildsequenzen, die mit einem Roboterarm aufgenommen wurden, an dem die Kamera befestigt wurde. Dieser hat eine Wiederholgenauigkeit von ± 0,05 mm. Damit sind die programmierten Bewegungen des Roboterarms mehr als genau genug, um als Referenzwerte geeignet zu sein.

[Sturm, Engelhard u. a. 2012] verwenden einen optisches Motion-Capture-System bestehend aus acht Trackingkameras. An der Kamera befestigte, retroreflektierende Marker werden mit 100 Hz verfolgt. Die relative Anordnung aller, an der Kamera befestigter Marker wird als fest angenommen. Damit liegt die Kamerabewegung mit sechs Freiheitsgraden vor. Die mit einem optischen Tracker erreichbare Genauigkeit hängt von der Kamerakonfiguration und den Ausdehnungen des beobachteten Raumes ab.

Im Outdoorbereich, bei Fahrzeugen [Kitt u. a. 2010] und UAV greift man auf eine Kombination von GNSS und IMU zurück um eine Trajektorie mit max. sechs Freiheitsgraden zu erhalten, je nach Anwendungsfall. Bei Fahrzeugen könnte man die Freiheitsgrade auf die 2D-Position und den Gierwinkel (drei DoF) beschränken. Als Ersatz für die beschriebenen Systeme kommen in dieser Untersuchung zwei Methoden zum Einsatz: eine Kameragleitschiene und bildbasierte Offlineverfahren zur Bewegungsberechnung mit Bündelblockausgleichung.

7.1.1 Kameragleitschiene

Für einfache lineare Bewegungen kommt eine *Kameragleitschiene* zum Einsatz, die auf zwei Stativen aufgebaut wird und somit eine geradlinige Bewegung eines Aufnahmesystems im Raum erlaubt. Mit Hilfe der metrischen Skala können mehrere statische Aufnahmen mit gleichem Abstand zueinander zu einer Sequenz mit bis zu 1,05 m Gesamtstrecke erzeugt werden (siehe Abb. 7.1, S. 97). Die erreichbare Ge-

nauigkeit zur Positionierung des Gleitschlittens beträgt 1 mm. Die absolute Lage der Kameragleitschiene im Raum kann mit einfachen Mitteln bestimmt werden.

Auf diese Weise können Referenztrajetorien für typische lineare Bewegungen erzeugt werden, die eine Person etwa in 1 s zurücklegen kann. Die hier untersuchten bildbasierten Trackingverfahren sollen in der Lage sein Strecken dieser Länge zwischen zwei absoluten Positionsangaben, etwa durch GNSS, zu überbrücken.

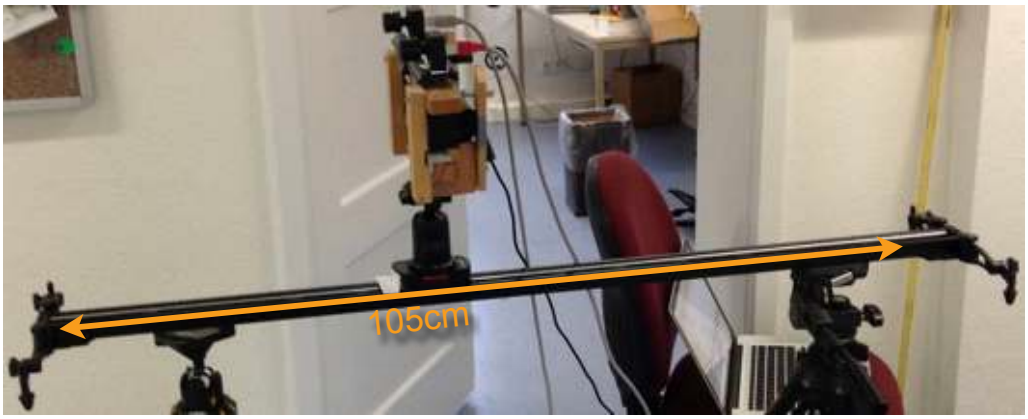


Abbildung 7.1: Die Kameragleitschiene im Einsatz. Das Aufnahmesystem ist an einem Kugelkopf auf der Gleitschlitten montiert. Erreichbare Gesamtlänge: 105 cm

7.1.2 Offlineverfahren

Für längere Sequenzen mit komplexeren Bewegungen, die mehrere Freiheitsgrade besitzen, werden die Aufnahmen mit einem bildbasierten *Offlineverfahren* orientiert um Referenzwerte zu erhalten.

- VoodooTracker ist eine Software zur Berechnung der Kameratrajektorie aus einer Bildsequenz. Die Implementierung basiert auf Arbeiten von [Thormählen 2006]. Aus den 3D-Koordinaten der verfolgten Merkmalspunkte kann der Maßstab anhand einer bekannten Strecke berechnet werden.

- Apéro [Pierrot-Deseilligny und Clery 2011] ist ein freies und quelloffenes Photogrammetrie-Softwarepaket mit dem innere und äußere Orientierungen einer Bildsequenz sowie die 3D-Koordinaten der Merkmalspunkte über eine Bündelblockausgleichung berechnet werden können. Zur Lösung des Maßstabs können Passpunkte eingeführt werden.

Die Eignung beider Verfahren zur Erzeugung von Referenzwerten kann wiederum mit Sequenzen auf der Kameragleitschiene getestet werden. Wegen des größeren Öffnungswinkels werden die Kinect RGB-Bilder für die Orientierungen des Aufnahmesystems verwendet.

7.2 Testsequenzen

Für die Evaluierung werden die akkumulierten Bewegungen jeweils über die gesamte Sequenz pro Verfahren aus Tabelle 7.1 berechnet. Zum Vergleich werden der *Absolute Trajectory Error* (ATE), der ausschließlich die Translationen der Bewegung bewertet und der *Relative Pose Error* (RPE, siehe Abschn. 3.5 S. 36) berechnet und in Tabellen gegenübergestellt. Zur Berechnung der ATE und RPE für 6DoF Trajektorien wurde die Implementierung[1] von [Sturm, Magnenat u. a. 2011] verwendet. Die Bewertungen der einzelnen Verfahren sind für die verschiedenen Sequenzen in den Tabellen 7.2, 7.4 und 7.5 aufgelistet. Die Spalten „T-RPE“ und „R-RPE“ sind der *Relative Pose Error* jeweils für Translationen und Rotationen. Das Kürzel „KF“ bedeutet, dass die berechnete Trajektorie mit einem Kalman Filter geglättet wurde. Der jeweils beste (=niedrigste) Wert ist **fett** hervorgehoben.

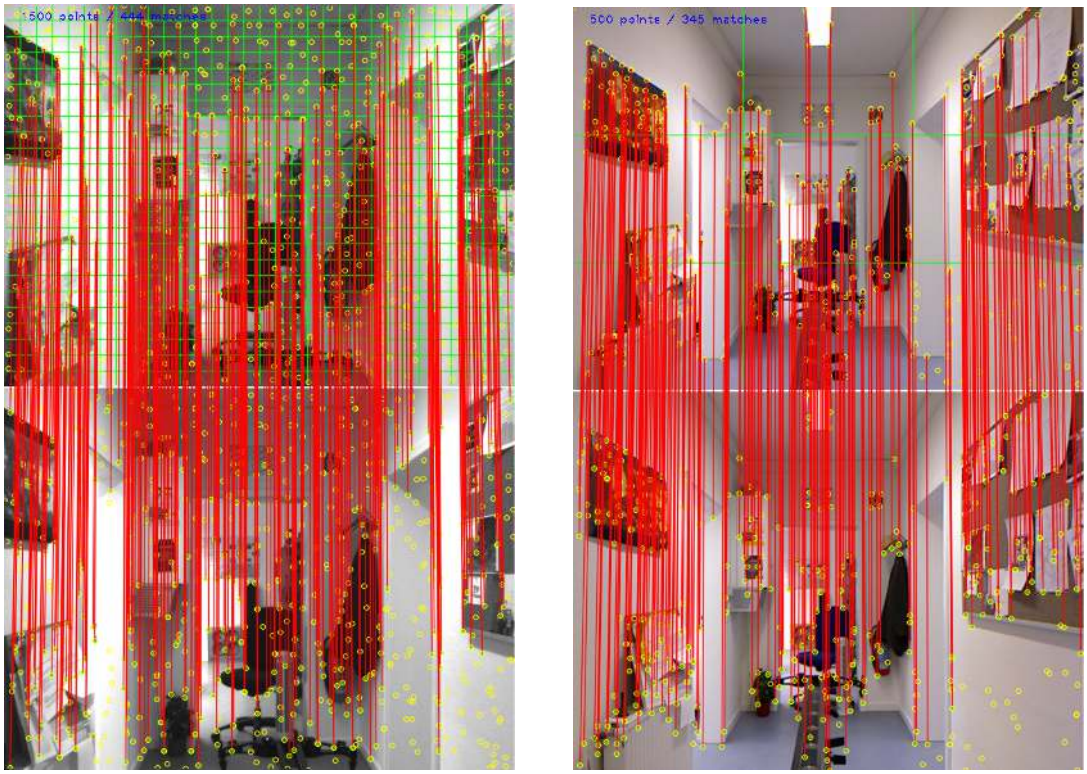
Wegen der schwachen aktiven Beleuchtung der Kinect Kamera ist diese im Freien nicht einsetzbar. Die Indoor-Sequenzen im Gang sind ein Kompromiss um Verfahren für beide Kamerasysteme (Stereokamera und Kinect) miteinander vergleichen zu können.

7.2.1 Sequenz: „Schiene“

Die Referenzdaten dieser Sequenz wurden mit der Kameragleitschiene erzeugt. Die Schiene wurde auf zwei Stativen in der Mitte eines Ganges von 1,7 m Breite waagerecht aufgestellt (siehe Abb. 7.3a, S. 100). Die Ausrichtung des Aufnahmesystems war nach vorne, in Bewegungsrichtung gerichtet. Die Translation erfolgte in 5 cm-Schritten in Richtung der X-Achse. Die Referenzwerte bestehen somit nur aus X-Werten von 0 m bis 1,05 m. Alle anderen Werte (Y, Z und die Rotationswinkel) sind null, soweit dies mit den Einstellmöglichkeiten des Kugelkopfes möglich war. Die relativen Bewegungen zwischen den Aufnahmen können für alle Werte außer X als null angesehen werden.

Diese Szene ist nicht ideal für Verfahren, die Eckmerkmale verfolgen (siehe Abb. 7.3a, S. 100). Die weiße Wand und die Decke sind in größeren Bereichen im Bild texturlos. Der Fußboden hat eine schwache Textur, die wegen der geringen Auflösung der Kameras schlecht für die Merkmalsverfolgung geeignet ist. Abb. 7.2 zeigt die Punktzuordnungen (Matches) aus dem Vorwärts-Rückwärts-Schema mit dem Kanade-Lucas Tracker, das jeweils in den Verfahren Stereo EPnP und RGB-D EPnP (siehe Kap. 6.5) zum Einsatz kommt. Die Punktzuordnungen in beiden Beispielen sind wegen einiger untexturierter Bereiche nicht gleichmäßig über das Bild verteilt, so wie man es für einen Rückwärtsschnitt wünschen würde. Die Aufnahmen wurden in Bereiche (Bucketing, siehe 6.5) unterteilt um dennoch eine möglichst

[1] „rgbd-benchmark-tools“, Python Skripte von `https://vision.in.tum.de/data/datasets/rgbd-dataset/tools`, besucht im März 2015



(a) Stereo EPnP: 345 Matches **(b)** RGB-D EPnP: 444 Matches

Abbildung 7.2: Beispiel für gefundene Matches (rote Linien) in Stereo EPnP und RGB-D EPnP

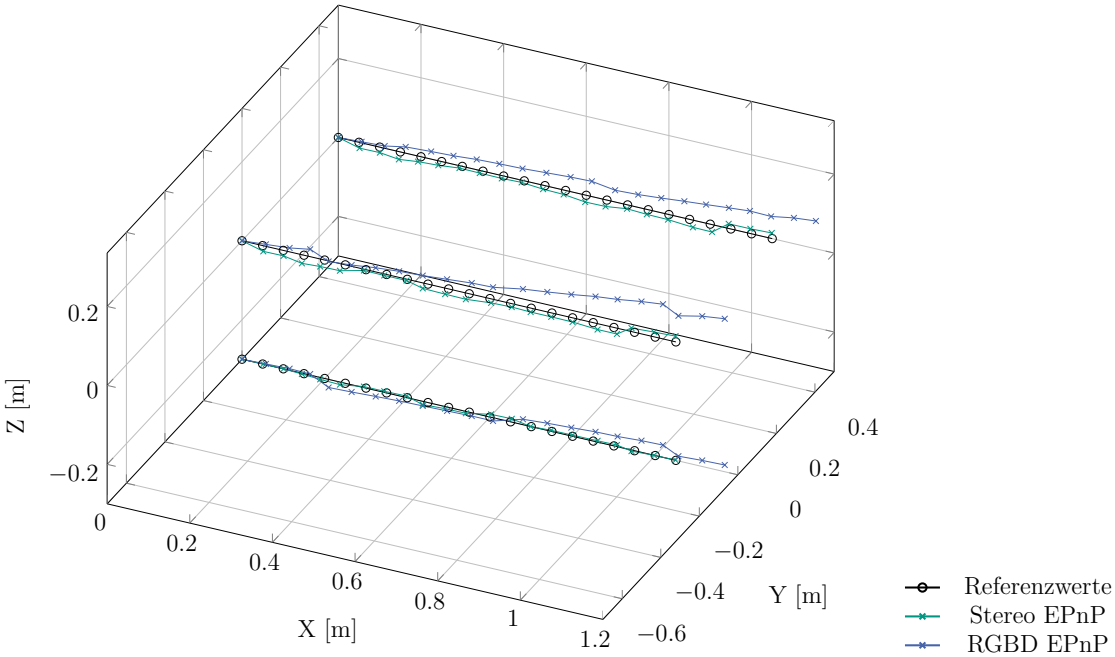
gute Verteilung der detektierten Merkmale über die Bildfläche zu erreichen. Diese Bereiche sind in den oberen Aufnahmen durch die grünen Linien gekennzeichnet.

Nach der Berechnung der 3D-Koordinten und der EPnP-Ransac-Schätzung verbleiben in dem gezeigten Beispiel bei Stereo EPnP noch 36 robuste 3D-2D Punktzuordnungen. Wegen schlechter Zuordnungen von Merkmalen zwischen linkem und rechten Bild der Stereokamera werden während der EPnP-Ransac-Berechnung viele der bereits gefundenen 3D-2D-Punktzuordnungen doch noch verworfen. Bei dem Verfahren RGB-D EPnP sind es noch 247, da hier die Disparitäten direkt aus dem aktiv gemessenen Tiefenbild abgelesen und in 3D-Koordinaten umgewandelt werden können. Unter den 197 verworfenen Punktzuordnungen sind bei der Kinect-Lösung oft Punkte dabei, die an Kanten liegen oder auf Flächen, die beim Folgebild unter einen ungünstigen Winkel fallen und somit mit keinem Tiefenwert mehr besetzt sind und auch wegfallen.

In Abb. 7.3b ist die Trajektorie der Referenzwerte innerhalb einer Kinect Punktwolke dargestellt. Die Koordinatenachsen sind an der Startposition dargestellt: X-Achse in rot, Y-Achse in grün und die Z-Achse in blau. Diese Referenzwerte sind

(a) Aufnahmekonfiguration mit Gleitschiene. Viele Bereiche im Bild sind nur schwach oder überhaupt nicht texturiert

(b) Kinectpunktwolke und Referenzpositionen aller Aufnahmen. Die Koordinatenachsen sind an der Startposition dargestellt: X-Achse in rot, Y-Achse in grün und die Z-Achse in blau



(c) Berechnete 3D-Trajektorien für die eigenen Verfahren Stereo EPnP und RGB-D EPnP

Abbildung 7.3: Aufnahmekonfiguration mit Gleitschiene

Methode	ATE	ATE KF	T-RPE	R-RPE	T-RPE KF	R-RPE KF
Stereo EPnP	0.011	0.032	0.442	0.212	0.431	**0.082**
RGB-D EPnP	0.037	0.013	0.057	0.511	0.030	0.344
RGB-D Odometry	0.068	0.033	0.092	0.296	0.045	0.225
GICP	0.085	0.064	0.240	3.755	0.288	3.876
Stereo LibViso	**0.005**	**0.005**	**0.010**	**0.202**	**0.010**	0.198

Tabelle 7.2: Vergleich der Werte für ATE (Absolute Translation Error), RPE (Relative Pose Error). KF=Kalman Filter; T-RPE = RPE der Translationen; R-RPE = RPE der Rotationen (Sequenz „Schiene“)

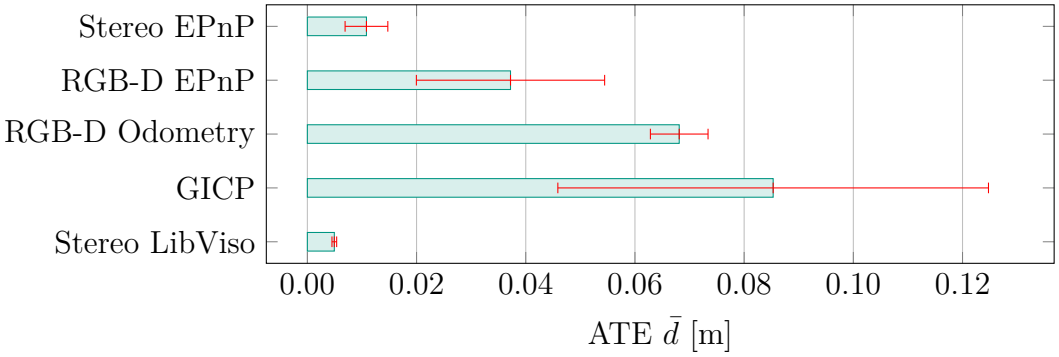


Abbildung 7.4: Vergleich der ATE (Sequenz „Schiene“)

auch in dem 3D-Plot in Abb. 7.3c als Gerade im Raum (in schwarz) zu sehen. Im Vergleich dazu sind die berechneten Trajektorien der eigenen Verfahren Stereo EPnP (grün) und RGB-D EPnP (blau) geplottet. Bei RGB-D EPnP kann man deutlich die Drift und die, im Vergleich zur Referenztrajektorie zu lang ermittelte Strecke erkennen. In Abb. 7.5 sind die Translationen pro Koordinatenachse und in Abb. 7.6 die Rotationen um jede Koordinatenachse aller Verfahren aus Tabelle 7.1 für die Sequenz „Schiene“ dargestellt.

Die Ergebnisse dieser Sequenz sind für alle Verfahren in Tabelle 7.2 aufgeführt. Der Mittelwert der Translationsfehler $\bar{d}$ des ATE ist zusätzlich in Abb. 7.4 grafisch dargestellt. Der rote Balken zeigt das dazugehörige 3σ-Konfidenzintervall. In dieser Auswertung erreicht das Verfahren Stereo LibViso in allen Werten außer der letzten Spalte in Tabelle 7.2 das beste Ergebnis. Es endet bei X=102,9 m und unterschreitet damit die Gesamtstrecke von 1,05 m nur um 2,1 cm. Diese entspricht einer Abweichung von 2 % von der Gesamtstrecke. Die Anteile an der Referenzstrecke sind für alle Verfahren in Tabelle 7.3 aufgeführt. Das eigene Verfahren Stereo EPnP erreicht mit einem 99,9 % Anteil der akkumulierten Bewegungen zur Referenzstrecke das

Verfahren	% Referenzstrecke von 1,05m
Stereo EPnP	99,9
RGB-D EPnP	110,1
RGB-D Odometry	124,8
GICP	78,6
Stereo LibViso	98,0

Tabelle 7.3: Anteile der akkumulierten Strecken pro Verfahren zur Referenzstrecke (Sequenz „Schiene“)

bessere Ergebnis. Allerdings zeigt das Verfahren eine kleine Drift durch Rotationen um die X-Achse (siehe Abb. 7.6, S. 104). Die Variante RGB-D EPnP liefert über die gesamte Sequenz zu hohe Translationen, so dass am Ende die Referenzstrecke um 10,1 % überschritten ist. Die RPE Werte sind grundsätzlich schlechter als die ATE Werte, da sie die Drift bewerten. Das GICP-Verfahren schneidet am schlechtesten ab. Die Trajektorie erreicht mit 0,83 m Länge in X nur 79 % der Gesamtstrecke. Das mag daran liegen, das die Geometrie der Punktwolke des Ganges hauptsächlich Flächen an den seitlichen Wänden und wenig Flächen an Decke, Boden und in Bewegungsrichtung bietet. Die Drift „nach unten“ ab der vierten Aufnahme, die in den Translationen für die Z-Achse in Abb. 7.5 zu sehen ist, kann durch das Wegfallen der Gangdecke und des Gangbodens aus der Punktwolken erklärt werden. Die Wand im Raum am Ende des Ganges befindet sich außerhalb der Kinect Reichweite, so dass in Bewegungsrichtung nur die kleinen Flächen um den Türrahmen verbleiben.

Die Varianten mit Kalman Filter führen zu einer geglätteten Trajektorie ohne Sprünge. Schlechte Bewegungsschätzungen können aber beim Akkumulieren zu einer Richtungsänderung führen, die mangels zusätzlicher absoluter Bewegungsschätzungen durch andere Sensoren oder Verfahren im Kalman Update nicht wieder korrigiert werden. Dies kann sich für ATE und RPE sowohl positiv als auch negativ auswirken. Zudem wurde kein spezifisches Bewegungsmodell verwendet, so dass die Bewegung in allen Freiheitsgraden gleich behandelt und berücksichtigt wurde.

Wegen der offensichtlichen Probleme des GICP mit dieser Sequenz werden im Folgenden kurz die Ergebnisse einer günstigeren Aufnahmekonfiguration eingefügt.

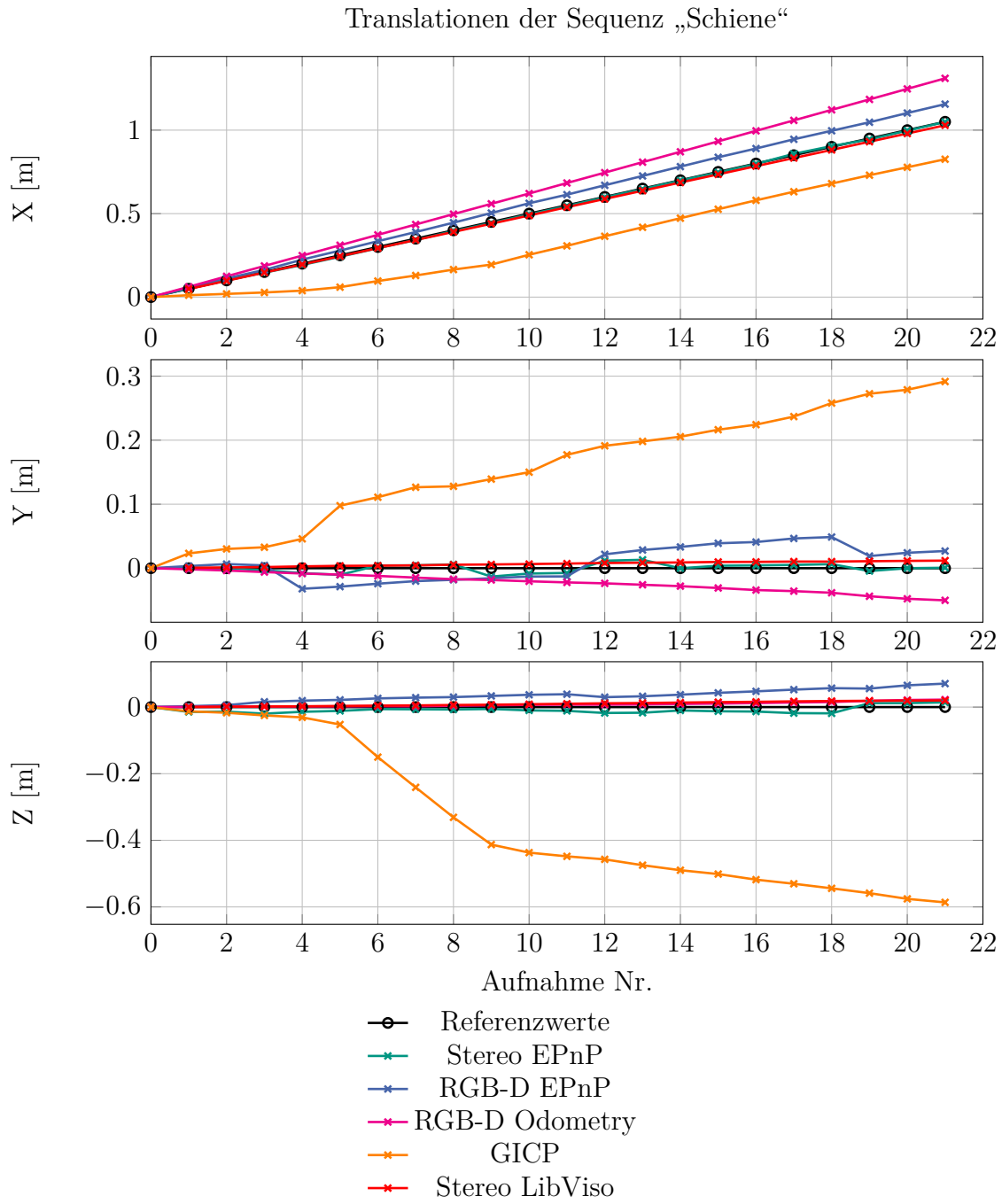


Abbildung 7.5: Akkumulierte Bewegung für die Sequenz „Schiene“: Translationen. Zur besseren Sichtbarkeit der Abweichungen sind die Achsen für X, Y, Z unterschiedlich skaliert

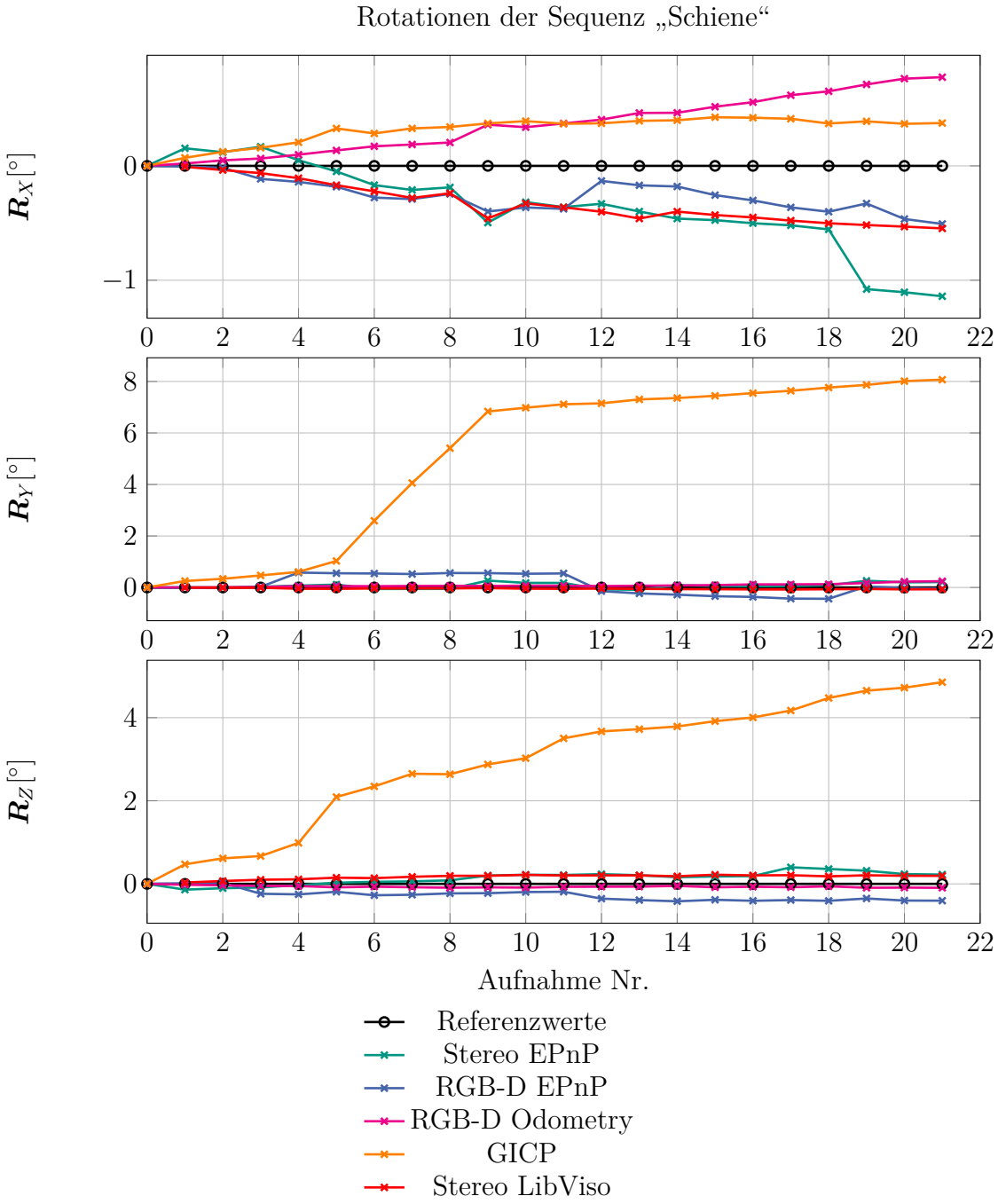


Abbildung 7.6: Akkumulierte Bewegung für die Sequenz „Schiene“: Rotationen. Zur besseren Sichtbarkeit der Abweichungen sind die Achsen für R_X, R_Y, R_Z unterschiedlich skaliert. Die Konvention der Winkel sind: R_Z = Gierwinkel (*yaw*), R_Y = Nickwinkel (*pitch*), R_X = Rollwinkel (*roll*).

7.2.2 Sequenz „Schiene Ecke“

Die Aufnahmekonfiguration für diese Sequenz ist so gewählt, dass mehrere zueinander konvergente Flächen vorhanden sind. Tischplatte und Wände bilden eine Ecke, auf die sich die Kamera in etwa zubewegt. Die Sequenz wurde mit der Kameragleitschiene erzeugt. Die Translationsschritte entlang der Schiene betragen 10 cm. Gerade beim GICP führt dies zu einem besseren Ergebnis, wie man Tabelle 7.4 entnehmen kann. Die Stereokamera-basierten Verfahren (Stereo EPnP und Stereo LibViso) haben im oberen Bereich der Szene Probleme ausreichend gute 2D-3D-Punktzuordnungen zu finden. In diesem Bereich befinden sich typische runde Zielmarken, die für die automatische Kodierung zusätzliche Kreissegmente besitzen.

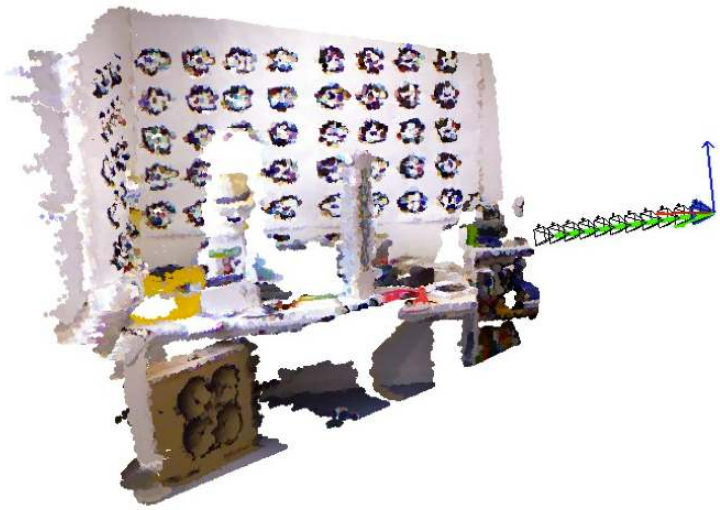


Abbildung 7.7: Aufnahmekonfiguration mit Gleitschiene der Sequenz „Schiene Ecke“

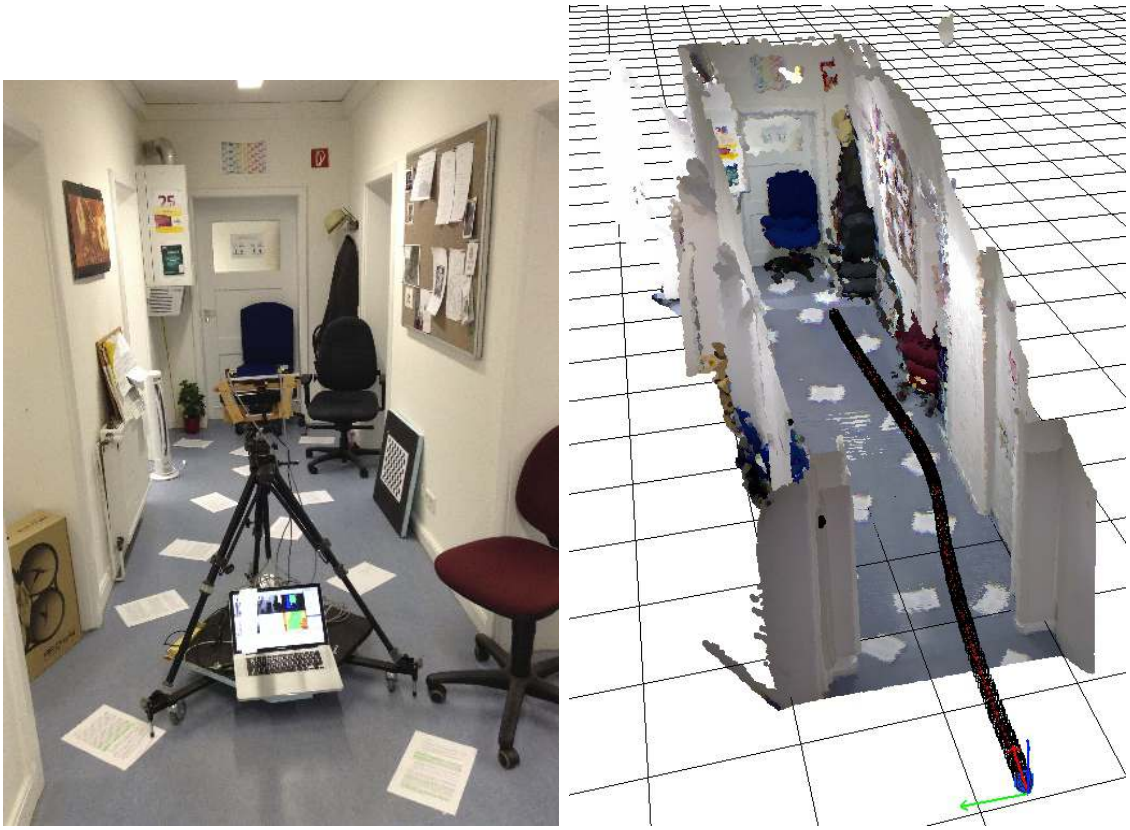
Methode	ATE	ATE KF	T-RPE	R-RPE	T-RPE KF	R-RPE KF
Stereo EPnP	0.101	0.049	0.162	1.258	0.103	1.257
RGB-D EPnP	**0.026**	0.034	**0.049**	0.496	0.059	0.248
RGB-D Odometry	0.049	**0.016**	0.069	**0.181**	**0.032**	**0.140**
GICP	0.048	0.018	0.070	0.683	0.036	0.499
Stereo LibViso	0.127	0.115	0.182	2.222	0.178	2.219

Tabelle 7.4: Vergleich der ATE der Sequenz „Schiene Ecke“

7.2.3 Sequenz „Stativwagen“

Diese Sequenz wurde in dem selben Gang wie die Schienensequenz durchgeführt. Das Aufnahmesystem befindet sich hier auf einem Stativwagen (siehe Abb. 7.8a, S. 106), der durch den Gang geschoben wird. Diese Bewegung hat im Prinzip drei Freiheitsgrade: X, Y und Gierwinkel. Durch Unebenheiten des Ganges wie z. B. eine Bodenwelle in der Mitte ergeben sich auch kleine Bewegungen in den anderen Parametern.

Die Verfügbarkeit natürlicher Merkmale, die für das Tracking benötigt werden, ist auch hier wieder nicht optimal. Aus diesem Grund wurden einige Gegenstände am Rande der Strecke platziert. Der Fußboden wurde zusätzlich mit bedruckten Zetteln ausgelegt (siehe Abb. 7.8a, S. 106). Es gibt aber dennoch untexturierte Bereiche, wie man in den drei Aufnahmen Nr. 1, 100 und 180 der linken Stereokamera in Abb. 7.9 erkennen kann.



(a) Stativwagen

(b) Trajektorie aus Apéro. Mehrere Kinect Punktwolken wurden mit Hilfe der Trajektorie zusammengefügt

Abbildung 7.8: Aufnahmekonfiguration für die Fahrt mit dem Stativwagen

Abbildung 7.9: Aufnahmen (1, 100, 180) aus der Sequenz „Stativwagen“. Obere Reihe: linke Stereokamera; untere Reihe: Kinect RGB Kamera

Die Referenzwerte für diesen Test wurden mit Bündelblockausgleichung der Kinect RGB-Bildsequenz mit Apéro berechnet. Natürliche Eckmerkmale, deren 3D-Koordinaten leicht messbar waren, wurden in den Aufnahmen am Anfang und am Ende der Sequenz gemessen und als Passpunkte eingeführt. Die resultierende Trajektorie ist in Abb. 7.8 zu sehen. Posen innerhalb der Trajektorie wurden genutzt um die Kinect-Punktwolken zur Darstellung der Szene zusammenzufügen. Die Sequenz besteht aus 180 Aufnahmen.

Methode	ATE	ATE KF	T-RPE	R-RPE	T-RPE KF	R-RPE KF
Stereo EPnP	0.459	0.786	4.022	7.596	3.578	5.591
RGB-D EPnP	0.137	0.169	4.564	10.146	4.376	8.455
RGB-D Odometry	0.367	0.246	4.977	**7.563**	4.832	**5.528**
GICP	1.355	1.285	**2.855**	8.622	**2.813**	6.176
Stereo LibViso	**0.136**	**0.136**	4.351	10.185	4.351	10.191

Tabelle 7.5: Vergleich ATE, RPE (Sequenz „Stativwagen“)

Für die Berechnung des ATE werden Trajektorie und Referenz vor dem Berechnen der Abstände zueinander ausgerichtet. Beim Vergleich der Translationen sind in Abb. 7.12 diese transformierten Trajektorien dargestellt.

Das Verfahren Stereo LibViso hat mit einem ATE von 0,136 m das beste Ergebnis.

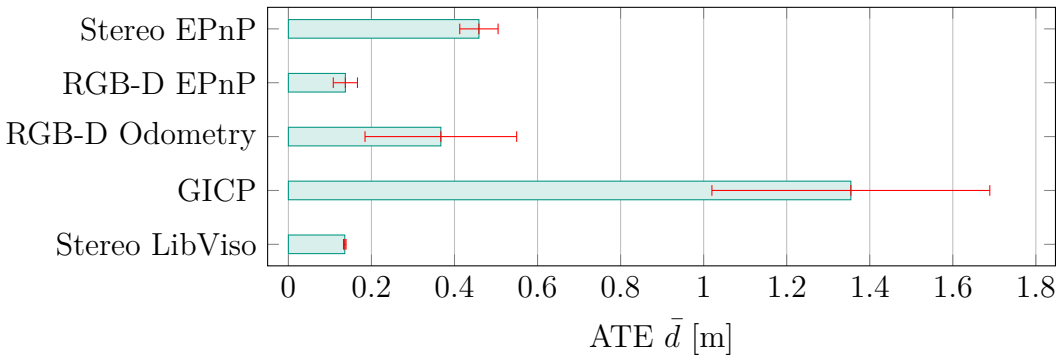


Abbildung 7.10: Vergleich der ATE (Sequenz „Stativwagen“)

Bei GICP zeigt die akkumulierte Trajektorie in Y- und Z-Achse deutliche Abweichungen zur Referenz (Abb. 7.12 „Translationen“). Der Plot der X-Werte zeigt, dass die ursprüngliche Trajektorie zu kurz ist.

Bei den eigenen Verfahren erzielt das RGB-D EPnP das deutlich bessere Ergebnis im Vergleich zu Stereo EPnP. Ähnlich wie bei der Sequenz „Schiene“ werden mit dem Kinect-Ansatz oft mehr robuste 2D-3D-Punktkorrespondenzen für die Bewegungsberechnung gefunden als mit dem Stereomatching. Die jeweils ermittelte Anzahl der Inlier aus der EPnP-Ransac-Schätzung, die für beide Zeitpunkte gültig sind, ist in Abb. 7.11 für Stereo EPnP und RGB-D EPnP pro Aufnahme gegenübergestellt. Beim Stereoverfahren wird die minimal benötigte Anzahl von 2D-3D-Punktkorrespondenzen fünfmal unterschritten. Bei RGB-D EPnP geschieht dies in dieser Sequenz kein einziges mal.

Ungünstige Verteilungen der Bildpunkte einer Aufnahme sowie eine zu geringe Anzahl von 2D-3D-Punktkorrespondenzen führen zu falschen Bewegungen, die den weiteren Verlauf der Trajektorie negativ beeinflussen. Durch das Fehlen von Schätzungen der absoluten Posen durch andere Sensoren bzw. Verfahren können diese Fehler nicht mehr korrigiert werden.

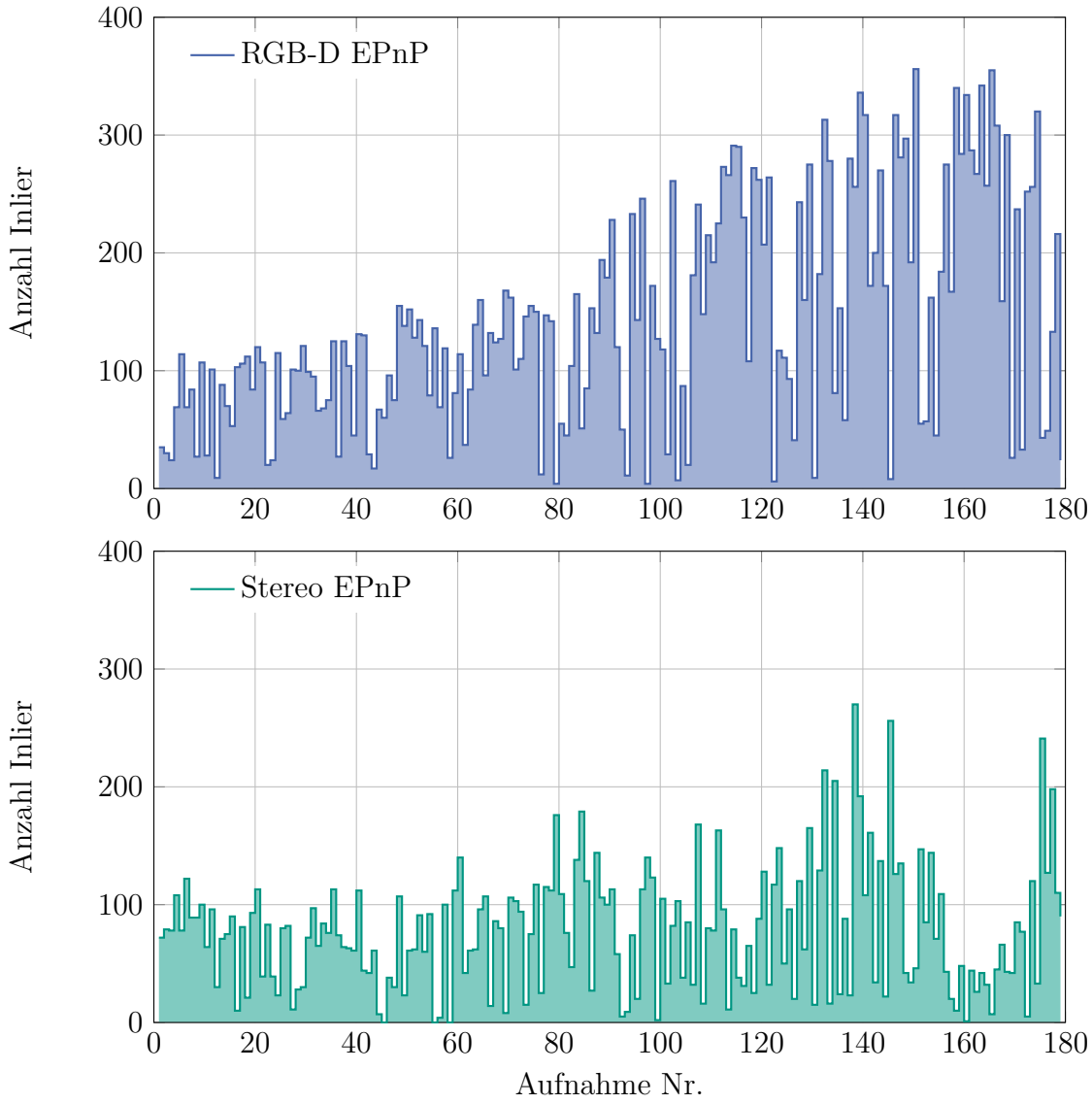


Abbildung 7.11: Anzahl der EPnP-Ransac-Inlier für die Verfahren RGB-D EPnP und Stereo EPnP (Sequenz „Stativwagen“). Beim Verfahren Stereo EPnP stehen weniger Merkmale für die Bewegungsberechnung zur Verfügung. Die minimale Anzahl von vier Inliern wird hier fünfmal unterschritten

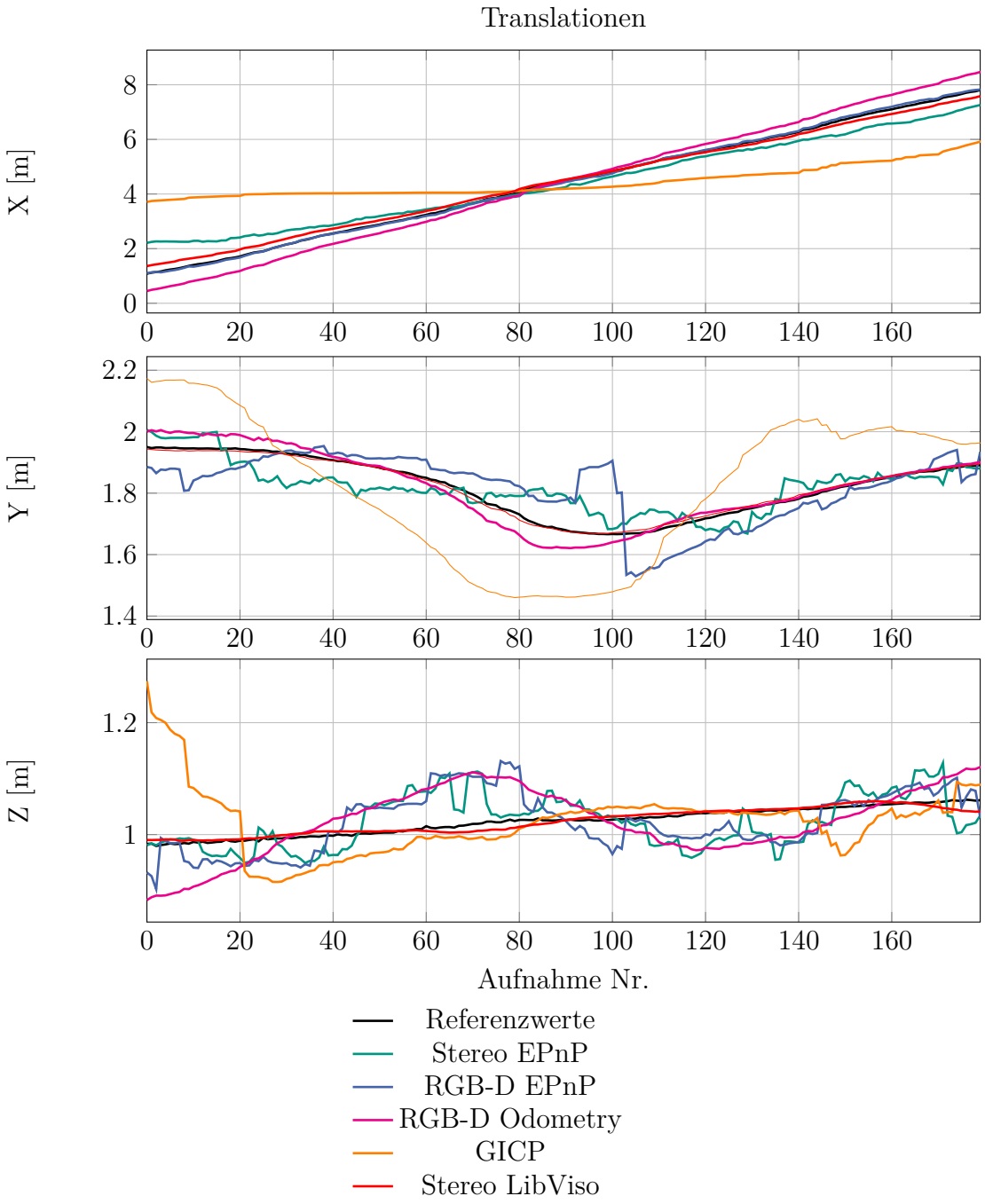


Abbildung 7.12: Akkumulierte Translationen für Sequenz „Stativwagen“

7.3 Laufzeiten der Verfahren

Abschließend wird die Laufzeit der Verfahren gegenübergestellt. Abb. 7.13 stellt die ermittelten Zeiten, die für eine Bewegungsberechnung zwischen zwei Aufnahmezeitpunkten auf einem Desktop Computer mit einem Intel „Core 2 Duo“ Prozessor mit 2,53 GHz Taktfrequenz benötigt wurden grafisch dar. Auf der unteren Achse ist die Zeit in [s] und auf der oberen Achse ist die entsprechende Bildwiederholrate in [FPS] angegeben. Das schnellste Verfahren ist RGB-D EPnP mit einer mittle-

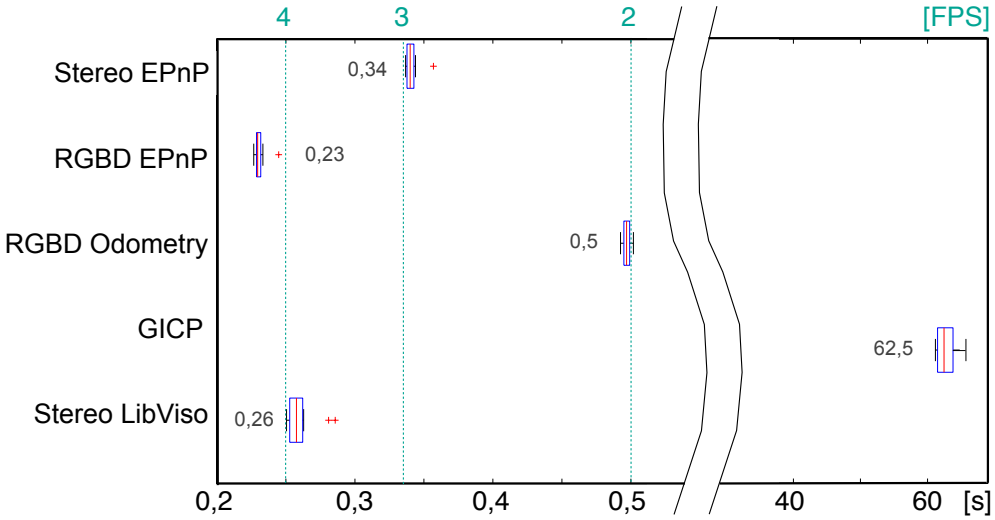


Abbildung 7.13: Laufzeiten der Verfahren pro Bewegungsberechnung zwischen zwei Aufnahmezeitpunkten

ren Berechnungszeit von 0,23 s. Das robusteste und gleichzeitig auch komplexeste Verfahren in dieser Untersuchung, Stereo LibViso, erreicht eine mittlere Berechnungszeit von 0,26 s. Stereo LibViso hat keine weiteren Abhängigkeiten zu externen Bibliotheken. Ein Großteil des Codes ist mit Intel SSE (Streaming SIMD Extensions) Anweisungen optimiert. Das Verfahren liest allerdings direkt Epipolarbilder ein. Die Zeit, die zur Erzeugung aus den Kameraaufnahmen hierfür benötigt wird, fehlt in dieser Untersuchung. Das eigene Verfahren (Stereo EPnP), das u.a. mit Hilfe der OpenCV-Bibliothek implementiert ist, lädt die Originalaufnahmen und erzeugt daraus Epipolarbilder zur Laufzeit.

Das GICP-Verfahren benötigt mit den Kinect-Punktwolken, die im Schnitt 300000 Punkte enthalten, deutlich länger als alle anderen Verfahren. Es ist mit durchschnittlich 62,5 s Berechnungszeit weit von einer Echtzeitfähigkeit entfernt. Für diesen Test wurden die Tiefenbilder zuvor in Punktwolken umgewandelt und diese direkt als fertige XYZ-Dateien geladen. Dieser Arbeitsschritt müsste noch hinzugezählt werden. Mit einem optimalen Wert für $d_{\max}$ konnte die Laufzeit experimentell auf etwa 45 s reduziert werden.

Grundsätzlich sind die Verfahren, die mit wenig Merkmalen arbeiten schneller als jene Verfahren, die mit „dichten“ Merkmalen arbeiten. Die eigenen, EPnP-basierten Verfahren könnten durch die Nutzung von GPU-gestützten Berechnungen für den Lucas-Kanade Tracker oder beim Verfahren Stereo EPnPfür das Stereomatching und die Epipolarbildberechnung noch weiter beschleunigt werden.

8 Zusammenfassung und Ausblick

Zielsetzung der Arbeit war die Untersuchung von Verfahren zur Egomotion-Bestimmung von Kamerasystemen. Als Anwendungsrahmen für die Fragestellung diente die Erweiterte Realität. Es wurde erläutert, das AR-Anwendungen sowohl eine genaue Positionierung und Orientierung der Kamera als auch detaillierte Informationen über die direkte Umgebung in 3D benötigen. Stereokamerasysteme und Tiefenbildkameras sind in der Lage die Umgebung direkt in 3D zu erfassen. Aus diesem Grund bilden diese beiden Kamerasysteme den Schwerpunkt bei der Frage nach geeigneten Verfahren zur Eigenbewegungsschätzung.

Die Betrachtungen in dieser Arbeit beziehen sich auf Verfahren aus dem Bereich Visual Odometry. Die VO beschränkt sich rein auf die Schätzung der Eigenbewegung zwischen zwei Aufnahmezeitpunkten. Bei SLAM-Verfahren wird gleichzeitig eine Karte der unbekannten Umgebung erfasst. Dabei steht die 3D-Information der Umgebung nicht fortlaufend in Echtzeit zur Verfügung. Gerade dies ist aber für die Nutzerinteraktion in AR-Anwendungen wichtig. Auf diese Weise können Verdeckungen virtueller Inhalte durch die Hände des Nutzers oder durch Gegenstände die Illusion einer räumlichen Überlagerung ermöglichen.

Aus möglichen Optionen für die Bewegungsschätzung mit Bildmerkmalen wurden zwei Konfigurationen hervorgehoben: die Berechnung der Bewegung zwischen zwei Zeitpunkten auf Basis von 3D-Punktkorrespondenzen und auf Basis von 2D-3D-Punktkorrespondenzen. Für die 3D-Punktkorrespondenzen wurde das Generalized ICP-Verfahren von Segal [Segal u. a. 2009] vorgestellt, das gut mit 2,5D-Punktwolken zurechtkommt und für 2D-3D-Punktkorrespondenzen das Efficient-PnP-Verfahren von Moreno-Noguer [Moreno-Noguer u. a. 2007], das in Verbindung mit RANSAC eine hohe Anzahl von Ausreißern verkraftet. Wegen des größeren Fehlers der 3D-Punkte in Richtung der Kameratiefe ist das EPnP-Verfahren zuverlässiger bei Rotationen.

Die Aufnahmesysteme Stereokamera und Kinect Tiefenbildkamera sind in ihren wichtigsten Eigenschaften beschrieben worden. Beide Systeme arbeiten nach einem

Stereoverfahren. Die Kinect Kamera liefert zuverlässiger dichte 3D-Informationen als es bei Stereoverfahren der Fall ist, da ein aktives Kamerasystem mit Musterprojektion nicht auf ausreichend texturierte Oberflächen angewiesen ist. Für einen Betrieb auf einem mobilen Endgerät würde die dichte Tiefenbildschätzung mit einem Stereokamerasystem zu viel Rechenleistung beanspruchen. Allerdings kann das dichte Stereomatching durch Hardware-Lösungen, wie FPGAs in Echtzeit betrieben werden [Howard u. a. 2012]. Die Kinect Tiefenbildkamera wiederum kann nicht im Freien betrieben werden. Als Erweiterung einer Spielkonsole ist diese dafür ursprünglich nicht entwickelt worden. Die Qualität und einfache Handhabung der 3D-Punktwolke in Verbindung mit dem geringen Anschaffungspreis machen Hoffnung auf zukünftige Sensoren dieser Art, wie dies am Beispiel des „Structure Sensors“ [*Occipital Structure Sensor* 2014] zu sehen ist. Dabei handelt es sich um einen Tiefensensor nach dem „Kinect-Prinzip“, der für den mobilen Einsatz konzipiert wurde.

Aus den möglichen Verfahren für die automatische Merkmalsextraktion- und Verfolgung in Bildsequenzen wurde der Kanade-Lucas-Tracker gewählt. Das Verfolgen von Eckmerkmalen mit dem KLT passt mit seinen Bedingungen für die Bewegungen im Bild gut zu den zeitlich konstanten Bildfolgen der beiden Aufnahmesysteme.

Eckpunkte sind allerdings kritisch bei der Verwendung mit Tiefenbildkameras, da Ecken oft auch durch eine Staffelung von Objekten in der Tiefe zustande kommen. Gerade in diesen Bereichen können Punkte in der Tiefe springen oder durch Verdeckung bzw. fehlende Tiefenwerte beim Tracking wegfallen.

Eine Analyse vorhandener Stereo-Egomotion Verfahren unterschiedlicher Plattformen und Anwendungen für Stereokamerasysteme und Tiefenkamerasysteme hat gezeigt, dass es bei beiden Sensorarten punktbasierte und dichte Verfahren gibt, wobei die dichten Verfahren bei den Tiefensensoren überwiegen.

Die aufwendigsten Verfahren findet man im Bereich der autonomen Fahrzeuge und Fahrerassistenzsysteme. Diese müssen die Umgebung dynamisch betrachten, da auf die Bewegungen der verschiedenen Verkehrsteilnehmer in Echtzeit reagiert werden muss. Dieser Szenenfluss ist das stärkste Konzept unter den Verfahren der VO. Mobilität bedeutet in diesem Anwendungsfeld aber auch, die dafür notwendige Rechenleistung im Kofferraum des Fahrzeuges mitführen zu können. Für den Betrieb auf typischen mobilen Endgeräten, die ein Fußgänger mitführen kann, ist dieses Verfahren zu rechenintensiv.

Wegen der gemeinsamen Eigenschaften von Stereokamera und Tiefenbildkamera wurde ein neuer Ansatz entwickelt, der auf Basis des EPnP die Bewegung mit beiden Sensorarten gleichermaßen schätzen kann. Somit steht ein Ansatz zur Verfügung, der auf der gleichen methodischen Grundlage beruht.

Aus den analysierten, vorhandenen Lösungen wurden drei Verfahren mit einer verfügbaren Implementierung gewählt. Es wurde eine Methodik entwickelt, diese an gemeinsamen Stereo- und Kinect-Sequenzen zu testen und diese mit Referenzwerten zu vergleichen. Das hierfür zusammengestellte Aufnahmesystem kann die Bilder beider Kamerasysteme, das Stereobildpaar sowie Farb- und Tiefenbild der

Kinect Kamera, gemeinsam mit 12 Hz speichern. Mit Hilfe eine Kameragleitschiene und einer photogrammetrischen Software zur Bündelblockausgleichung wurden Referenzwerte der Bewegungen erstellt. Für die Bewertung der Kameratrajektorien wurde der *Absolute Translation Error* als Maß für den akkumulierten Fehler und der *Relative Pose Error* als Maß für die Drift berechnet.

Die Verfahren zeigen mit zunehmender Dauer der Bildsequenzen alle eine Drift. Für den ATE zeigt sich der Szenenfluss als das stärkste Konzept; auch im Falle einer längeren Kamerafahrt. Bei den relativen Fehlermaßen zeigen die Verfahren unterschiedliche Stärken und Schwächen. Die beiden eigenen EPnP-basierten Verfahren liefern dabei nahezu in allen Fehlermaßen gute bis sehr gute Ergebnisse. Es ist aber nicht notwendig die VO-Verfahren isoliert zu betrachten. Vielmehr sollen sie in Kombination mit weiteren Verfahren und Sensoren zu einem möglichst robusten Gesamtsystem führen indem alle vorhandenen Navigationslösungen eines Systems in einer Kalman Filter Schleife integriert werden und auf diese Weise z. B. die Zeit zwischen zwei absoluten Positionsangaben eines GNSS überbrücken zu können.

Vielversprechend sind auch modellbasierte Trackinglösungen, die ein VO-System in regelmäßigen Abständen mit absoluten Positionsangaben stützen können und so in Zukunft ein rein kamerabasiertes Trackingsystem ohne Drift ermöglichen können [Urban u. a. 2013].

Ausblick

Die Kinect Kamera ist mit 28 cm Breite, ihrem Gewicht und Stromverbrauch nicht für den mobilen Betrieb ausgelegt. Der mittlerweile erhältliche Nachfolger (Kinect 2) hat eine bessere Tiefenbildauflösung und eine andere zugrunde liegende Technologie. Größe, Stromverbrauch und Gewicht sprechen auch hier gegen eine mobile Anwendung im Rahmen von Augmented Reality. Zudem können die Sensordaten nur über die offizielle API des Herstellers ausgelesen werden, da der Datenstrom verschlüsselt ist. Eine ähnliche Verbreitung und Fülle unterschiedlichster Projekte und Forschungsarbeiten, wie man es bei der ersten Kinect-Version beobachten kann, wird es hier wohl nicht geben.

Für die Zukunft sind jedoch mobile Geräte mit integriertem Tiefensensor angekündigt:

- Das Google Projekt „Tango“[1] beschäftigt sich mit mobilen Endgeräten (Smartphones und Tablets), die neben den üblichen Sensoren für Positionierung und Orientierung eine Fisheyecamera und einen nicht weiter spezifizierten Tiefenbildsensor integriert haben. Entwickler können sich für den Zugang zum Hardwareprototypen bewerben. Für die Allgemeinheit sind diese Geräte noch nicht zugänglich.

[1] `https://www.google.com/atap/projecttango`, besucht im März 2015

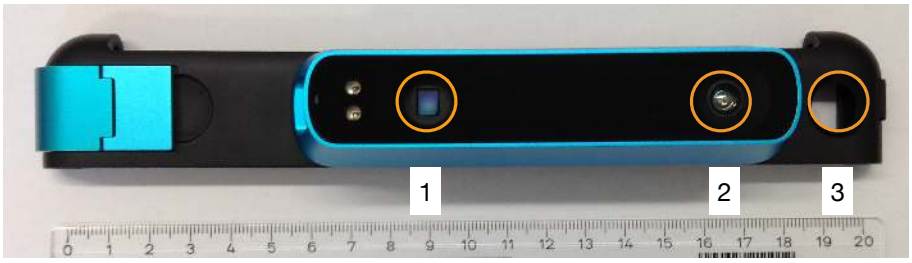


Abbildung 8.1: Structure Sensor: Tiefensensor für mobile Geräte. 1: Musterprojektor, 2: IR-Kamera, 3: Loch in Tablethalterung für integrierte RGB Kamera

- Microsoft hat eine Augmented-Reality-Brille „HoloLens“[2] mit integriertem Tiefensensor angekündigt, der einen Öffnungswinkel von 120×120 Grad abdeckt. Diese Konzeptstudie ist noch nicht als Produkt verfügbar.

Für die Zwischenzeit ist der oben erwähnte Structure Sensor als Zusatz für Tablets oder Smartphones bereits erhältlich. Die Sensortechnologie stammt von PrimeSense, den Entwicklern der Tiefenbildkamera in der ersten Version der Kinect. Detaillierte Informationen hierzu verschweigt der Hersteller (Occipital). Auf jeden Fall hat dieser Tiefensensor nahezu ähnliche technische Spezifikationen wie die Kinect Kamera in der ersten Version, eine eigene Stromversorgung für bis zu vier Stunden Betrieb und ein Gewicht von 99 g (siehe Abb. 8.1, S. 116). Der Sensor hat keine eigene Farbkamera integriert. Es wird die Tabletkamera genutzt.

Bei allen angekündigten und echten Produkten ist die zusätzliche Sensorik durch den Wunsch motiviert, die direkte Umgebung des Nutzers erfassen zu können. Die Gründe hierfür sind:

- das einfache Erzeugen texturierter 3D-Modelle von Objekten aller Art um diese bspw. auf einem 3D-Drucker ausgeben zu können
- Interaktion mit der Umgebung (auch Tracking der Hände) und Verdeckungsberechnung für Augmented Reality Anwendungen

Sowohl die 3D-Objekterfassung als auch AR-Anwendungen benötigen robuste Verfahren zur Berechnung der Eigenbewegungen des Kamerasystems. Auf der Consumer Electronics Show (CES) 2015 wurde für den Structure Sensor ein Trackingverfahren für sechs Freiheitsgrade angekündigt. Da alle Funktionen des Tiefensensors in einer API für mobile Endgeräte zur Verfügung stehen, muss auch das angekündigte Verfahren auf leistungsschwacher Hardware in einer vernünftigen Geschwindigkeit laufen.

Die aktuellen Entwicklungen bestätigen die Bedeutung kamerabasierter Ansätze. Auch wenn der Structure Sensor eine gute Möglichkeit bietet, aktuell mit einem

[2] Janssen und Kulmann (2015). *Blicken statt klicken*, c't 5, Heise Verlag, S. 59–60

mobilen Tiefensensor zu arbeiten, wird erst die Integration solcher Sensoren in zukünftige mobile Endgeräte für den Massenmarkt den Bedarf an robusten Egomotion-Verfahren auf leistungsschwacher Hardware vorantreiben. Auf jeden Fall müssen die Verfahren auf mehrere CPUs und GPUs aufgeteilt werden können. Für solche Hardwareplattformen sind merkmalsbasierte Ansätze wegen des geringeren Rechenaufwandes besser geeignet.

In Tiefenbildern liegen markante Merkmale an Kanten, die durch unterschiedliche Tiefen im Raum entstehen. Gerade an diesen Kanten kann es, wie man am Beispiel der Kinect beobachten kann, durch Verdeckungen und Rauschen zu instabilen Merkmalen führen, die beim Tracking in einer Tiefenbildsequenz schnell verloren gehen. Eine ungünstige lokale Nachbarschaft in den Tiefenwerten wurde in den eigenen Ansätzen stets verworfen. Merkmalsextraktoren und Deskriptoren wie z. B. NARF (*Normal Aligned Radial Feature*) [Steder u. a. 2010] beachten diese Probleme und sind speziell für die Arbeit mit Tiefenbildern konzipiert. Kombinierte Verfahren zur gemeinsamen Merkmalsextraktion und -Verfolgung in Intensitätsbild und Tiefenbild könnten hier die Robustheit in Zukunft steigern.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
AR	Augmented Reality
ARS	AR-System, die Gesamtheit aller für AR notwendigen Komponenten
EPnP	Enhanced Perspective-n-Point Problem
FPGA	Field Programmable Gate Array
FPS	Frames per Second, Bildwiederholungsrate in [Hz]
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
ICP	Iterative Closest Points
IMU	Inertial Measurement Unit
KLT	Kanade-Lucas Feature Tracker, oft auch LKT
RGB-D	Kombination aus Farbbild und Tiefenbild einer Kinect Kamera
SAD	Sum of Absolute Differences
SLAM	Simultanious Location and Mapping
ToF-Kamera	Time-of-Flight-Kamera
UAV	Unmanned/Unpiloted Arial Vehicle

VO	Visual Odometry

Literaturverzeichnis

Adams, Jim, Ken Parulski und Kevin Spaulding (1998). "Color Processing in Digital Cameras". In: *IEEE Micro* 18.6, S. 20–30.

Arth, Clemens, Daniel Wagner, Manfred Klopschitz, Arnold Irschara und Dieter Schmalstieg (2009). "Wide Area Localization on Mobile Phones". In: *IEEE International Symposium on Mixed and Augmented Reality - Arts, Media and Humanities. ISMAR-AMH*, S. 73–82.

Arun, K. S., Thomas S. Huang und Steven D. Blostein (1987). "Least-squares Fitting of Two 3-D Point Sets". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence* 9.5, S. 698–700.

Azuma, Ronald (1997). "A Survey of Augmented Reality". In: *Presence: Teleoperators and Virtual Environments 6*, S. 355–385.

Badino, Hernan (2004). "A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform". In: *IWCM 2004: First Internation Workshop on Complex Motion*. Günzgburg, Germany.

Badino, Hernán und Takeo Kanade (2011). "A Head-Wearable Short-Baseline Stereo System for the Simultaneous Estimation of Structure and Motion". In: *IAPR Conference on Machine Vision Applications*. Nara, Japan, S. 1–5.

Baker, Simon, Daniel Scharstein, J.P. Lewis, Stefan Roth, Michael J. Black und Richard Szeliski (2011). "A Database and Evaluation Methodology for Optical Flow". In: *International Journal of Computer Vision* 92.1, S. 1–31.

Bay, Herbert, Tinne Tuytelaars und Luc Van Gool (2006). "SURF: Speeded Up Robust Features". In: *Proceedings of the ninth European Conference on Computer Vision*.

Bayer, Bryce E. (1976). *Color imaging array*. US Patent 3971065.

Behringer, Reinhold, Jun Park und Venkataraman Sundareswaran (2002). "Model-Based Visual Tracking for Outdoor Augmented Reality Applications". In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. Darmstadt, S. 277–278.

Benko, Hrvoje, Ricardo Jota und Andrew D. Wilson (2012). "MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. New York, NY, USA: ACM, S. 199–208.

Besl, Paul J. und Neil D. McKay (1992). "A Method for Registration of 3-D Shapes". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence* 14.2, S. 239–256.

Bimber, Oliver und Ramesh Raskar (2005). *Spatial Augmented Reality - Merging Real and Virtual Worlds*. A K Peters.

Blostein, Steven S. und Thomas S. Huang (1987). "Error Analysis in Stereo Determination of 3-D Point Positions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.6, S. 752–765.

Bouguet, Jean-Yves (2000). *Pyramidal Implementation of the Lucas Kanade Feature Tracker*.

— (2014). *Camera Calibration Toolbox for Matlab*. `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`, besucht im März 2014.

Brown, D. C. (1971). "Close-Range Camera Calibration". In: *Photogrammetric Engineering* 37.8, S. 855–866.

Burrus, Nicolas (2014). *Kinect RGB Demo*. `http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV4?from=Research.KinectRgbDemoV2`, besucht im März 2014.

Caudell, Thomas P. und David W. Mizell (1992). "Augmented Reality: an Application of Heads-up Display Technology Tomanual Manufacturing Processes". In: *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. Bd. 2. Kauai, Hawaii, USA, S. 659–669.

Ceriani, Simone, Giulio Fontana, Alessandro Giusti, Daniele Marzorati, Matteo Matteucci, Davide Migliorea, Davide Rizzi, Domenico G. Sorrenti und Pierluigi Taddei (2009). "Rawseeds Ground Truth Collection Systems for Indoor Self-localization and Mapping". In: *Autonomous Robots* 27.4, S. 353–371.

Chen, Ting (1999). *A Study of Spatial Color Interpolation Algorithms for Single-Detector Digital Cameras*. Techn. Ber. Information System Laboratory, Department of Electrical Engineering, Standford University.

Comport, Andrew I., Ezio Malis und P. Rives (2007). "Accurate Quadrifocal Tracking for Robust 3D Visual Odometry". In: *IEEE International Conference on Robotics and Automation*. Rom, S. 40–45.

Dahlkamp, Hendrik, Hans-Hellmut Nagel, Artur Ottlik und Paul Reuter (2007). "A Framework for Model-Based Tracking Experiments in Image Sequences". In: *International Journal of Computer Vision* 73.2, S. 139–157.

Daniel, Wagner, Reitmayr Gerhard, Mulloni Alessandro, Tom Drummond und Schmalstieg Dieter (2010). "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones". In: *IEEE Transactions on Visualization and Computer Graphics* 16.3, S. 355–368.

David, Philip, Daniel Dementhon, Ramani Duraiswami und Hanan Samet (2003). "Simultaneous Pose and Correspondence Determination using Line Features". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Madison, USA, S. 424–431.

— (2004). "SoftPOSIT: Simultaneous Pose and Correspondence Determination". In: *International Journal of Computer Vision* 59.3, S. 259–284.

Dominguez Quijada, Salvador, Eduardo Zalama, Jaime Gomez Garcia-Bermejo, Rainer Worst und Sven Behnke (2013). "Fast 6D Odometry Based on Visual Features and Depth". In: *Intelligent Autonomous Systems 12*. Hrsg. von Sukhan Lee, Hyungsuck Cho, Kwang-Joon Yoon und Jangmyung Lee. Bd. 193. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, S. 245–256.

Douxchamps, Damien (2014). *1394-based DC Control Library*. `http://damien.douxchamps.net/ieee1394/libdc1394/`, besucht im März 2014.

Drummond, Tom und Roberto Cipolla (2002). "Real-time Visual Tracking of Complex Structures". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence* 24.7, S. 932–946.

Eggert, D. W., A. Lorusso und R. B. Fisher (1997). "Estimating 3-D rigid body transformations: a comparison of four major algorithms". In: *Machine Vision and Applications* 9.5-6, S. 272–290.

Faugeras, Olivier und Quang-Tuan Luong (2001). *The Geometry of Multiple Images*. MIT Press.

Feiner, Steven, Blair MacIntyre und Tobias Höllerer (1997). "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment". In: *International Sympopium on Wearable Computing*. Cambridge, S. 74–81.

Fiedler, David und Heinrich Müller (2012). "Impact of Thermal and Environmental Conditions on the Kinect Sensor". In: *International Workshop on Depth Image Analysis at the 21st International Conference on Pattern Recognition*.

Fischler, Martin A. und Robert C. Bolles (1981). "Random Sample Consensus: a Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24.6, S. 381–395.

Förstner, Wolfgang (2000). "Moderne Orientierungsverfahren". In: *Photogrammetrie - Fernerkundung - Geoinformation* 3, S. 163–176.

— (2004a). *Projective Geometry for Photogrammetric Orientation Procedures I*. `http://www.ipb.uni-bonn.de/fileadmin/publication/pdf/Forstner2004Projective.pdf`, besucht im März 2015.

— (2004b). *Projective Geometry for Photogrammetric Orientation Procedures II*. `http://www.ipb.uni-bonn.de/fileadmin/publication/pdf/Forstner2004Projectivea.pdf`, besucht im März 2015.

Fossati, Andrea, Juergen Gall, Helmut Grabner, Xiaofeng Ren und Kurt Konolige, Hrsg. (2013). *Consumer Depth Cameras for Computer Vision*. Advances in Computer Vision and Pattern Recognition. Springer.

Franke, Uwe, Stefan Gehrig, Hernan Badino und Clemens Rabe (2008). "Towards Optimal Stereo Analysis of Image Sequences". In: *Second International Workshop Robot Vision (RobVis)*. Bd. 4931/2008. LNCS. Auckland: Springer Berlin/Heidelberg, S. 43–58.

Franke, Uwe, David Pfeiffer, Clemens Rabe, Carsten Knoeppel, Markus Enzweiler, Fridtjof Stein und Ralf G. Herrtwich (2013). "Making Bertha See". In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*, S. 214–221.

Franke, Uwe, Clemens Rabe, Hernán Badino und Stefan Gehrig (2005). "6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception". In: *Pattern Recognition, 27th DAGM Symposium*. Hrsg. von Walter Kropatsch, Robert Sablatnig und Allan Hanbury. Bd. 3663. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, S. 216–223.

Freedmann, Barak, Alexander Shpunt und Yoel Arieli (2010). *Depth Mapping Using Projected Patterns*. US Patent 2010/0118123 A1.

Fried, Limor (2010). *diykinect - Detaillierte Beschreibung des Kinect Reverseengineering über die USB-Schnittstelle*. `http://www.ladyada.net/wiki/tutorials/learn/diykinect/index.html?s[]=kinect`, besucht im Februar 2011.

Gao, Xiao-Shan, Xiao-Rong Hou, Jianliang Tang und Hang-Fei Cheng (2003). "Complete Solution Classification for the Perspective-three-point Problem". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence* 25.8, S. 930–943.

Geiger, Andreas (2013). "Probabilistic Models for 3D Urban Scene Understanding from Movable Platforms". Diss. Department of Measurement und Control Systems, Karlsruhe Institute of Technology (KIT).

— (2014). *LibViso2: C++ Library for Visual Odometry*. `http://www.cvlibs.net/software/libviso`, besucht im April 2014.

Geiger, Andreas, Philip Lenz und Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 3354–3361.

Geiger, Andreas, J. Ziegler und Christoph Stiller (2011). "StereoScan: Dense 3d Reconstruction in Real-time". In: *IEEE Intelligent Vehicles Symposium (IV)*, S. 963–968.

Georgel, Pierre, Pierre Schroeder, Selim Benhimane und Stefan Hinterstoissen (2007). "An Industrial Augmented Reality Solution For Discrepancy Check". In: *The Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, S. 111–115.

Goedemé, Toon, Marnix Nuttin, Tinne Tuytelaars und Luc Van Gool (2007). "Omnidirectional Vision Based Topological Navigation". In: *International Journal of Computer Vision* 74.3, S. 219–236.

Gonzalez-Jorge, H., B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez und P. Arias (2013). "Metrological evaluation of Microsoft Kinect and Asus Xtion sensors". In: *Measurement* 46.6, S. 1800–1806.

Gunturk, Bahadir K., Yucel Altunbasak und Russell M. Mersereau (2002). "Color Plane Interpolation Using Alternating Projections". In: *IEEE Transactions on Image Processing* 11.9, S. 997–1013.

Gunturk, Bahadir K., John Glotzbach, Yucel Altunbasak, Ronald W. Schafer und Russel M. Mersereau (2005). "Demosaicking: Color Filter Array Interpolation". In: *IEEE Signal Processing* 22.1, S. 44–54.

Haralick, Bert M., Chung-Nan Lee, Karsten Ottenberg und Michael Nölle (1994). "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem". In: *International Journal of Computer Vision* 13.3, S. 331–356.

Harris, Chris und Mike Stephens (1988). "A Combined Corner and Edge Detector". In: *Proceedings of the 4th Alvey Vision Conferences*, S. 147–151.

Hartley, Richard I. (1997). "In Defense of the Eight-Point Algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Vision*. IEEE, S. 580–593.

Hartley, Richard und Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.

Heikkilä, Janne und Olli Silvén (1997). "A Four-step Camera Calibration Procedure with Implicit Image Correction". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. San Juan, Puerto Rico, S. 1106–1112.

Hinz, Stefan, Dominik Lenhart und Jens Leitloff (2007). "Detection and Tracking of Vehicles in low Framerate Aerial Image Sequences". In: *Proc. Workshop on High-Resolution Earth Imaging for Geo-Spatial Information*. Citeseer. Hannover, Germany, S. 1–6.

Holloway, Richard L. (1995). *Registration Error Analysis for Augmented Reality*. Techn. Ber. TR95-001. Chapel Hill, NC: Department of Computer Science University of North Carolina.

Holz, Dirk, Stefan Holzer, Radu Bogdan Rusu und Sven Behnke (2012). "Real-time Plane Segmentation Using RGB-D Cameras". In: *RoboCup 2011: Robot Soccer World Cup XV*. Springer, S. 306–317.

Horn, Berthold K. P. (1987). "Closed-form solution of absolute orientation using unit quaternions". In: *Journal of the Optical Society of America* 4.4, S. 629–642.

Horn, Berthold K. P., Hugh M. Hilden und Shahariar Negahdaripour (1988). "Closed-form solution of absolute orientation using orthonormal matrices". In: *Journal of the Optical Society of America* 5, S. 1127–1135.

Horn, Berthold K. P. und Brian G. Schunck (1981). "Determining Optical Flow". In: *ARTIFICIAL INTELLIGENCE* 17, S. 185–203.

Howard, T.M., A. Morfopoulos, J. Morrison, Y. Kuwata, C. Villalpando, L. Matthies und M. McHenry (2012). "Enabling Continuous Planetary Rover Navigation through FPGA Stereo and Visual Odometry". In: *IEEE Aerospace Conference*, S. 1–9.

Huang, Albert (2014). *Fast Odometry from VISion*. `http://code.google.com/p/fovis`, besucht im April 2014.

Huang, Albert, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox und Nicholas Roy (2011). "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera". In: *International Symposium on Robotics Research (ISRR)*. Flagstaff, Arizona, USA.

Irschara, Arnold, Christopher Zach, Jan-Michael Frahm und Horst Bischof (2009). "From Structure-from-Motion Point Clouds to Fast Location Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Miami, Florida, USA, S. 2599–2606.

Israël, Jonathan und Aurélien Plyer (2013). "A Brute Force Approach to Depth Camera Odometry". In: *Consumer Depth Cameras for Computer Vision*. Hrsg. von Andrea Fossati, Juergen Gall, Helmut Grabner, Xiaofeng Ren und Kurt Konolige. Advances in Computer Vision and Pattern Recognition. Springer London, S. 49–60.

Izadi, Shahram, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison und Andrew Fitzgibbon (2011). "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera". In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. UIST '11. Santa Barbara, California, USA: ACM, S. 559–568.

Jähne, Bernd (2002). *Digitale Bildverarbeitung*. 5. Aufl. Berlin Heidelberg: Springer-Verlag.

Jean, Jared St. (2012). *Kinect Hacks*. O'Reilly Media.

Jirawimut, Rommanee, Simant Prakoonwit, Franjo Cecelja und Wamadeva Balachandran (2003). "Visual Odometer for Pedestrian Navigation". In: *IEEE Transactions on Instrumentation and Measurement* 52.4, S. 1166–1173.

Jutzi, Boris (2009). "Investigations on Ambiguity Unwrapping of Range Images". In: *Laserscanning 2009. International Archives of Photogrammetry and Remote Sensing 38 (Part 3/W8)*. Hrsg. von F. Bretar, M. Pierrot-Deseilligny und G. Vosselman, S. 265–270.

— (2010). "Extending the range measurement capabilities of modulated range imaging devices by time-frequency-multiplexing". In: *Allgemeine Vermessungs-Nachrichten (AVN)* 2, S. 54–62.

Kalal, Zdenek, Krystian Mikolajczyk und Jiri Matas (2010). "Forward-Backward Error: Automatic Detection of Tracking Failures". In: *20th International Conference on Pattern Recognition (CVPR)*, S. 2756–2759.

Kato, Hirokazu und Mark Billinghurst (1999). "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System". In: *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*. San Francisco, USA.

Kerl, C., J. Sturm und D. Cremers (2013). "Robust Odometry Estimation for RGB-D Cameras". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Khoshelham, Kourosh (2011). "Accuracy Analysis of Kinect Depth Data". In: *ISPRS Workshop Laser Scanning*. Bd. XXXVIII. 5/W12.

Khoshelham, Kourosh und Sander Oude Elberink (2012). "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications". In: *Sensors* 12.2, S. 1437–1454.

Kitt, Bernd, Andreas Geiger und Henning Lategahn (2010). "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme". In: *IEEE Intelligent Vehicles Symposium*. San Diego, USA.

Kiyokawa, Kiyoshi, Mark Billinghurst, Bruce Campbell und Eric Woods (2003). "An Occlusion-Capable Optical See-through Head Mount Display for Supporting Co-located Collaboration". In: *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*. Tokyo, Japan, S. 133–141.

Klein, Georg und David Murray (2009). "Parallel Tracking and Mapping on a Camera Phone". In: *IEEE International Symposium on Mixed and Augmented Reality - Arts, Media and Humanities. ISMAR-AMH*. Orlando, Florida, USA, S. 83–86.

Klippenstein, J. und Hong Zhang (2007). "Quantitative Evaluation of Feature Extractors for Visual SLAM". In: *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*, S. 157–164.

Kodak (1999). *Kodak Lossless True Color Image Suite*. `http://r0k.us/graphics/kodak/index.html`, besucht im März 2015.

Konolige, Kurt (1997). "Small Vision Systems: Hardware and Implementation". In: *Proceedings of the International Symposium on Robotics Research*.

Konolige, Kurt, Motilal Agrawal und Joan Solà (2007). "Large Scale Visual Odometry for Rough Terrain". In: *Proceedings of the International Symposium on Robotics Research*.

Konolige, Kurt und Patrick Mihelich (2011). *Technical description of Kinect calibration*. `http://www.ros.org/wiki/kinect_calibration/technical`, besucht im März 2015.

Kramer, Jeff, Nicolas Burrus, Florian Echtler, Daniel Herrera C. und Matt Parker (2012). *Hacking the Kinect*. Apress.

Lamb, Philip (2010). *ARToolKit*. `http://www.hitl.washington.edu/artoolkit/`, besucht im Mai 2010.

Lee, Hsien-Che (2005). *Introduction to color imaging science*. eng. Cambridge Univ. Press.

Lee, Jong Weon, Suya You und Ulrich Neumann (2002). "Tracking with Omni-directional Vision for Outdoor AR Systems". In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR02)*. IEEE. Darmstadt, S. 47–56.

Leebmann, Johannes (2003). "A stochastic analysis of the calibration problem for Augmented Reality systems with see-through head-mounted displays". In: *ISPRS Journal of Photogrammetry and Remote Sensing*. ISPRS, S. 400–408.

— (2004). "An Augmented Reality System for Earthquake disaster response". In: *XXth ISPRS Congress*. Istanbul.

Lepetit, Vincent und Pascal Fua (2005). "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey". In: *Foundations and Trends in Computer Graphics and Vision* 1.1, S. 1–89.

Lepetit, Vincent, Pascal Lagger und Pascal Fua (2005). "Randomized Trees for Real-Time Keypoint Recognition". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Bd. 2, S. 775–781.

Livingston, Mark A., Dennis Brown und Joseph L. Gabbard (2002). "An Augmented Reality System for Military Operations in Urban Terrain". In: *Proceedings of Interservice / Industry Training, Simulation and Education Conference (I/ITSEC) 2002*. Orlando, Florida.

Longuet-Higgins, H. C. (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293, S. 133–135.

Lowe, David (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2, S. 91–110.

Lu, Wenmiao und Yap-Peng Tan (2003). "Color Filter Array Demosaicking: new Method and Performance Measures". In: *IEEE Transactions on Image Processing* 12.10, S. 1194–1210.

Lucas, B.D. und T. Kanade (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *IJCAI81*, S. 674–679.

Lui, Wen Lik Dennis, Titus Jia Jie Tang, Tom Drummond und Wai Ho Li (2012). "Robust Egomotion Estimation Using ICP in Inverse Depth Coordinates". In: *IEEE International Conference on Robotics and Automation (ICRA)*, S. 1671–1678.

Maimone, Mark, Yang Cheng und Larry Matthies (2007). "Two years of Visual Odometry on the Mars Exploration Rovers". In: *Journal of Filed Robotics* 24.3, S. 169–186.

Malvar, Henrique S., Li-wei He und Ross Cutler (2004). "High-quality Linear Interpolation for Demosaicing of Bayer-patterned Color Images". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Bd. 3. IEEE, S. 485–488.

Marchand, Éric und François Chaumette (2002). "Virtual Visual Servoing: a framework for real-time augmented reality". In: *Computer Graphics Forum*. Bd. 21. 3. Rennes, Frankreich, S. 289–297.

Martinez, Manuel und Rainer Stiefelhagen (2013). "Kinect Unleashed: Getting Control over High Resolution Depth Maps". In: *IAPR Conference on Machine Vision Applications*. Kyoto, Japan.

McGlone, J. Chris, Hrsg. (2004). *Manual of Photogrammetry*. 5. Aufl. ASPRS.

McGuire, Morgan (2008). "Efficient, High-Quality Bayer Demosaic Filtering on GPUs". In: *Journal of Graphics, GPU, and Game Tools* 13.4, S. 1–16.

Merrill, Richard Billings (1998). *Color Separation in an Active Pixel Cell Imaging Array Using a Triple-well Structure*. US Patent 5965875.

Mikolajczyk, Krystian und Cordelia Schmid (2005). "A Performance Evaluation of Local Descriptors". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence* 27.10, S. 1615–1630.

Milgram, Paul und Fumio Kishino (1994). "A Taxonomy of Mixed Reality Visual Displays." In: *IEICE Transactions on Information Systems* E77-D.12, S. 1321–1329.

Molton, Nicholas David (1998). "Computer Vision as an Aid for the Visually Impaired". Diss. Robotics Research Group, Department of Engineering Science University of Oxford.

Montiel, J. M. M., Javier Civera und Andrew J. Davison (2006). "Unified Inverse Depth Parametrization for Monocular SLAM". In: *Robotics: Science and Systems*. Hrsg. von Gaurav S. Sukhatme, Stefan Schaal, Wolfram Burgard und Dieter Fox. University of Pennsylvania, Philadelphia, Pennsylvania, USA: The MIT Press.

Moreno-Noguer, Francesc, Vincent Lepetit und Pascal Fua (2007). "Accurate Non-Iterative O(n) Solution to the PnP Problem". In: *IEEE International Conference on Computer Vision*. Rio de Janeiro, Brazil.

— (2008). "Pose Priors for Simultaneously Solving Alignment and Correspondence". In: *Proceedings of the 10th European Conference on Computer Vision: Part II*. Bd. 5303. LNCS. Springer Berlin Heidelberg, S. 405–418.

Mutto, Carlo Dal, Pietro Zanuttigh und Guido M. Cortalazzo (2012). *Time-of-Flight Cameras and Microsoft Kinect*. Springer US.

Najafi, Hesam und Gudrun Klinker (2003). "Model-based Tracking with Stereovision for AR". In: *International Symposium on Augmented Reality (ISMAR)*.

Nistér, David (2004). "An Efficient Solution to the Five-point Relative Pose Problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6, S. 756–770.

Nistér, David, Oleg Naroditsky und James Bergen (2004). "Visual Odometry". In: *Proceedings on Computer Vision and Pattern Recognition (CVPR)*. Bd. 1, S. 652–659.

— (2006). "Visual Odometry for Ground Vehicle Applications". In: *Journal of Field Robotics* 23.1, S. 3–20.

Occipital Structure Sensor (2014). `http://structure.io`, besucht im April 2014.

Ohta, Naoya und Kenichi Kanatani (1998). "Optimal Estimation of Three-Dimensional Rotation and Reliability Evaluation". In: *5th European Conference on Computer Vision (ECCV)*. Bd. 1406. Lecture Notes in Computer Science. Freiburg: Springer Berlin/Heidelberg, S. 175–188.

Open Source Computer Vision Library (OpenCV) (2015). `http://opencv.org`, besucht im März 2015.

Ottlik, Artur und Hans-Hellmut Nagel (2007). "Initialization of Model-Based Vehicle Tracking in Video Sequences of Inner-City Intersections". In: *International Journal of Computer Vision* 80.2, S. 211–225.

Parker, T. J., M. C. Malin, F. J. Calef, R. G. Deen, H. E. Gengl, M. P. Golombek, J. R. Hall, O. Pariser, M. Powell, R. S. Sletten u. a. (2013). "Localization and'Contextualization'of Curiosity in Gale Crater, and Other Landed Mars Missions". In: *Lunar and Planetary Institute Science Conference Abstracts*. Bd. 44.

Patras, I., N. Alvertos und G. Tziritas (1996). "Joint disparity and motion field estimation in stereoscopic image sequences". In: *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. Bd. 1, 359–363 vol.1.

Paz, Lina M., Pedro Piniés, Juan D. Tardós und J. Neira (2008). "Large Scale 6DOF SLAM with a stereo camera in hand". In: *IEEE Transactions on Robotics* 24.5, S. 946–957.

Perko, Roland, Philipp Fürnstahl, Joachim Bauer und Andreas Kuhn (2005). "Geometrical Accuracy of Bayer Pattern Images". In: *13th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*. Bd. 13, S. 117–120.

Pfeiffer, David und Uwe Franke (2011). "Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data". In: *British Machine Vision Conference (BMVC)*. Dundee, Scotland.

Piekarski, Wayne und Bruce H. Thomas (2002a). "Measuring ARToolKit Accuracy in Long Distance Tracking". In: *Augmented Reality Toolkit, The First IEEE International Workshop*, S. 2.

— (2002b). "Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments". In: *Augmented Reality Toolkit, The First IEEE International Workshop*.

Pierrot-Deseilligny, Marc und I. Clery (2011). "Apero, an Open Source Bundle Adjusment Software for Automatic Calibration and Orientation of set of Images". In: *Proceedings of the ISPRS Symposium, 3DARCH11*, S. 269–277.

Pollefeys, Marc (1999). "Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences". Diss. Katholieke Universiteit Leueven.

Pradeep, Vivek, Gerard Medioni und James Weiland (2010). "Robot vision for the visually impaired". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, S. 15–22.

PrimeSense NITE Algorithms 1.5 (2011). PrimeSense.

Ramanath, Rejeev, Welsley Snyder, Griff Bilbro und William Sander (2002). "Demosaicking methods for Bayer Color Arrays". In: *Journal of Electronic Imaging* 11.3, S. 306–315.

Regenbrecht, Holger, Claudia Ott, Michael Wagner, Tim Lum, Petra Kohler, Wilhelm Wilke und Erich Mueller (2003). "An Augmented Virtuality Approach to 3D Videoconferencing". In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. Tokyo, Japan, S. 290–291.

Reitmayr, Gerhard und Tom Drummond (2006). "Going Out: Robust Model-based Tracking for Outdoor Augmented Reality". In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*. Santa Barbara, California, USA.

Rodehorst, Volker (2004). "Photogrammetrische 3D-Rekonstruktion im Nahbereich durch Auto-Kalibrierung mit projektiver Geometrie". Diss. Technische Universität Berlin.

Rodriguez, J.J. und J.K. Aggarwal (1990). "Stochastic analysis of stereo quantization error". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.5, S. 467–470.

Rosten, Edward und Tom Drummond (2005). "Fusing Points and Lines for High Performance Tracking". In: *IEEE International Conference on Computer Vision (ICCV)*. Bd. 2. Peking, S. 1508–1515.

— (2006). "Machine learning for high-speed corner detection". In: *European Conference on Computer Vision*. Bd. 1, S. 430–443.

Rusinkiewicz, Szymon und Marc Levoy (2001). "Efficient Variants of the IPC Algorithm". In: *Third International Conference on 3-D Digital Imaging and Modeling*. Quebec City, Canada, S. 145–152.

Sanden, Horst von (1908). "Die Bestimmung der Kernpunkte in der Photogrammetrie". Diss. Universität Göttingen.

Sands, Jamie, Shaun W. Lawson und David Benyon (2004). "Target Selection in Augmented Reality Worlds". In: *4th International Conference on Computational Science (ICCS)*. Bd. III. Krakau, Polen: Springer, S. 936–945.

Scaramuzza, Davide und Friedrich Fraundorfer (2011). "Visual Odometry: Part I - The First 30 Years and Fundamentals". In: *IEEE Robotics and Automation Magazine* 18.4, S. 80–92.

— (2012). "Visual Odometry: Part II - Matching, Robustness, Optimization, and Applications". In: *IEEE Robotics and Automation Magazine* 19.1, S. 1–11.

Scharstein, Daniel und Richard Szeliski (2002). "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *International Journal of Computer Vision* 47.1, S. 7–42.

Schindler, Grant, Matthew Brown und Richard Szeliski (2007). "City-scale location recognition." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, USA, S. 1–7.

Schmidt, Florian und Stefan Hinz (2011). "A Scheme for the Detection and Tracking of People Tuned for Aerial Image Sequences". In: *Photogrammetric Image Analysis (PIA)*. Hrsg. von Uwe Stilla, Franz Rottensteiner, Helmut Mayer, Boris Jutzi und Matthias Butenuth. LNCS 6952. ISPRS. Munich, Germany: Springer, Heidelberg, S. 257–270.

Schneider, Johannes, Falko Schindler, Thomas Läbe und Wolfgang Förstner (2012). "Bundle Adjustment for Multi-camera Systems with Points at Infinity". In: *22nd Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*.

Schreer, Oliver (2005). *Stereoanalyse und Bildsynthese*. Springer Verlag.

Segal, Aleksandr, Dirk Haehnel und Sebastian Thrun (2009). "Generalized-ICP". In: *Robotics: Science and Systems*. Bd. 2, S. 4.

Shi, Jianbo und Carlo Tomasi (1994). "Good Features to Track". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE CS Press, S. 593–600.

Siciliano, Bruno und Oussama Khatib, Hrsg. (2008). *Handbook of Robotics*. Springer.

Sinha, S., J.-M. Frahm und Marc Pollefeys (2006). *GPU-based Video Feature Tracking And Matching*. Techn. Ber. TR06-012. University of North Carolina at Chapel Hill.

SIPI (2015). *USC SIPI Image Database*. `http://sipi.usc.edu/database/`, besucht im März 2015.

Smisek, Jan, Michal Jancosek und Tomas Pajdla (2013). "3D with Kinect". In: *Consumer Depth Cameras for Computer Vision*. Hrsg. von Andrea Fossati, Juergen Gall, Helmut Grabner, Xiaofeng Ren und Kurt Konolige. Advances in Computer Vision and Pattern Recognition. Springer London, S. 3–25.

Souma, Yasuyuki, Hidemi Yamachi, Yasuhiro Tsujimura und Yasushi Kambayashi (2012). "Interaction in Augmented Reality by Means of Z-buffer Based Collision Detection". In: *ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions*, S. 315–318.

Steder, Bastian, Radu Bogdan Rusu, Kurt Konolige und Wolfram Burgard (2010). "NARF: 3D Range Image Features for Object Recognition". In: *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Bd. 44.

Steinbrücker, Frank, Jürgen Sturm und Daniel Cremers (2011). "Real-Time Visual Odometry from Dense RGB-D Images". In: *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, S. 719–722.

Sturm, J., N. Engelhard, F. Endres, W. Burgard und D. Cremers (2012). "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proceedings of the International Conference on Intelligent Robot Systems (IROS)*.

Sturm, J., S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers und R. Siegwart (2011). "Towards a benchmark for RGB-D SLAM evaluation". In: *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*. Los Angeles, USA.

Sünderhauf, Niko, Kurt Konolige, Simon Lacroix und Peter Protzel (2006). "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle". In: *Autonome Mobile Systeme 2005*. Springer, S. 157–163.

Suthau, Tim (2003). "Augmented Reality Techniken für den Einsatz in der Leberchirurgie". In: *23. Wissenschaftlich-Technische Jahrestagung der DGPF*. Bd. 12. DGPF. Bochum, S. 301–310.

Suthau, Tim (2006). "Augmented Reality - Positionsgenaue Einblendung räumlicher Informationen in einem See Through Head Mounted Display für die Medizin am Beispiel der Leberchirurgie". Diss. Institut für Computer Vision und Fernerkundung, TU Berlin.

Sutherland, Ivan E. (1968). "A head-mouted three-dimensional display". In: *AFIPS Conference*. Bd. 33, S. 757–764.

Szalavari, Zsolt, Dieter Schmalsieg, Anton Fuhrmann und Michael Gervautz (1998). "Studierstube, An Environment for Collaboration in Augmented Reality". In: *Virtual Reality*. Bd. 3 (1).

Thormählen, Thorsten (2006). "Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge". Diss. Laboratorium für Informationstechnologie (LFI), Universität Hannover.

Tuytelaars, Tinne und Krystian Mikolajczyk (2008). *Local Invariant Feature Detectors: A Survey*. Bd. 3. Foundation and Trends in Computer Graphics and Vision 3. Now Publishers.

Ulrich, Markus, Christian Wiedemann und Carsten Steger (2009). "CAD-Based Recognition of 3D Objects in Monocular Images". In: *IEEE International Conference on Robotics and Automation*. Kobe, Japan, S. 1191–1198.

Urban, Steffen, Jens Leitloff, Sven Wursthorn und Stefan Hinz (2013). "Self-localization of a Multi-fisheye Camera Based Augmented Reality System in Textureless 3d Building Models". In: *ISPRS Workshop on Image Sequence Analysis (ISA)*. Hrsg. von C. Mallet, A. Yilmaz, Y. Vizilter und M. Ying Yang. Bd. II-3/W2, S. 43–48.

Vacchetti, Luca, Vincent Lepetit und Pascal Fua (2004a). "Combining edge and texture information for real-time accurate 3D camera tracking". In: *International Symposium on Mixed and Augmented Reality*. Arlington.

— (2004b). "Stable Real-Time 3D Tracking using Online and Offline Information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wagner, Daniel, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond und Dieter Schmalstieg (2008). "Pose Tracking from Natural Features on Mobile Phones". In: *7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Campridge, UK, S. 125–134.

Wagner, Martin (2002). "Building Wide-Area Applications with the AR Toolkit". In: *First IEEE International Augmented Reality Toolkit Workshop*. Darmstadt, S. 1–7.

Walker, Michael W., Lejun Shao und Richard A. Volz (1991). "Estimating 3-D location parameters using dual number quaternions". In: *CVGIP: Image Understanding* 54.3, S. 358–367.

Wedel, Andreas, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke und Daniel Cremers (2011). "Stereoscopic Scene Flow Computation for 3D Motion Understanding". In: *International Journal of Computer Vision* 95, S. 29–51.

Wedel, Andreas und Daniel Cremers (2011). *Stereoscopic Scene Flow for 3D Motion Analysis*. Springer-Verlag London Limited.

Weinmann, Martin (2013). "Visual Features – From Early Concepts to Modern Computer Vision". In: Hrsg. von G. M. Farinella, S. Battiato und R. Cipolla. Advanced Topics in Computer Vision. Springer London, S. 1–35.

Weinmann, Martin, Sven Wursthorn und Boris Jutzi (2011). "Semi-automatic image-based co-registration of range imaging data with different characteristics". In: *PIA11 - Photogrammetric Image Analysis*. Bd. 38 Part 3/W22, S. 119–124.

Welch, Greg und Eric Foxlin (2002). "Motion Tracking: No Silver Bullet, but a Respectable Arsenal". In: *IEEE Computer Graphics and Applications*.

Welch, Robert B. (1978). *Perceptual Modification: Adapting to Altered Sensory Environments*. Academic Press.

Wuest, Harald, Florent Vial und Didier Stricker (2005). "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality". In: *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Santa Barbara, California, USA.

Wuest, Harald, Folker Wientapper und Didier Stricker (2007). "Adaptable Model-Based Tracking Using Analysis-by-Synthesis Techniques". In: *12th International Conference on Computer Analysis of Images and Patterns (CAIP)*. Bd. 4673. LNCS. Wien: Springer Berlin/Heidelberg, S. 20–27.

Wujanz, Daniel, Sven Weisbrich und Frank Neitzel (2011). "3D-Mapping mit dem Microsoft Kinect Sensor - erste Untersuchungsergebnisse". In: *Oldenburger 3D Tage*. Bd. 10, S. 1–10.

Wursthorn, Sven, Alexandre Hering Coelho und Guido Staub (2004). "Applications for Mixed Reality". In: *XXth ISPRS Congress*. Istanbul, Türkei.

Wursthorn, Sven, Alexandre Hering Coelho, Johannes Leebmann und Guido Staub (2005). In: *Digitale Bildverarbeitung. Anwendungen in Photogrammetrie, Fernerkundung und GIS*. Hrsg. von Hans-Peter Bähr und Thomas Vögtle. 4. Aufl. Wichmann. Kap. Erweiterte Realität.

Zheng, Yan-Tao, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua und Hartmut Neven (2009). "Tour the World: building a web-scale landmark recognition engine". In: *International Conference on Pattern Recognition (CVPR)*. Miami, Florida, USA, S. 1085–1092.

Zhu, Zhiwei, Taragay Oskiper, Supun Samarasekera, Rakesh Kumar und Harpreet S. Sawhney (2008). "Real-time Global Localization With a Pre-built Visual Landmark Database". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, USA, S. 1–8.