INAUGURAL - DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht - Karls - Universität
Heidelberg

vorgelegt von
Diplom - Informatiker Florian Haug
aus Kempten

Tag der mündlichen Prüfung: 16.12.2010

Ansichtsbasierte 6 DoF Objekterkennung
mit
lokalen kovarianten Regionen

Gutachter : Prof. Dr. Bernd Jähne
Prof. Dr. Reinhard Männer

Abstract

This thesis presents a new approach to object detection and pose estimation, that uses the local deformations of corresponding *covariant* regions to predict the *6 degrees of freedom* (DoF) rigid transformation between a set of aligned model and camera views. Thereto algorithms are developed, which allows the prediction of an independent 6 DoF pose hypothesis from a *single* feature correspondence, given the depth and plane normal at one region center. Clusters of these local hypotheses are used as a coarse object localization and as a segmentation respectively outlier removal for a global pose refinement step. The approach allows an integrated handling of multiple camera and model images and different local covariant feature types, including features on depth images. The subsequent step determines globally the 6 DoF object pose which fits best to the local 2D-3D or 3D-3D correspondences of the region centers within a cluster. The combination of local and global analysis allows an accurate localization even with illumination distortions, large view point changes, occlusion, ambiguities and cluttered scenes. This was evaluated by means of 6 objects and numerous experiments, in which a pose accuracy of under 1 mm and 1° could be achieved. Almost all steps are fine granular parallelizable and therefore enable a runtime on modern hardware below 0.4 s. The training of new object models can be done autonomously with the aid of an industrial robot and a mounted stereo-camera.

Zusammenfassung

Diese Arbeit präsentiert einen neuen Ansatz zur Detektion und Lokalisation von Objekten, welcher die lokale Deformation korrespondierender, *kovarianter* Regionen nutzt, um die *6 Freiheitsgrade* (DoF) einer Starrkörpertransformation zwischen einer Menge registrierter Modell- und Kameraansichten zu schätzen. Dazu werden Algorithmen entworfen, die es erlauben, aus jeder *einzelnen* Regionenkorrespondenz eine unabhängige 6 DoF Lagehypothese abzuleiten, falls die Oberflächennormale und Tiefe eines Regionenzentrums bekannt ist. Cluster dieser lokalen Hypothesen werden als grobe Lokalisierung und robuste Segmentierung bzw. Ausreißereliminierung für eine nachfolgende globale Lageerkennung genutzt. Dieses Vorgehen erlaubt eine integrierte Verarbeitung aller vorhandener Modell- und Kameraansichten und erlaubt die Fusion unterschiedlicher kovarianter Regionentypen, inkl. Regionen auf Basis von Tiefenbildern. Die nachfolgende Auswertung ermittelt die 6 DoF Objektlage, welche am besten den 2D-3D oder 3D-3D Korrespondenzen der Regionenzentren innerhalb eines Clusters entspricht. Die Kombination von lokaler und globaler Auswertung erlaubt selbst bei starken Beleuchtungsstörungen, großen Blickwinkeländerungen, Verdeckungen, Mehrdeutigkeiten und komplexen Szenen eine akkurate und robuste Lokalisation. Dies wurde anhand 6 Bauteilen und ausführlichen Experimenten verifiziert, wobei Genauigkeiten der Lage unter 1 mm und 1° erreicht werden konnten. Nahezu alle Algorithmen sind fein granular parallelisierbar und ermöglichen daher eine Auswertezeit auf moderner Hardware unter 0.4 s. Das Einlernen eines Objektmodells erfolgt mit Hilfe eines Industrieroboters und einer darauf montierten Stereokamera vollständig autonom.

Danksagung

Ich möchte allen danken, die mir bei der Erstellung dieser Arbeit geholfen haben. Zu großem Dank bin ich meinem Doktorvater Prof. Bernd Jähne vom Heidelberg Collaboratory of Image Processing (HCI) der Universität Heidelberg verpflichtet, der mir jederzeit, auch kurzfristig in allen wissenschaftlichen und verwaltungstechnischen Fragen auf eine angenehme Art und Weise mit Rat und Tat zur Seite stand. Weiterhin möchte ich Walter Happold vom Zentralbereich Forschung und Vorausentwicklung der Robert Bosch GmbH danken, der mir meine Forschungen im Rahmen eines spannenden Projekts ermöglichte und mich aufs Vollste unterstützte.

Von Herzen möchte ich auch meinen Kollegen und Freunden in Schwieberdingen danken, die eine so angenehme Arbeitsatmosphäre geschaffen haben und in den schwierigen Zeiten, die so eine Arbeit unweigerlich mit sich bringt, immer wieder für heitere, kraftspendende Momente gesorgt haben. Besonderen Dank gilt dabei Dr. Christian Knoll, meinem Mitdoktoranden Manuel Glas und meinen drei Diplomanden Hagen Schlegel, Kahnh Q. Tran und Christian Zöllner für ihre inspirierenden wissenschaftlichen Diskussionen und ihrer Unterstützung bei der Umsetzung meiner Ideen. Weiterhin möchte ich Dr. Christian Perwass für die Bereitstellung seiner 3D-Visualisierungssoftware CluViz danken. Ebenso möchte ich (in alphabetischer Reihenfolge) Alexander, Andreas (3x), Annika, Christian, Corinna, Dieter, Frank, Jan, Jens, Marc, Maria, Marina, Martin, Matthias (2x), Michael, Patrick, Peter (2x), Sabine (2x), Stefan, Sven, Thomas (2x) und Winfried danken.

Zum Schluss möchte ich noch meinen beiden Mitbewohnern Arvid und Georg, all meinen Freunden, Verwandten und vor allem meinen Eltern danken, dich mich im Vorfeld und während dieser Arbeit unterstützt und umsorgt haben - ohne Euch hätte ich es nicht geschafft!

Inhaltsverzeichnis

EINE MÄCHTIGE FLAMME
ENTSTEHT AUS EINEM
WINZIGEN FUNKEN.

Dante Alighieri

Kapitel 1

Einleitung

Bereits im Jahr 1954 reichte Georg Devol ein Patent ein, welches den Grundstein für den 1961 in Betrieb genommenen ersten industriellen Roboter *Unimate* bilden sollte (Dev54). Schon der Name *Programmed Article Transfer*, Programmierter Warentransfer des US-Patents 2988237 weist auf das bis heute wichtigste Aufgabengebiet der Robotik hin, der Manipulation von Objekten. Während die frühen Einsatzszenarien durch starre Abläufe in vordefinierten Arbeitsumgebungen geprägt waren, spielt in modernen Aufgabenstellungen aufgrund geringerer Produktlebenszeiten und damit einhergehender flexiblerer Anlagen in zunehmendem Maße die Kognition, d. h. die Erfassung einer sich teilweise unbekannten Umgebung und eine entsprechende Anpassung der Abläufe eine Rolle. Objekte befinden sich nicht mehr in Werkstückträgern sondern müssen lose von einem Band abgeholt werden; Zuführmaterial wird nicht mehr in eine Maschine einsortiert sondern als Schüttgut aus einer Kiste abgegriffen; Paletten werden nicht mehr exakt ausgerichtet sondern nur noch grob platziert. Daher beschäftigt sich die vorliegende Arbeit mit der Detektion von Objekten in Grauwertbildern und der Bestimmung deren *Lage* innerhalb der beobachteten Umgebung in allen 6 Freiheitsgraden (DoF's[1]), genauer den 3 translatorischen DoF's der *Position* und den 3 rotatorischen DoF's der *Orientierung*.

Des Weiteren soll das Verfahren im Gegensatz zu frühen Objekterkennungssystemen nicht mehr speziell auf ein Werkstück optimiert werden, sondern eine universelle Lösungsstrategie für eine große Klasse unterschiedlicher Objekte bereitstellen. Dazu gehört auch die selbständige Akquise aller relevanten Informationen zur Lokalisation eines Objekts in einem offline, d. h. nicht innerhalb des normalen Produktionszyklusses befindlichen Trainingsschritts.

Im Folgenden wird der konzeptionelle Entwurf dieser Arbeit motiviert, danach das entwickelte System, die wissenschaftlichen Neuerungen und die eingesetzte Sensorik vorgestellt und abschließend ein Überblick über den Aufbau der Arbeit gegeben.

1.1 Motivation

Bildverstehen oder *maschinelles Sehen* beschreibt den kognitiven Prozess, mit einem künstlichen System aus einer Matrix von Intensitätswerten einer Kamera, abstrakte Informationen über die beobachtete Szene abzuleiten. Dabei treten zwei ganz entscheidende Probleme auf.

Zum einen lassen sich Objekte einer Szene nicht direkt beobachten, sondern nur durch ein komplexes Zusammenspiel zwischen Objektoberfläche und Beleuchtung. Modelliert wird dies in der *bidirektionalen Reflektanzverteilungsfunktion*, kurz BRDF[2] welche das von der Kamera beobachtete Abstrahlverhalten eines Objekts in Abhängigkeit der einfallenden Beleuchtung modelliert. Diese Funk-

[1] engl.: degree of freedom

[2] engl.: bidirectional reflectance distribution function

tion ist von der Oberflächennormalen, den Materialeigenschaften, dem Einfallswinkel des Lichtes, dessen Wellenlänge und dem Beobachtungswinkel abhängig. Dies führt dazu, dass sich das *Erscheinungsbild* eines Objekts, d. h. dessen Abstrahlung auf den Kamerachip bzw. dessen Intensitätswerte in der Bildmatrix, schon durch kleine Szenenänderungen stark wandeln kann. Da beim maschinellen Sehen im Allgemeinen weder der Szenenaufbau noch die BRDF der einzelnen Objekte bekannt ist, können diese Veränderungen des Erscheinungsbilds nicht vernünftig modelliert werden. Man behilft sich daher in der Praxis mit sehr starken Vereinfachungen, wie z. B. linearen Beleuchtungsmodellen, die die Realität allerdings nur ungenügend widerspiegeln.

Zum anderen kann mittels einer Kamera die 3D-Szene nicht direkt beobachtet werden sondern nur deren Zentralprojektion auf der Bildebene. Dies führt zu einem Verlust der Tiefeninformationen sowie aller Oberflächeninformationen verdeckter Bereiche. Bewegungen eines Objekts innerhalb der 3D-Szene verursachen daher komplexe geometrische Änderungen des projizierten 2D-Erscheinungsbilds. Diese Änderungen lassen sich innerhalb der Bilddomäne nur schwer und mit der beim maschinellen Sehen normalerweise nicht vorhandenen Kenntnis aller 3D-Informationen modellieren. In der Praxis behilft man sich mit Annahmen, wie einer lokalen Parallelprojektion oder einem affinen Transformationsmodell des Erscheinungsbilds, welche die Realität allerdings nur fehlerhaft widerspiegeln.

Zentraler Bestandteil des maschinellen Sehens ist daher die Entwicklung von Algorithmen, die sich im Bezug auf die ungenügend modellierbaren photometrischen und geometrischen Einflüssen robust verhalten. Ein sinnvoller Ausgangspunkt ist dabei der Mensch, dessen visuelle kognitive Fähigkeiten im Umgang mit den beiden obigen Problemen hervorragend sind. Auch wenn dessen visuelle Sinnesverarbeitung und Abstraktionsfähigkeit in der Physiologie nur Ansatzweise verstanden ist, liegt es nahe, sich bei dem Entwurf eines Objektlokalisationssystems an deren Theorien zur menschlichen Wahrnehmung zu orientieren.

I. Biederman hat 1987 die Theorie der *komponentialen Erkennung* vorgestellt (Bie87), die zwei interessante Konzepte beinhaltet. Das erste besagt, dass wahrgenommene Objekte vom Menschen in einfache lokale *Komponenten*, den Geons zergliedert werden. Diese Geons können aufgrund von charakteristischen Strukturen ihres Erscheinungsbildes auch bei Beleuchtungsschwankungen und Blickwinkeländerungen robust erkannt werden. Bei Biederman bilden sich diese Strukturen aus Kanten mit bestimmten Eigenschaften, wie Parallelität, Symmetrie oder Krümmung. Ganz allgemein kann man sie aber auch als *saliente*, d. h. herausragende bzw. gut erkennbare Strukturen bezeichnen, die unter photometrischen und geometrischen Einflüssen robust erkennbar sind.

Das zweite Konzept besagt, dass die Objekterkennung in unbekannten Szenen nicht über einzelne Geons erfolgt, sondern über deren relative Anordnung, d. h. der *räumlichen Komposition* mehrerer Komponenten. Die Geons werden in unbekannten Bildern bzw. Szenen aufgrund der salienten Strukturen unabhängig voneinander erkannt und sind wegen ihrer lokalen Struktur für sich alleine allerdings weder aussagekräftig noch zuverlässig. Über eine globale Betrachtung der räumlichen Beziehung zwischen mehreren lokalen Komponenten, lassen sich Unsicherheiten auflösen und die gefundene Komposition daher selbst bei Verdeckungen oder lokalen Störungen robust einem Objekt zuordnen.

Das neu entwickelte Objektlokalisierungssystem dieser Arbeit orientiert sich an den Konzepten zur Detektion salienter lokaler Komponenten und der Auswertung ihrer räumlichen Komposition und versucht eine ähnliche Auswertestrategie mit lokaler und globaler Verarbeitung auf einer Maschine nachzubilden.

Der Mensch baut sich ein *Modell* eines Objekts, d. h. die Komponenten und ihre räumliche Anordnung intuitiv auf, indem er das Objekt von verschiedenen *Ansichten* bzw. Blickwinkeln betrachtet. Für diese ansichtsbasierte Erfassung muss kein vollständiges 3D-Modell des Objekts rekonstruiert werden. Es genügt die Beschreibung und Lage der lokalen Komponenten in einer gemeinsamen Repräsentation. Dieses Vorgehen wird auf den offline Trainingsschritts des neuen Systems übertragen

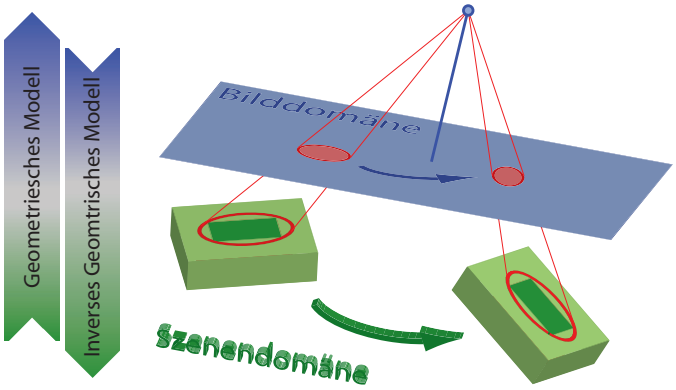

Abbildung 1.1: Darstellung der Beziehungen (rot) von Bewegungen bzw. Lageunterschieden eines Objektausschnitts in der 3D-Szene (grün) und der beobachteten Deformationen der zugehörigen Projektion in der 2D-Bildebene (blau). Die Beschreibung der Bildtransformation bei bekannter 3D-Bewegung wird in dieser Arbeit als geometrisches Modell bezeichnet, der umgekehrte Fall als inverse Problemstellung.

und ermöglicht eine autonome Modellgenerierung.

1.2 Überblick und wissenschaftlicher Beitrag

In der vorliegenden Arbeit wird ein neues System präsentiert, welches automatisch detektierbare saliente Regionen in Grauwertbildern nutzt, um daraus lokale Korrespondenzen, d. h. übereinstimmende Bereiche zwischen eingelernten Ansichten eines Objekts und unbekannten Suchansichten zu finden. Aus jeder Korrespondenz wird eine (als lokal bezeichnete) Hypothese über die Position und Orientierung des eingelernten Objekts in den Suchansichten geschätzt und übereinstimmende Gruppen davon zur robusten Berechnung einer (globalen) Lagehypothese herangezogen.

Zum Auffinden der salienten Regionen und der Korrespondenzen werden in dieser Arbeit bekannte Verfahren wie SIFT (Low04) oder MSER (MCUP04) verwendet, die sich unter dem Begriff „*Interessante Regionen*“ zusammenfassen lassen. In der weiteren Ausführung unterscheidet sich das neue System von den etablierten Verfahren wie z. B. (Low04) darin, dass die Informationen der Korrespondenzen nicht in der Domäne der Bilder bzw. Ansichten verarbeitet werden, sondern durch eine Überführung in lokale Lagehypothesen wesentlich universeller im 3D-Raum der Szene für eine robuste 6 DoF Lageauswertung herangezogen werden können (vgl. Abb. 1.1).

Dazu wird in dieser Arbeit ein neues (geometrisches) Modell entwickelt, welches 6 DoF Bewegungen bzw. Lageunterschiede von (starren) Objektausschnitten in der 3D-Szene approximativ mit linearen Deformationen des zugehörigen Erscheinungsbilds in der Bildebene in Beziehung setzt. Darauf aufbauend wird eine neue analytisch geschlossene Lösung zur Invertierung des Modells vorgestellt, welche es ermöglicht, aus einer beobachteten linearen Transformation in der Bildebene die zugehörige Bewegung in der 3D-Szene zu rekonstruieren. Damit kann aus einer *einzelnen* Regionenkorrespondenz zwischen einer eingelernten Modellansicht eines Objekts und einer Suchansicht der (lokale) 6 DoF Lageunterschied des Objekts zwischen Training und aktueller Ansicht bestimmt werden. Dieser ist aber aufgrund von Fehlern während der Korrespondenzfindung, der lokalen Informationsauswertung und der photometrischen und geometrischen Einflüsse noch nicht sehr robust.

Durch die universelle Repräsentation der Korrespondenzen als Lageunterschiede bzw. lokale 6 DoF Lagehypothesen im 3D-Raum der Szene wird durch ein Gruppierung ähnlicher Hypothesen ein neues und effizientes Verfahren zum Erkennen von Fehlkorrespondenzen, zur Segmentierung und zur

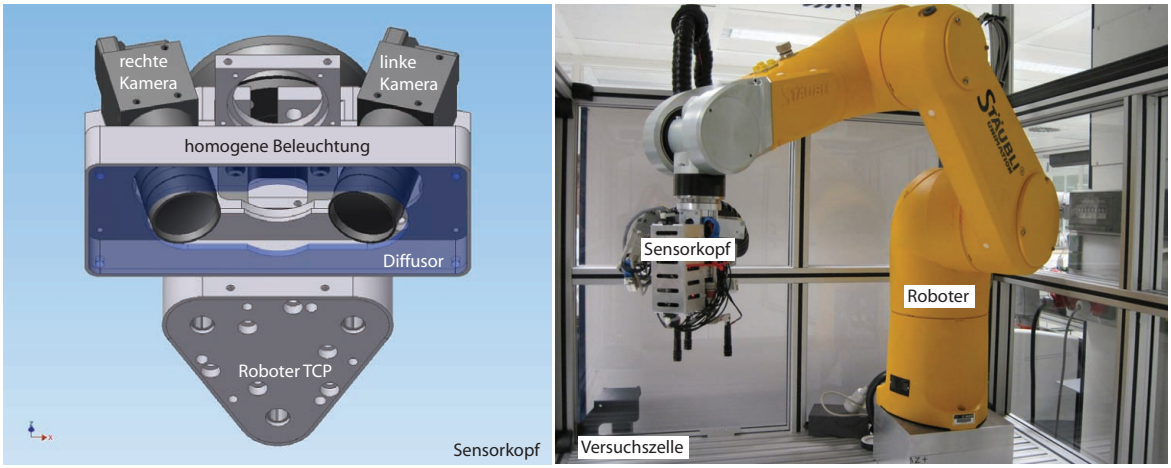

Abbildung 1.2: Die in dieser Arbeit eingesetzte Sensorik (links) und Aktorik (rechts). Der Sensorkopf besteht aus einer Stereokamera samt homogener Blitzbeleuchtung mit integriertem Diffusor und ist an dem Werkzeugträger (TCP) eines Industrieroboters befestigt. Der Roboter befindet sich in einer Sicherheitszelle, in der alle Versuche dieser Arbeit aufgebaut und durchgeführt wurden.

Fusion vieler Modellansichten, Suchansichten und unterschiedlicher salienter Strukturen ermöglicht. Ebenfalls lässt sich aus den Clustern der lokalen Hypothesen bzw. der räumlichen Komposition der zugehörigen Regionen auf robuste Weise eine exakte und vertrauenswürdige globale Lagehypothese einzelner Objekte in allen 6 Freiheitsgraden ableiten. Dazu werden in dieser Arbeit unterschiedliche Algorithmen vorgestellt, wovon die genauesten und robustesten die 2D-3D oder 3D-3D Punktkorrespondenzen zwischen den Regionenzentren der Modell- und Suchansichten nutzen, um eine globale Transformation zwischen den Ansichten zu schätzen.

Das offline Training eines Objektmodells erfolgt vollständig autonom mit den selben Algorithmen zum Finden salienter Regionen wie in der online Lokalisation, wobei die Ermittlung der räumlichen Beziehungen über einen kalibrierten Stereoaufbau erfolgt. Die unterschiedlichen einzulernenden Ansichten werden selbständig mithilfe eines Roboter angefahren und über dessen kinematisches Modell automatisch registriert. Die genaue Beschreibung der verwendeten Sensorik und Aktorik erfolgt im nächsten Abschnitt.

1.3 Sensorik und Aktorik

Die in dieser Arbeit verwendete Hardware besteht aus der in Abb. 1.2 rechts gezeigten Versuchszelle, in der verschiedene Szenarien aufgebaut werden können, einem Roboter innerhalb der Zelle und einem Sensorkopf, der an dem TCP[3] des Roboters montiert ist.

Bei dem Roboter handelt es sich um einen TX90 mit 6 Achsen von Stäubli[4], der aufgrund seiner 6 Freiheitsgrade den Sensorkopf präzise in beliebiger Lage innerhalb seines kugelförmigen Arbeitsbereichs von ca. 1 m Radius positionieren kann. Der Sensorkopf (Abb. 1.2 links) besteht aus zwei Kameras, einer homogenen Blitzbeleuchtung und einer Triggereinheit, die eine zeitsynchrone Aufnahme beider Kamerabilder und der Beleuchtung ermöglicht.

[3] engl.: tool center point

[4] Um den Kreis zu schließen soll nicht unerwähnt bleiben, dass Stäubli 1989 den Roboterpionier Unimation übernommen hat, der u. a. von G. Devol gegründet wurde und den in der Einleitung erwähnten ersten Industrieroboter Unimate hergestellt hat.

Die Beleuchtung besteht aus mehreren Hochleistungs-LED's, die für den Menschen unsichtbares Licht im infraroten Spektrum mit einer Wellenlänge von 850 nm emittieren. Sowohl die Belichtungszeit der Kameras als auch die Blitzdauer der LED's beträgt 2 ms[5] und ermöglicht damit eine effektive Unterdrückung von Fremdlicht. Allerdings kommt es trotz Diffusor aufgrund der punktförmigen Lichtquellen zu schwierigen Lichtverhältnissen mit Glanzpunkten, die die Auswertung der Bilder stark beeinflussen kann. Ein Vorteil einer mitfahrenden Beleuchtung ist dagegen, dass es keine durch den Roboterarm induzierten Abschattungsprobleme gibt.

Die beiden Kameras UI-2220-C-GL von U-Eye sind normale Farbkameras, die ohne IR-Sperrfilter betrieben werden und damit eine Empfindlichkeit im IR-Spektrum aufweisen. Für die Auswertung in dieser Arbeit wird das IR-beleuchtete Farbbild in ein Grauwertbild umgewandelt. Die Kameras haben einen $\frac{1}{2}$-Zoll Chip mit einer Auflösung von 768×576 pix und einer Pixelgröße von ca. $8.3\,\mu$m. Die Objektive haben eine Brennweite von 6 mm, eine Blende von 4 und sind so eingestellt, dass der Tiefenschärfebereich ca. $80 - 220$ mm beträgt.

Die beiden Kameras sind zueinander um 24° verkippt und bilden ein kalibriertes Stereosystem mit einer Basisbreite von ca. 80 mm. Der Aufbau ist so ausgelegt, dass ein Objekt innerhalb des Schärfebereichs im Zentrum beider Kamerabilder beobachtet werden kann.

1.4 Aufbau der Arbeit

Kapitel 2 beschreibt den Stand der Technik im Bereich Objekterkennung bzw. -lokalisation, der für die beschriebene Aufgabenstellung relevant ist. Die einzelnen Verfahren werden dabei klassifiziert und abschließend miteinander verglichen.

Kapitel 3 führt in das Gebiet der interessanten Regionen zum Auffinden robuster Korrespondenzen in Bildern ein und stellt insbesondere die Methoden zur Detektion salienter, kovarianter Regionen ausführlich vor. Dort ist auch in Abschnitt 3.1.2 die Herleitung des in dieser Arbeit zugrundeliegenden geometrischen Modells beschrieben.

Kapitel 4 enthält den konzeptionellen Entwurf des neu entwickelten Systems zur 6 DoF Lageschätzung mittels kovarianter Regionen aus Ansichten. Abschnitt 4.3.1 beschreibt dabei die Herleitung der analytisch geschlossenen Lösung des inversen geometrischen Modells, welche die Rekonstruktion lokaler Lagehypothesen aus affinen Verzerrungen des Erscheinungsbilds ermöglicht.

Kapitel 5 gibt einen kompakten Überblick über die in dieser Arbeit verwendeten und insbesondere neu erstellten Softwarekomponenten zur Realisierung des neuen Objektlokalisationssystems.

Kapitel 6 enthält alle Experimente dieser Arbeit, vom Auffinden der besten Parametereinstellungen der einzelnen Algorithmen bis zu einer ausführlichen Evaluierung des gesamten Systems bzgl. Laufzeit, Genauigkeit und Robustheit.

Kapitel 7 fasst abschließend die Arbeit zusammen, gibt Hinweise auf offene Punkte und weiterführende Arbeiten und einen generellen Ausblick der in dieser Arbeit behandelten Problemstellung.

Der Anhang enthält in Abschnitt A die in dieser Arbeit durchgängig verwendete Notation, deren Kenntnis insbesondere für die Kap. 2, 3 und 4 von Vorteil ist. Des Weiteren werden in Abschnitt B relevante mathematische Grundlagen beschrieben, die sich mit der Repräsentation von Position und Orientierung im 3D-Raum und Beweisen bei der Invertierung des geometrischen Modells beschäftigen. Zu guter Letzt wird in Abschnitt C eine Einführung in die Modellierung von Einzelkameras und Stereoaufbauten gegeben.

[5] Aufgrund der Augensicherheit bei unsichtbaren Strahlungen sind größere Blitzzeiten nicht möglich.

WIR STEHEN IMMER NOCH VOR
DER TÜR, HINTER DER DIE
GROSSEN ANTWORTEN WARTEN.

Arthur Miller

Kapitel 2

Stand der Technik im Bereich Objektlokalisation

Jedes kognitive System im Bereich von (teil-) autonomen Fahrzeugen, mobilen Plattformen oder manipulierenden Robotern in einer flexiblen Umgebung benötigt Informationen über seine Umwelt (im Weiteren als Szene bezeichnet) um seine Aufgabe erledigen zu können. Aufgrund des Preises, der Kompaktheit, der 2D-Erfassung, der Aufnahmegeschwindigkeit und nicht zuletzt der großen Flexibilität sind Kameras die am häufigsten eingesetzten Sensoren im wissenschaftlichen Umfeld. Dafür sind allerdings Methoden notwendig, die aus der großen Menge an rohen Intensitätsdaten aufgabenspezifische Informationen extrahieren und auf einem abstrakteren Level zur Verfügung stellen. Es ist daher nicht verwunderlich, das eines der meist beachtetsten Themen im Bereich der Bildauswertung die Objekterkennung ist. Je nach Aufgabe unterscheidet man zwischen der *Objektdetektion*, deren Ziel nur eine Anwesenheitsprüfung ist und der *Objektlokalisation*, deren Ergebnis ebenfalls die exakte Lage in bis zu 6 Dimensionen (3 translatorische und 3 rotatorische Freiheitsgrade) beinhaltet.

Während der Mensch diese Probleme selbst unter schwierigen Bedingungen meist unterbewusst meistert, zeigt sich deren ganze Komplexität bei dem Versuch visuelle Kognition auf einer Maschine zu realisieren. Die größten Herausforderungen sind Unsicherheit, lokale Störungen und Mehrdeutigkeiten in den Daten sowie Szenenvielfalt, Beleuchtungseinflüsse, Verdeckungen und die Menge der Daten bzw. der große Lösungsraum. Allerdings eröffnen sich der Maschine Möglichkeiten im Bereich der absoluten Messung von Größen, wie der Position oder Orientierung von Objekten bei der das menschliche Wahrnehmungssystem ab einer gewissen geforderten Genauigkeit versagt.

Wie in Abb. 2.1 schematisch dargestellt, lässt sich jedes Objekterkennungssystem grob in drei Schritte zerlegen, wobei oftmals mehrere Schritte von einem Algorithmus behandelt werden. Innerhalb des ersten Schrittes werden die Eingangsdaten in relevante Objektinformationen, Hintergrund und Störungen zerteilt. Der Sinn ist eine Reduktion der Information für die nachfolgenden Schritte. Dies lässt sich entweder durch eine *Segmentierung* der Daten oder eine *ausführliche Suche* innerhalb der Daten erreichen. Bei der Segmentierung wird meist unter bestimmten Annahmen auf intelligente Weise Objekt und Hintergrund voneinander getrennt. Bei beliebigen Szeneninhalt ist dieses Problem allerdings meist genauso komplex wie die Objekterkennung an sich. Bei der ausführlichen Suche werden dagegen viele Möglichkeiten der Datenteilung ausprobiert. Bereiche ohne Objekt müssen dann von dem nachfolgenden Schritt zurückgewiesen werden. Dieser wird daher auch als *Verifikation* bzw. *Groblageschätzung* bezeichnet. Er überprüft ob sich im übergebenen Datenbereich ein gesuchtes Objekt befindet und schätzt dafür notwendigerweise auch gleichzeitig eine erste Lage des Objekts. Sie ist normalerweise noch sehr ungenau, schränkt den Bereich im Idealfall aber so weit ein, dass die abschließende, meist iterative *Feinlageschätzung* zur exakten Position und Orientierung konvergieren kann. Dieser abschließende Schritt ist nur für die Objektlokalisation notwendig.

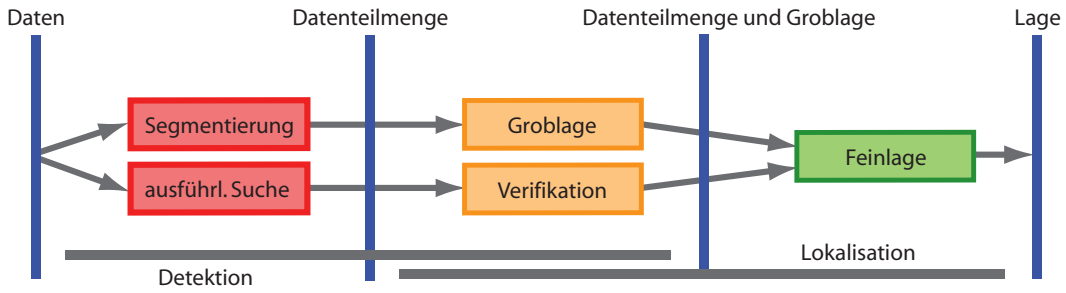

Abbildung 2.1: Teilschritte der Objekterkennung. Während für die Objektdetektion eine Segmentierung bzw. ausführliche Suche mit anschließender Groblageschätzung bzw. Verifikation genügt, benötigt die Lokalisation normalerweise noch eine Feinlageschätzung.

Eine mögliche Taxonomie in der Objekterkennung lässt sich anhand des Wissens angeben, welches verwendet wird um innerhalb der Eingangsdaten nach Objektausprägungen zu suchen. Weiß man so gut wie nichts über das Aussehen oder die geometrische Beschaffung der Objekte, kann man meist nur Expertenwissen wie z.B. einen Satz von Regeln oder Annahmen angeben. Die dominierende Anwendung ist die Hinderniserkennung im Automobilbereich, bei mobilen Plattformen oder der Pfadplanung von stationären Industrierobotern. Meist werden Häufungen von Abweichungen bezüglich dem angenommenen Normalzustand im Bereich der Geschwindigkeit (NA98), der Farbe (UN00) oder des Raumes (LAT02) herangezogen und bei entsprechender Ausprägung als Hindernis erkannt. Sind zumindest die Objektklassen bekannt, werden meist klassische Methoden der Mustererkennung angewendet. Oftmals findet im ersten Schritt eine ausführliche Suche in Form eines gleitenden Fensters über den Bildbereich statt, dessen Inhalt dann von einem Klassifikator (vgl. hierzu z.B. (DHS00)) bewertet wird. Das Objekt- bzw. Klassenmodel wird in einem (meist) überwachten Schritt trainiert und zeichnet sich durch eine große Flexibilität bzgl. bestimmter Transformationen aus. Im Bereich der Fahrzeug- und Gesichtserkennung sind dies die hohe innere Klassenvariabilität, im Bereich der Objektdetektion oftmals Änderungen durch unterschiedliche 3D-Orientierungen. (SK00) trainiert z.B. unterschiedliche Klassifikatoren für wenige grobe Lagen um damit über deren Flexibilität dennoch das gesamte Lage- und Klassenspekturm während der Detektion zu erfassen. Um aber den gesamten Datenbereich abzuarbeiten, müssen diese komplexen Klassifikatoren für jeden gleitenden Fensterausschnitt aufgerufen werden, was für eine echtzeitfähige Anwendung nicht praktikabel ist. (VJ01) nutzt daher eine Kaskade von einfachen, aber schnellen Klassifikatoren, um viele Fensterausschnitte frühzeitig zurückweisen zu können und damit die Laufzeit deutlich zu erhöhen. In der Kombination erreichen die Klassifikatoren dennoch eine hohe Erkennungsrate und lassen daher diesen Ansatz zu einem der beliebtesten Gesichts-, Schilder- und Fahrzeugerkennungssysteme in einer Echtzeit notwendigen Umgebung werden. Anstatt einzelne Lagen von Objektklassen zu trainieren, lernt (LSS08) aus konkreten 3D-Repräsentationen von Fahrzeugen ein lokales, merkmalsbasiertes 3D-Modell. Sowohl die Segmentierung als auch die Klassifikation erfolgt über die Merkmalszuordnung, wobei über deren Lage ebenfalls eine grobe Lageschätzung zurückgegeben werden kann.

Für eine genaue Lokalisation benötigt man im Allgemeinen nicht nur eine Objektklasse sondern ein konkretes, starres Objekt mit festen Bezügen. Zweierlei Repräsentationen dominieren in der Wissenschaft, das ansichtsbasierte und geometriebasierte Modell. Ersteres verwendet das Erscheinungsbild des Objektes und codiert Texturinformationen, wohingegen Letzteres die geometrische Struktur verwertet. Die Abschnitte 2.1 und 2.2 stellen die relevantesten Ansätze auf Basis ihrer dominierenden Repräsentation vor. Weiterhin erfolgt darunter eine grobe Aufgliederung in globale und lokale Verfahren. Globale Verfahren behandeln die Objekte in ihrer Gesamtheit, sind daher exakter in der

Lageschätzung und können lokale Mehrdeutigkeiten leicht auflösen. Ihr großer Nachteil ist aber, dass sie meist auf eine genaue Segmentierung angewiesen sind und daher anfällig bzgl. Verdeckungen, Störungen und Hintergrundschwankungen sind. Lokale Verfahren repräsentieren das Objekt dagegen als eine Menge von salienten, d. h. herausragenden Merkmalen und nutzen meist Gruppen davon zur Objekterkennung. Sie sind anfälliger gegenüber verrauschten Daten und daher normalerweise weniger exakt. Dafür sind sie aber wesentlich robuster gegenüber lokalen Störungen, Hintergrundänderungen oder komplexen Szenen. Die Verfahren werden soweit möglich bzgl. der Aufgabe und Randbedingungen aus Abschnitt 1.1 beschrieben und bewertet. Ein besonderes Augenmerk wird auf einen möglichen Trainingsschritt sowie die Praxistauglichkeit gelegt.

Unabhängig von der Objektrepräsentation unterscheidet man zwischen *statischer* und *aktiver* Objekterkennung. In beiden Fällen können zwar Beobachtungen der Objekte von verschiedenen Sensorlagen verarbeitet werden, allerdings werden diese im statischen Fall starr festgelegt, wohingegen bei der aktiven Objekterkennung unter Berücksichtigung vorangegangener Ergebnisse und der verwendeten Objektmodelle eine möglichst optimale neue Sensorlage ermittelt wird. Die Rückkopplung wird meist visuell durch einen Roboter mit einer darauf befestigen (Stereo-) Kamera realisiert. Damit können Mehrdeutigkeiten innerhalb der Szene oder der Objektmodelle leichter aufgelöst werden und Unsicherheiten in der Sensorkalibrierung, in der Roboterkinematik und im Objektmodell einfacher kompensiert werden. Einen guten Überblick solcher aktiver Verfahren gibt (RCB04), hier werden sie, da sie meist ähnliche Ideen und Objektrepräsentationen verwenden, in den entsprechenden Unterabschnitten erwähnt.

Eine ausführliche Zusammenfassung in Abschnitt 2.3 mit einer Bewertung und einem Vergleich aller Ansätze schließt dieses Kapitel ab.

2.1 Ansichtsbasierte Techniken

Ansichtsbasierte Techniken repräsentieren Objekte in Form von einzelnen Bildern, den Objektansichten. Während der Lokalisation müssen diese einem aktuellen Bild zugeordnet werden und daraus die Lage des Objekts abgeleitet werden. Da die Bilder keine reine Objektinformation beinhalten, sondern eine Kombination von relevanter Texturinformation und irrelevanter Beleuchtung darstellen, muss dieser Fakt bei der Zuordnung explizit berücksichtigt werden. Die Ansichten werden daher meist auf eine robuste oder invariante Weise codiert (vgl. dazu auch Abschnitt 3.1.1), wobei globale Methoden dafür das gesamte Bild verwenden, lokale Methoden nur einzelne Bildausschnitte.

2.1.1 Globale Ansätze

Es besteht eine Vielzahl von Möglichkeiten um Ansichten global zu beschreiben und zuzuordnen. Im Bereich der Objektlokalisation sind die Korrelations-, Histogramm- und Momentenbasierten Techniken die wichtigsten, die zusammen mit ausgewählten anderen Techniken in den folgenden Unterabschnitten beschrieben werden.

2.1.1.1 Korrelationsbasierte Techniken

Korrelationsbasierte Ansätze repräsentieren die Objektansichten in ihren ursprünglichen Intensitätsbildern oder den davon abgeleiteten Gradientenbildern. Das für die Zuordnung notwendige Ähnlichkeitsmaß wird über Korrelationsmethoden definiert.

Eines der ersten Verfahren wird in (MN95) vorgestellt. Die von einer Kamera aufgenommene Ansicht eines Objektes wird als eine Kombination der intrinsischen Eigenschaften Form und Reflexion und der extrinsischen Beleuchtung angesehen. Daher werden von beliebigen 3D-Objekten

mit Hilfe eines computerangesteuerten Drehtellers und einer computergesteuerten Beleuchtungseinheit automatisch verschiedene Grauwertbilder mit variierender 1D-Orientierung als auch variierender Beleuchtung aufgenommen. Der Bildersatz wird auf eine einheitliche Größe von 128×128 Pixeln und einheitliche Beleuchtung mit selben Intensitäts-Mittelwert und Kontrast normiert und als 2^{14}-dimensionaler Vektor dargestellt. Zur Datenreduktion wird der enorme Datensatz mit Hilfe der *Hauptkomponentenanalyse* (PCA[1]) (Dun89) in den Eigenraum transformiert und auf die k signifikantesten Dimensionen reduziert. Nach (MN95) wird zwar der gesamte Raum für eine exakte Objektrepräsentation benötigt, es reicht aber schon eine niedrige Dimensionalität von $k = 20$ aus, um die charakteristischen Eigenschaften eines Objektes zu modellieren. Weiterhin werden die Ansichtsvektoren in den reduzierten Eigenraum projiziert und mittels Spline-Interpolation aus den diskreten Punkten eine durch den Orientierungswinkel des Drehtellers und der Beleuchtungsstärke parametrisierte kontinuierliche Mannigfaltigkeit gebildet.

(MN95) unterscheidet den *universellen Eigenraum*, der aus Ansichten aller zu erkennenden Objekte gebildet wird und den *Objekt-Eigenraum*, der nur Ansichten eines Objektes enthält. In den universellen Eigenraum wird für jedes Objekt eine eigene Mannigfaltigkeit eingebettet und zur Objektidentifizierung genutzt, während die einzelne Mannigfaltigkeit des Objekt-Eigenraums für die Orientierungsschätzung herangezogen wird. Während der Objekterkennung wird ein Objektkandidat über eine einfache Schwellwertoperation von dem schwarzen Hintergrund segmentiert, das ausgeschnittene Bild in Größe und Intensität normiert und in den universellen Eigenraum transformiert. Dort wird unter Verwendung der euklidischen Distanz die nächstgelegene Mannigfaltigkeit ermittelt und darüber der Objekttyp detektiert. Nach (MN95) entspricht dies einer Approximation der Korrelation des aktuellen Bildausschnitts mit den Modellansichten. Die 1D-Lokalisation erfolgt dann im typspezifischen Objekt-Eigenraum durch die Parameterwerte der Projektion des aktuellen Ansichtsvektors auf die eingebettete Mannigfaltigkeit. Durch deren interpolierten kontinuierlichen Verlauf lassen sich selbst Zwischenansichten robust behandeln.

In (NNM96) wird das System auf Farbbilder erweitert und mit 100 verschiedenen 3D-Objekten gleichzeitig getestet. Jeder der drei Farbkanäle wird getrennt voneinander behandelt, nur die Intensitäts-Normierung erfolgt so, dass das spektrale Verhältnis der Farben beibehalten wird. Überschreitet eine Objekthypothese in einem der Farbkanäle eine festgelegte Distanz zu den universellen Mannigfaltigkeiten, wird das Objekt als unbekannt zurückgewiesen. Im Training wurden pro Objekt in 7.5° Schritten jeweils 5 Ansichten mit unterschiedlicher Beleuchtung aufgenommen und alle $40 \cdot 5 \cdot 100 = 20000$ Ansichten innerhalb eines Tages verarbeitet. Das System erreichte mit 1000 Testansichten eine Erkennungsrate von 100% bei einem mittleren Fehler von 2.02° und einer Standardabweichung von 1.67° in der Lokalisierung. Die durchschnittliche Verarbeitungszeit lag auf einem Dec Alpha 3600 bei 0.7 s pro Bild inklusive Segmentierung.

Teilen sich die zu erkennenden Objekte viele ähnliche Ansichten, lassen sich einzelne Ansichten nicht mehr eindeutig zu einer der Objekt-Mannigfaltigkeiten im universellen Eigenraum zuordnen. Daher verwendet (BPPP98) ein aktives System, um die Kamera so zu positionieren, das die neue Ansicht in einen Bereich des Eigenraums fällt, wo die konkurrierenden Mannigfaltigkeiten einen möglichst großen Abstand zueinander haben. Neben einer robusten Unterscheidung einseitig verschieden markierter Objekte, kann dadurch die verwendete Dimension des Eigenraums deutlich reduziert werden. Im Schnitt waren 2.6 Ansichten nötig um bei einem Eigenraum mit drei Dimensionen eine eindeutige Detektion und Lokalisation zu erreichen.

Für die Lokalisation von einfarbigen Objekten wird das (statische) System in (AAD06) auf 3 Freiheitsgrade in der Orientierung erweitert. Der komplexere Einlernvorgang wird mit Hilfe von virtu-

[1] engl.: Principal component analysis

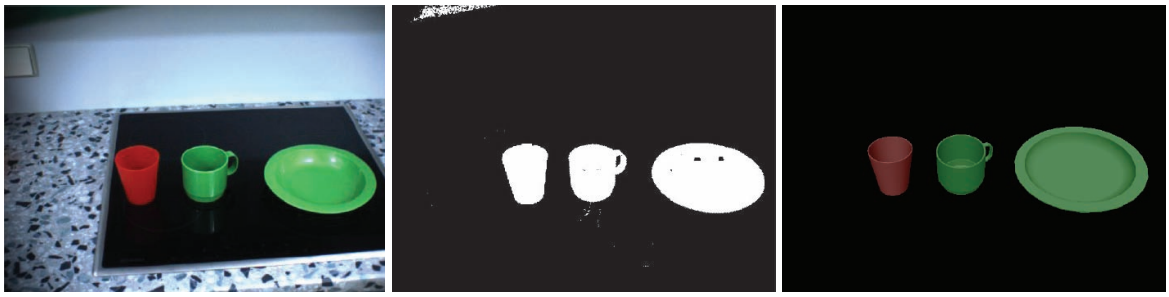

Abbildung 2.2: Farbsegmentierung (Mitte) und Ergebnis der Lokalisation (rechts) mittels der korrelationsbasierten Methode aus (AAD06).

ellen Ansichten eines CAD-Modells durchgeführt. Anstatt die Beleuchtung zu variieren wird mit Gradientenbildern gearbeitet, die vor allem die weniger anfällige geometrische Struktur des Objektes erfassen. Mittels einer Stereokamera werden ebenfalls die 3 translatorischen Freiheitsgrade ermittelt und damit eine vollwertige 6D-Lokalisation realisiert. Über die Schwerpunkte der über die Farbe segmentierten Regionen in den Stereobildern wird durch Triangulation approximativ auf den 3D-Masseschwerpunkts des Objekts geschlossen und über einen beim Training berechneten Offset der Objektursprung im Referenz-Kamerasystem geschätzt. Die Orientierungsschätzung erfolgt unabhängig davon auf dem Monobild der Referenzkamera. Im Gegensatz zu (MN95) wird keine kontinuierliche Mannigfaltigkeit im Objekt-Eigenraum geschätzt, sondern über eine Nächste-Nachbar (NN) Suche mit Hilfe der euklidischen Distanz das ähnlichste Bild bzgl. der approximierten Korrelation ermittelt und deren hinterlegte Orientierung zurückgegeben. Da sich bei unterschiedlichen Translationen selbst bei gleicher Orientierung die Ansicht von (nicht planaren) 3D-Objekten leicht ändert sowie aufgrund der approximativen Schwerpunktberechnung, sind Korrekturen der Lage notwendig. Die Orientierungskorrektur lässt sich analytisch angeben, die Translationskorrektur muss allerdings über simulierte Ansichten mittels des CAD-Modells berechnet werden. (Aza08) beschreibt die Details und gibt auch ein iteratives Schema zur Ermittlung der genauen Lage an. Das System wertet ein Bild inklusive Segmentierung und iterativer Korrektur auf einem 3 GHz Single-Core in $30-40$ ms für ein einzelnes Objekt mit ca. 10000 Ansichten aus. Diese Anzahl ist in der Datenbank nötig um eine ausreichende Genauigkeit zum Greifen mit einem humanoiden Roboter zu erhalten, da im Gegensatz zu (MN95) durch die NN-Suche zwischen den Ansichten nicht interpoliert werden kann. Abb. 2.2 zeigt ein Beispiel der Farbsegmentierung und des Lokalisationsergebnisses.

2.1.1.2 Histogrammbasierte Techniken

(SB91) verwendet erstmals Farbhistogramme zur Repräsentation von Objektansichten. Ein Histogramm $\mathrm{h}(c)$ gibt die absolute Häufigkeit einer Farbe c innerhalb einer Objektansicht an und ist unabhängig von Translation und Rotation in der Kameraebene. Das Ähnlichkeitsmaß zwischen zwei Ansichten wird über die *Histogrammschneidung* [2] definiert:

$$d = n^{-1} \sum_{c} \min\left(\mathrm{h}_1(c), \mathrm{h}_2(c)\right) \tag{2.1}$$

Mittels des Normierungsfaktors $n = \sum_c \mathrm{h}_2(c)$ wird die Distanz d auf das Intervall $[0..1]$ abgebildet. d besteht aus der aufsummierten minimalen Übereinstimmung beider Histogramme und ist robust gegenüber kleinen Verdeckungen, geringen Blickwinkeländerungen und variierendem Hintergrund ins-

[2] engl.: histogram intersection

Abbildung 2.3: Beispiel der Objektdetektion mittels des Verfahrens aus (CK99). Links ist ein erfolgreich gefundenes Objekt in einer komplexen Szene eingezeichnet, dass durch CCH's der rechts abgebildeten Ansichten repräsentiert wird. Aus (CK99) entnommen.

besondere der Ansicht 1. h_1 repräsentiert daher auch eine unbekannte Ansicht, während h_2 für die hinterlegten segmentierten Modellansichten benutzt wird. Die verwendeten Farbbereiche $c = (rg, by, wb)$ entstehen aus der Diskretisierung eines dem Menschen nachempfundenen Farbraums, der aus dem roten r, grünen g und blauen b Farbkanal durch die Transformation $rg = r - g$, $by = 2b - r - g$ und $wb = r + g + b$ gebildet wird. rg wird in 8 Stufen quantisiert, by und wb in jeweils 16, was zu 2^{12} Einträgen innerhalb des Farbhistogramms führt. Die Histogrammschneidung ist allerdings sehr anfällig gegenüber Beleuchtungsschwankungen. (SB91) schlägt dafür eine einfache Normierung von c mittels der absoluten Intensität wb vor, um ein Histogramm mit konstanten Farbwerten zu erhalten. Alternativ dazu verwendet (FF95) das Verhältnis der Farbwerte benachbarter Pixel.

Der größte Nachteil von Farbhistogrammen ist allerdings, dass sie keinerlei Informationen über die räumliche Farbverteilung codieren. Verschiedene Erweiterungen werden daher vorgeschlagen. (SD96) verwendet Histogramme von 5 überlappenden Fuzzyregionen der Ansicht und speichert jeweils die ersten 3 Momente der ermittelten Häufigkeitsverteilungen. (HKM$^+$99) erweitert das Farbhistogramm zum *Korrelogramm* $h(c_1, c_2, k)$, welches die Häufigkeit einer Farbe c_1 in einem Bildabstand k zu einer gegebenen Farbe c_2 angibt. Einen ähnlichen Ansatz verfolgen die *Color Cooccurrence Histograms* (CCH) $h(c_1, c_2, \Delta x, \Delta y)$ für einen beliebigen Translationsvektor $(\Delta x, \Delta y)^{\mathrm{T}}$ (CK99).

CCH's werden in (CK99) erfolgreich für die Detektion von Objekten eingesetzt (siehe Bsp. in Abb. 2.3). Zur Reduktion der Datenmenge wird die Translation auf die Distanz $k = \sqrt{\Delta x^2 + \Delta y^2}$ beschränkt. Während im Korrelogramm die Maximumsnorm L_∞ zur Berechnung von k benutzt wird, verwendet (CK99) die L_2-Norm um eine rotationsinvariante Repräsentation beizubehalten. Alternativ kann anstatt der Distanz auch der Winkel des Translationsvektor herangezogen werden (Winkel-CCH's). Es wird der RGB-Farbraum verwendet und mittels k-Means (KMN$^+$02) über alle Modellansichten in n_c Bereiche quantisiert. Für k werden n_k gleichmäßig verteilte Bereiche verwendet. Abgeleitet von einem probabilistischen Fehlermodell werden die optimalen Parameter auf $n_c = 8$ und $n_d = 12$ festgelegt. Zum Vergleichen zweier CCH's wird die erweiterte Form der Histogrammschneidung verwendet:

$$d = n^{-1} \sum_{c_1} \sum_{c_2} \sum_{k} \min\left(\mathrm{h}_1(c_1, c_2, k), \mathrm{h}_2(c_1, c_2, k)\right)$$

Dieser Ansatz wird in (EHK03) zu einem vollwertigen 1D-Lokalisationssystem ausgebaut. Um den Effekt von Beleuchtungseinflüssen zu vermindern und unabhängig von der zugrundeliegenden Regionengröße zu werden, werden die CCH's normiert. Mittels eines gleitenden Fensters wird zuerst

eine grobe Likelihood-Karte des Bildes berechnet. Dazu wird nur jeweils eine Front- und Rückansicht der Objekte verwendet. Die groben Maxima dieser Karte werden dann iterativ durch Anpassung der Fenstergröße verfeinert und die vielversprechendsten Ausschnitte zusammen mit dem Objekttyp der Lokalisation übergeben. Für eine grobe Orientierungsschätzung von Objekten auf einem Tisch werden in einem Trainingsschritt jeweils ca. 50, um die Oberflächennormale des Tisches gedrehte Ansichten aufgenommen und mittels eines merkmalsbasierten Ansatzes der zugehörige Winkel auf ca. 5° genau geschätzt. Die Zuordnung der Testausschnitte zu den Modellansichten erfolgt mit der erweiterten Histogrammschneidung. Eine Interpolation des Winkels erfolgt durch eine Faltung, eines mit der Ähnlichkeit d gewichteten Gaußkernels in dem Ring der Winkel über alle Modellansichten des Objekts. Das Maximum wird dann als grobe 1D-Orientierung mit einem mittleren Fehler von ca. 14° zurückgegeben. Als Konfidenzmaß wird das Verhältnis des größten und zweitgrößten Maxima abzüglich des Mittelwerts verwendet. Ist es nahe bei 1, ergibt sich eine globale Mehrdeutigkeit der CCH's und eine alternative Repräsentation (z. B. Winkel-CCH's) muss für die Zuordnung des Testausschnitts herangezogen werden. Die grobe Lage wird abschließend mittels einer iterativen, kantenbasierten Feinregistrierung unter Verwendung eines geometrischen Kantenmodells in allen 6 Freiheitsgraden geschätzt. Details dazu finden sich in (EKH05). Die Detektion benötigt ca. 6 s auf einem 1.8 GHz Rechner und die anschließende Grob-Lokalisation ca. 0.7 s.

2.1.1.3 Momentenbasierte Techniken

Während Korrelations- und reine Histogrammbasierte Techniken durch Repräsentation von vollständiger und keinerlei räumlicher Information zwei entgegengesetzte Extremstellungung bei der Ansichtscodierung einnehmen, liegen die momentenbasierte Techniken dazwischen. Momente werden in einem breiten Spektrum in der Bildverarbeitung eingesetzt [3] und codieren auf kompakte Weise sowohl Intensität als auch deren räumliche Verteilung. Ihr großer Vorteil ist, dass daraus invariante Größen bzgl. bestimmter Transformationen abgeleitet werden können und daraus eine flexiblere Zuordnung der Ansichten ermöglicht wird. Wichtige Momente sind die translations- und rotationsinvarianten Alt- (Alt62) und Zernike-Momente (Zer34), die zusätzlich skalierungsinvarianten Hu-Momente (Hu62) und deren affin-invarianten Erweiterungen (Rei91). Ebenfalls ist eine Invarianz bzgl. linearer Beleuchtungsschwankungen (vgl. Abschnitt 3.1.1) möglich (Rei91).

(KH90) verwendet z. B. Zernike-Momente zur rotations- und translationsinvarianten Bilderkennung. (ANR98) erweitert diesen Ansatz und schätzt über die Phase zusätzlich die Rotationsänderung innerhalb der zugeordneten Bilder von industriellen Objekten.

Zernike-Momente werden ebenfalls in (CG00) zur Klassifikation und 3D-Orientierungsschätzung verschiedener Flugzeugtypen verwendet. Mittels einer virtuellen Kamera, die sich auf einer Sphäre um virtuelle 3D-Modelle bewegt, werden synthetische Ansichten mit unterschiedlichen 2D-Orientierungen erzeugt. Diese werden binarisiert, in der Größe normiert und die gewonnene Objektsilhouette mittels bis zu 49 verschiedener komplexer Zernike-Momenten codiert. Der Betrag dieser Momente bildet dann einen globalen Merkmalsvektor der invariant bzgl. des letzten Orientierungsfreiheitsgrads, der Rotation in der Kameraebene ist. Anhand einer nichtlinearen Abbildung werden die globalen Merkmalsvektoren mit den 2D-Orientierungen verknüpft und damit auf eine sphärische 2D-Mannigfaltigkeit abgebildet. Aus diesen diskreten Punkten wird mittels *Probabilistic Principal Surfaces* (HS88b; CG01) für jedes Objekt eine kontinuierliche, sphärische Mannigfaltigkeit gebildet und als Modell aller möglichen Ansichten des jeweiligen Flugzeugtyps verwendet. Die Klassifikation einer neuen segmentierten Silhouette erfolgt dann wie in (MN95) durch den kleinsten Abstand aller Mannigfaltigkeiten und die Bestimmung der 2D-Orientierung durch Projektion auf die entsprechen-

[3] vgl. z. B. Abschnitt 3.2.1.4 zur Berechnung von Vorzugsrichtungen in Bildausschnitten

de Sphäre. Der letzte Freiheitsgrad in der Ansichtsebene lässt sich dann über den Phasenunterschied der komplexen Momente zwischen der Objekt- und interpolierten Modellansicht bestimmen. Für das Training des Modells und den Testdatensatz wurden zweimal 684 zueinander versetzte virtuelle Ansichten auf der Sphäre je Objekt erzeugt. Der Fehler der Lokalisation lag bei 55% der Testansichten unter 20° und bei 70% unter 40°. Eine Beleuchtungskompensation sowie das Laufzeitverhalten wurden nicht beschrieben.

Für ein 3D-Objekterkennungssystem nutzt (MOA04) die sieben translations-, rotations- und skalierungsinvarianten Hu-Momente. Auf einem Drehteller werden beliebig geformte Objekte von drei Kameras auf einem homogenen Hintergrund beobachtet und in 5°-Schritten aufgenommen. Die Hälfte der Ansichtentrippels werden genutzt um ein neuronales Netz zu trainieren. Dieses hat als Eingangsknoten die Hu-Momente aller drei Ansichten und für jedes Objekt einen Ausgabeknoten. Die Struktur des Netzes ist entweder ein *multi-layered perceptron* (MLP) Netzwerk oder ein *hybrid multi-layered perceptron* (HMLP) Netzwerk. Zum Testen wurde die um jeweils 5° verschobene restliche Hälfte der Ansichtentrippels genutzt. Das System erreichte mit HMLP eine Erkennungsrate von 100% bei Verwendung der drei weniger rauschanfälligen Momente niedrigster Ordnung. Das neuronale Netz wurde allerdings nur auf eine binäre Entscheidung trainiert, eine Erweiterung zum Schätzen des 1D-Lokalisationswinkels wurde nicht untersucht. Ebenfalls wurde keine Laufzeit des Systems angegeben.

2.1.1.4 Andere Techniken

Neben den in den vorangegangenen drei Abschnitten beschriebenen Objektrepräsentationen gibt es noch eine Vielzahl weiterer Möglichkeiten um Ansichten global zu beschreiben und um Zuordnungen, Interpolation oder Ähnlichkeitsmaße zwischen Ansichten zu realisieren. Prinzipiell können alle Mustererkennungsalgorithmen verwendet werden, die bestimmte Muster in Bildern wiederfinden können. Sie sind sehr flexibel und können bei einem entsprechenden Trainingsdatensatz gut mit Beleuchtungsvariationen umgehen, müssen allerdings für jede zu unterscheidende Ansicht normalerweise einen separaten Klassifikator trainieren. Dies führt bei über hundert Ansichten, die oft nötig sind um ein Objekt von allen Seiten robust zu beschreiben, zur gleichen Anzahl parallel auszuführender Klassifikatoren pro Eingabebild und daher zu unpraktikablen Laufzeiten. Ebenfalls ist es schwierig zwischen Modellansichten zu interpolieren, so dass der Abstand zwischen den eingelernten Ansichten die Lokalisationsgenauigkeit festlegt. Eine Verwendung von üblichen Mustererkennungsalgorithmen wie Neuronale Netze, Support-Vector-Machines oder ähnlichem (siehe dazu auch (DHS00)) zur vollständigen Lageschätzung bzw. Regression zwischen vielen Ansichten in realen Systemen ist dem Autor nicht bekannt.

Eine einfachere Möglichkeit verwendet (SVLP98) in einem aktiven System bestehend aus einer Mono-Kamera in Verbindung mit einem Industrieroboter. Objekte werden durch ihre Silhouette in eingelernten Ansichten in Form einer Sequenz von Bildpunkten repräsentiert. Mittels eines physikalischen motivierten Energie-Ansatzes wird ein Morphalgorithmus definiert, der zwei Punktsequenzen, d. h. zwei Silhouetten ineinander überführen kann. Damit kann eine Folge von virtuellen Zwischenansichten inklusive zugehöriger Objektlageschätzung erzeugt werden, mit deren Hilfe der Roboter aktiv von einer unbekannten Lage in eine bekannte Modell-Lage überführt werden kann. Das für die Zuordnung zur ähnlichsten Modellansicht notwendige Maß wird über die Energie definiert, die notwendig ist, um zwei Ansichten ineinander zu morphen. Das System wird verwendet, um sowohl planare als auch beliebig anders geformte 3D-Objekte aktiv in eine Referenzansicht zu überführen. Die Performance des Systems in Form von Genauigkeit und Laufzeit wird in (SVLP98) nicht angegeben.

2.1.2 Lokale Ansätze

Im Gegensatz zu globalen Verfahren werden bei den lokalen Ansätzen nur kleine, saliente Ausschnitte der Objektansichten betrachtet und zur Lokalisation herangezogen. Der Vorteil ist neben einer Robustheit gegenüber lokalen Störungen und Verdeckungen, dass die Veränderung der Objektansichten durch eine Blickwinkeländerung zwar global betrachtet komplex sind, auf der lokalen Ebene aber meist mit einfachen Transformationen approximiert werden können. In fast allen Fällen wird dazu eine *lokale Ebenen-Annahme* der entsprechenden Objektbereiche getroffen, was dazu führt, dass man zwei Ansichten des selben lokalen Ausschnittes im Bild durch eine *Homographie* mit acht Freiheitsgraden beschreiben kann. Nimmt man weiterhin anstatt einer perspektivischen Projektion lokal eine *parallele Projektion* an, lässt sich der Bereich durch eine *affine Transformation* mit sechs Freiheitsgraden beschreiben. Lässt man ebenfalls keine Verkippungen des Objektbereichs aus der Kameraebene zu, reduziert sich die Transformation auf eine *Ähnlichkeitsabbildung* mit vier Freiheitsgraden.

Lokale Verfahren haben im Kern meist dieselbe Struktur, deren Details in Kapitel 3 behandelt wird. Ein *Detektor* findet in Bildern saliente Bereiche in Form von Punkten oder Regionen, die meist robust gegenüber variierenden Beleuchtungsverhältnissen und kovariant, d.h. angepasst an die lokale Bildstruktur bzgl. der geometrischen Veränderungen ermittelt werden. Bekannte Punktdetektoren sind z.B. die *Harris-Ecken* (HS88a) oder *Shi-Thomasi-Merkmale* (ST94), bekannte ähnlichkovariante Regionendetektoren die *SIFT-* (Low04) oder *SURF-Regionen* (BTVG06) und bekannte affin-kovariante Detektoren die *MSER-Regionen* (MCUP04). Einen genaueren und vollständigeren Überblick gibt Abschnitt 3.2 sowie (MO04; MTS$^+$05; TM08).

Die lokalen Bildinformationen in einer Umgebung der gefundenen Punkte oder Regionen werden dann mittels eines *Deskriptors* auf robuste oder invariante Weise in einem *Merkmalsvektor* codiert, wobei Invarianz gegenüber den geometrischen Veränderungen meist durch die Abbildung kovarianter Regionen auf eine kanonische Form realisiert wird. Für weitere Informationen siehe Abschnitt 3.3 sowie (MS05; TW09).

In einem weiteren Schritt müssen *Matcher* auf effiziente Weise Korrespondenzen zwischen verschiedenen lokalen Merkmalen finden. Dafür muss ein Ähnlichkeitsmaß auf dem Raum der Merkmalsvektoren definiert sein und eine darauf angepasste Datenstruktur zur schnellen Indizierung großer Mengen von Merkmalen. Details werden in Abschnitt 3.4 behandelt.

Die gefundenen Korrespondenzen werden dann je nach Verfahren auf verschiedene Weise weiterverarbeitet. Diese weiterführenden Ansätze zur Objektdetektion werden im weiteren Verlauf vorgestellt und nach Informationsgehalt, den sie aus den Korrespondenzen ziehen, klassifiziert. Die geringste Information verwenden Verfahren in Abschnitt 2.1.2.1, da sie nur die Existenz bzw. Translation der korrespondierenden Punkte betrachten. Einen größeren Informationsgehalt verwerten Ansätze in Abschnitt 2.1.2.2 durch eine zusätzliche Betrachtung des Rotations- und Skalenunterschieds, aber erst die Verfahren aus Abschnitt 2.1.2.3 greifen auf alle sechs Freiheitsgrade des affinen Transformationsunterschieds zurück. Weitere Verfahren, die in obige Klassifikation nicht eingeordnet werden können, werden in Abschnitt 2.1.2.4 beschrieben.

2.1.2.1 Techniken auf Basis von Punkten

Dieser Abschnitt beschreibt Verfahren, die *nach* der Korrespondenzzuordnung ausschließlich die Mittel- bzw. Ankerpunkte der Regionen in die weitere Auswertung einbeziehen und nicht die Transformationsunterschiede der zugeordneten Regionen.

(OM02) nutzt MSER-Regionen (MCUP04) (siehe Abschnitt 3.2.4.1) zur robusten, affin- und beleuchtungsinvarianten Detektion von *lokalen affinen Frames* (LAF). Diese werden in eine kanonische Form, den normalisierten LAF's übergeführt und können dann mittels normalisierter Korrelation aller Farbkanäle miteinander verglichen werden. Die kanonische Form entspricht einer rektifizierten

Region mit dem mittelwertfreien Zentrum, $\mathbf{I}_{2\times 2}$ als Kovarianzmatrix sowie einer ausgerichteten Orientierung. Sie wird durch 21×21 Farb-Pixel über eine affine Transformation gesampelt (vgl. für die Formeln auch Abschnitt 3.3). Im Training werden aus einer Menge von Bildern die LAF's extrahiert, in ihre kanonische Form übergeführt und mit dem Bildindex gelabelt.

Die Zuordnung eines Testbildes zu den Trainingsbildern erfolgt dann über ein einfaches Abstimmungsschema ohne jegliche geometrische oder photometrische Validierung. Jedes extrahierte LAF wird rektifiziert, über Korrelation dem nächsten LAF der Modelldatenbank zugeordnet und gibt eine Stimme für das zugrundeliegende Modellbild ab. Das Bild mit der maximalen Stimmenanzahl wird dann mit einer eventuell hinterlegten Lage zurückgegeben. Experimente werden auf der COIL-100 (COI10) Bilddatenbank durchgeführt, die 100 Objekte auf schwarzem Hintergrund in 72 Lagen (alle 5° mit 1D-Orientierungsvariation) enthält. Die Erkennungsrate lag bei 100%, wenn alle 20° ein Bild zum Training verwendet wurde und bei 76% bei einem *einzelnen* Trainingsbild pro Objekt. Mit 50% Okklusion der Testbilder wurden immerhin noch 92.6% bzw. 63.3% erreicht. Eine Betrachtung der Lagegüte wurde nicht durchgeführt. Die Laufzeit pro Testbild ohne MSER-Berechnung lag bei 0.8 s auf einem 1.4 GHz PC mit 400 Bildern der Größe 128×128 in der Modelldatenbank.

Im Gegensatz zu einer Abbildung von lokalen Regionen auf eine kanonische Form werden alle möglichen Variationen der Regionen in (LPF04) eingelernt und das Zuordnungsproblem als Klassifikationsproblem aufgefasst. Sei $y(\mathbf{p}) \in [0,1,\ldots,n] = \mathcal{M}$ eine Funktion, die einer Region $\mathbf{p} \in \mathcal{P}$ die korrespondierende Region m aus einer n-elementigen Modelldatenbank oder dem Hintergrund 0 optimal zuordnet. Formal bedeutet das für den zu trainierenden Klassifikator

$$\hat{y} : \mathcal{P} \rightarrow \mathcal{M}$$

das jede Modellregion als eigene Klasse angesehen wird und die Wahrscheinlichkeit der Abweichung $p(y \neq \hat{y})$ minimal werden soll. Regionen werden als die 32×32 Umgebung von interessanten Punkten, hier o. B. d. A. den Harris-Ecken (HS88a) definiert. Um den Klassifikator zu trainieren werden für jede Klasse bzw. Region verschiedene Ansichten aller zu berücksichtigenden Transformationen benötigt. Um den Aufwand in der Trainingsphase zu reduzieren, werden diese Ansichten virtuell aus einer geringen Anzahl von realen Modellbildern erzeugt. Handelt es sich bei einer Bildregion lokal um eine Ebene, werden die geometrischen Transformationen (vgl. Abschnitt 3.1.2) durch eine affine Abbildung approximiert, ansonsten wird ein texturiertes 3D-Polygonmodell des einzulernenden Objekts bzw. dessen Ausschnitts genutzt um die Ansichten synthetisch zu generieren. Beleuchtungsvariationen werden nicht eingelernt, sondern alle Regionen bzgl. Kontrast und mittlerem Intensitätswert normiert. Optional können auch Ansichten der Hintergrundklasse erzeugt werden. Alle Ansichten werden mittels der Hauptkomponentenanalyse in den Eigenraum transformiert und auf 20 Dimensionen komprimiert. Mittels Clustering durch k-Means (KMN+02) werden nun für jede Klasse die 20 repräsentativsten Mittelwerte bzw. Ansichten bestimmt und in der Modelldatenbank zusammen mit dem Klassenlabel abgelegt. Klassen bzw. Regionen mit einer Missklassifikationsrate $p(y \neq \hat{y}|m) > 0.1$ über 10% werden entfernt. Die Klassifikation erfolgt dann über eine Nächste-Nachbar-Anfrage. Das Verfahren wird zur Objektlageschätzung herangezogen und mit dem Ansatz aus (Low99; Low01; Low04) verglichen (siehe Abschnitt 2.1.2.2). Bei 200 Regionen in der Datenbank benötigte das Verfahren ca. 0.2 s auf einem 2 GHz Rechen inkl. einer Homographieschätzung mit RANSAC (FB81) im Vergleich zu ca. 1 s bei dem System in (Low04). Eine quantitative Güte der Schätzung wurde nicht angegeben, allerdings verhält sich das System im Gegensatz zu (Low04) selbst bei großen Verkippungen über ca. 20° noch robust.

In (LF06) werden die Harris-Ecken durch eine schnelle Variante der robusteren Laplace-Maxima (LF04) ersetzt und mit den *Randomized Trees* (AG97) ein wesentlich effizienterer Klassifikator eingeführt. Dieser besteht aus einer Kollektion mehrerer binärer Entscheidungsbäume, die zwar an jedem

Knoten nur jeweils eine Dimension zerteilen, aber in ihrer Kombination eine feine Partition des Eigenraums erlauben. Daher kann die gesamte Verteilung der Ansichten berücksichtigt werden und das Clustering sowie die NN-Anfrage entfallen. Wiederum wurden keine quantitativen Ergebnisse der Lagegüte angegeben, die Framerate lag aber bei beachtlichen 25 Hz auf einem 2.8 GHz PC. Szenen mit mehreren (auch gleichen) Objekten wurden nicht behandelt.

2.1.2.2 Techniken auf Basis ähnlichkeits-kovarianter 2D-Regionen

Eines der ältesten und sicherlich das bekannteste Verfahren zu ansichtsbasierten Objekterkennung mit ähnlichkeits-kovarianten Regionen wird in (Low99; Low04) vorgestellt. Es nutzt sowohl den SIFT-Detektor (siehe Abschnitt 3.2.3.3) als auch -Deskriptor (siehe Abschnitt 3.3.1) zum Finden und Beschreiben von translations-, rotations- und skalierungsinvarianten lokaler Bildmerkmale. Die Zuordnung erfolgt über einen Vergleich der Deskriptoren, wobei zur schnellen Indexierung in der Modelldatenbank ein *kd-Tree* zusammen mit einem approximativen Nächster-Nachbar-Algorithmus, genannt *Best-Bin-First* (BBF) Technik (BL99) verwendet wird. Bei dieser werden anstatt des gesamten Baumes nur $n = 200$ Merkmale in den nächstgelegenen Zellen des Baumes untersucht. Im Training werden Bilder zusammen mit ihren SIFT-Merkmalen und dem kd-Tree in einer Modelldatenbank abgespeichert.

Für die Detektion werden die SIFT-Merkmale aus einem Testbild extrahiert und mittels BBF der nächste Nachbar aus der Modelldatenbank bestimmt. Für dieses Merkmalspaar werden nun die Unterschiede in der Translation, Rotation in der Bildebene und Skalierung bestimmt und damit für dieses Paar die Ähnlichkeitstransformation zwischen Modellbild bzw. des darin abgebildeten Objekts und des Testbildes. 3D-Verkippungen können damit allerdings nicht modelliert werden. Da die Wahrscheinlichkeit eines korrekten zugeordneten Paars laut (Low01) in komplexen Szenen bei ca. 1% liegt, werden über eine 4D-Hough-Transformation (Hou62) alle miteinander übereinstimmenden Merkmalspaare einer *einzelnen* Modellansicht gesucht. Der Hough-Raum wird in Zellen diskretisiert, die eine Größe von einem Viertel der Objektgröße in den beiden Translationsdimensionen entsprechen und 30° in der Rotationsdimension sowie 2 in der Skalierungdimension. Diese grobe Unterteilung ist nötig um tolerant gegenüber den nicht modellierten Verkippungen zu sein. Um Randeffekte zu vermeiden werden die Paare in die zwei benachbarten Zellen in jeder Dimension, d. h. in 16 Zellen insgesamt eingefügt. Befinden sich in einer Zelle mindestens 3 Paare, wird die affine Transformation der Mittelpunkte von der Modell- in die Testansicht bestimmt und als (affine) Lage des Objektes ausgegeben (Bsp. siehe Abb. 2.4). Dafür muss allerdings angenommen werden, dass die Modellpunkte coplanar sind, d. h. in der Realität in einer Ebene liegen. Es werden zwar keine quantitativen Ergebnisse präsentiert, das Verfahren soll aber bis ca. 20° Verkippung robust sein. Die Laufzeit lag bei 0.9 s für die Merkmalsberechnung und 0.6 s für die Zuordnung und affine Lagekonstruktion auf einen Sun Sparc 10.

(Low01) führt für obiges System ein verbessertes Training als auch eine probabilistische Lageverifikation ein. Weiterhin wird anstatt einer affinen Transformation nur eine Ähnlichkeitstransformation als Lage geschätzt, da diese robuster bei nicht komplanaren Objekten geschätzt werden kann. Seien k zugeordnete Bildpunktpaare $(\mathbf{p},\mathbf{q})_i$ zwischen Modell- und Testbild gefunden, so soll die Transformation $\mathbf{p} = s\mathbf{R}[\varphi]\mathbf{q} + \mathbf{t}$, $s > 0$ gefunden werden. Sei $l = s\cos\varphi$ und $n = s\sin\varphi$ dann folgt daraus das lineare Gleichungssystem

$$\mathbf{A}\mathbf{x} = \begin{pmatrix} u_\mathbf{p} & -v_\mathbf{p} & 1 & 0 \\ v_\mathbf{p} & u_\mathbf{p} & 0 & 1 \\ \vdots & & & \vdots \end{pmatrix} \begin{pmatrix} l \\ n \\ u_\mathbf{t} \\ v_\mathbf{t} \end{pmatrix} = \begin{pmatrix} u_\mathbf{q} \\ v_\mathbf{q} \\ \vdots \end{pmatrix} = \mathbf{b}$$

welches mittels der Methode der kleinsten Fehlerquadrate gelöst werden kann. Der durchschnittliche

Abbildung 2.4: Objektlokalisation mittels SIFT-Merkmalen und dem Verfahren aus (Low99). Links sind die zwei Trainingsbilder dargestellt, rechts die gefundenen Instanzen in einer komplexen Szene. Die großen Vierecke zeigen die affin transformierten Ansichtsgrenzen, die kleinen Rechtecke die verwendeten SIFT-Merkmale. Aus (Low04) entnommen.

Fehler e kann über

$$e = \sqrt{\frac{||\mathbf{b} - \mathbf{Ax}||}{k-2}}$$

angegeben werden. Er hat in etwa die Größe $\tau_e = 0.05l$ bei der Objektgröße l und einer (nicht modellierten) Verkippung um 20°.

In der Trainingsphase werden dem System nun sukzessive unregistrierte Bilder von Objekten (idealerweise auf schwarzem Hintergrund) präsentiert und mittels dem Verfahren aus (Low99) gegen die schon bestehende Modelldatenbank getestet. Sollte es keine Häufungen der Merkmalspaare in den Houghzellen geben, muss es sich offensichtlich um ein neues Objekt handeln und es wird in die Datenbank aufgenommen. Treten Häufungen auf und der Fehler der geschätzten Ähnlichkeitstransformation ist $e > \tau_e$, so handelt es sich um eine neue Ansicht eines vorhandenen Objekts und wird ebenfalls in die Modelldatenbank aufgenommen. Bei dem Fall $e < \tau_e$ ist die Ansicht schon vorhanden und nur die neuen, unähnlichen SIFT-Merkmale werden der Modellansicht zugeordnet. Da die Hough-Auswertung für jedes Modellbild separat erfolgt, werden ähnlichen Merkmale in verschiedenen benachbarten Ansichten während des Trainings miteinander verlinkt und es wird pro zugeordnetes Testmerkmal für alle verlinkten Modellmerkmale ein Hough-Eintrag erzeugt.

Die Objektdetektion erfolgt auf dieselbe Weise, wobei gefundene Lagehypothesen nicht nur über e sondern auch eine probabilistische Modellierung verifiziert werden. Sei eine Menge $\mathcal{F}$ von k zugeordneten Merkmalen gegeben und $p(m|\mathcal{F})$ die Wahrscheinlichkeit, dass das Modell an der ermittelten Lage ist. Sei weiterhin $q = dlrs$ die Wahrscheinlichkeit, dass ein zugeordnetes Merkmal fälschlicherweise in einer Hough-Zelle landet. d ist dabei die Wahrscheinlichkeit einer falschen Zuordnung, d. h. das Verhältnis der Anzahl Merkmale in der Modellansicht zu der Anzahl aller Modellmerkmale, $l = 0.2^2 = 0.04$ ist die räumliche Einschränkung, $r = 30^\circ/360^\circ = 0.085$ ist die rotative Einschränkung und $s = 0.5$ die Einschränkung in der Skalierung. Die Wahrscheinlichkeit $p(\mathcal{F}|\neg m)$, dass die Merkmalsmenge fälschlicherweise erzeugt wurde, beträgt dann

$$p(\mathcal{F}|\neg m) = \sum_{k}^{n} \binom{n}{j} q^j (1-q)^{n-j}$$

wobei n die Anzahl potentieller Falschkandidaten ist und sich über die Anzahl der Merkmale in der Testansicht, die innerhalb der projizierten Objektgrenzen liegen, definiert. Über die Formel von Bayes

und der Annahme $p(\neg m) \approx p(\mathcal{F}|m) \approx 1$ ergibt sich

$$p(m|\mathcal{F}) = \frac{p(m)}{p(m) + p(\mathcal{F}|\neg m)}$$

wobei $p(m) = 0.01$ als die Wahrscheinlichkeit einer korrekten Zuordnung eines einzelnen Merkmals angenommen wird. Eine Hypothese wird akzeptiert falls $p(m|\mathcal{F}) > 0.95$ ist, d. h. falls die Wahrscheinlichkeit, dass die Merkmale fälschlicherweise zugeordnet werden wesentlich geringer ist als die a priori Wahrscheinlichkeit, dass das Objekt an dieser Stelle ist. Es werden ebenfalls nur visuelle Ergebnisse präsentiert, die Laufzeit lag bei 0.8 s des Gesamtsystems auf einen 600 MHz Pentium 3.

Während in (Low99; Low01) nur affine bzw. ähnliche Projektionen der Lage eines Objektes in den Testbildern geschätzt wurden, erweitert (GL06) das System auf eine vollständige 3D-Modellierung des Modells und 6D-Lageschätzung im euklidischen Raum für eine Augmented Reality Anwendung, bei der in Video-Sequenzen virtuelle Gegenstände korrekt realen Objekten überlagert werden sollen. Dafür ist eine sehr exakte Schätzung der 6D-Lage der realen Objekte im Kamera-Koordinatensystem notwendig, deren Ideen hier beschrieben werden sollen.

Das Training erfolgt wie in (Low01), allerdings werden zwei benachbarte registrierte Bilder über die Schätzung der Epipolargeometrie (HZ04) im euklidischen Raum in Beziehung gesetzt und über deren Constraint die besten SIFT-Korrespondenzen zwischen den Bildern ermittelt. Sind alle Bilder eines Modells verarbeitet, werden diese Korrespondenzen herangezogen um mittels eines *Structure-From-Motion*-Ansatzes (SK94) ein vollständiges Modell aller Ansichten und der Zentren aller SIFT-Merkmale im 3D-Raum zu erhalten. Die Zuordnung eines Testbildes erfolgt dann wie in (Low99), allerdings erhält man nun n 2D-3D-Korrespondenzen, die mittels eines *PnP*-Verfahrens (siehe Abschnitt 2.2.2.2) gelöst werden können. Hier wird ein nichtlineares Gleichungssystem aufgestellt und mittels RANSAC (FB81) und einer iterativen Levenberg-Marquardt (Mar63) Minimierung gelöst. Quantitative Ergebnisse wurden nicht präsentiert, die Laufzeit lag aber bei 250 ms auf einem 1.8 GHz Pentium 4 unter Ausnutzung von OpenGL.

(CBSF09) erweitert obiges System insbesondere auf den Umgang mit mehreren gleichen Objekten während der Detektion und setzt es in einer Humanoiden-Robotik-Applikation zum Greifen realer Gegenstände ein. Anstatt einer Merkmalsgruppierung mit Hough wird hier der *MeanShift*-Algorithmus (Che95) zum Clustern der Merkmalszentren im Testbild verwendet und für jede gefundene Häufung die Objektdetektion aus (GL06) durchgeführt. Gleiche Objektlagehypothesen fälschlicherweise getrennter Cluster werden abschließend wieder zusammengeführt. Es wurden sowohl Genauigkeitsuntersuchungen als auch Greifversuche mit einer realen Coladose, Saftflasche, Reisschachtel und einem Notizblock durchgeführt. Zur Auswertung der Translationsgenauigkeit wurden die Objekte im Abstand von 30-90 cm entlang der Z-Achse der Kamera variiert und um bis zu 20 cm in X und Y verschoben. Für die Verkippungsgenauigkeit wurden das Objekt in 50 cm Abstand positioniert und in 15°-Schritten im Bereich $[-45°, 45°]$ aus der Kameraebene gedreht. Im selben Bereich wird auch die Rotation in der Kameraebene getestet. Der durchschnittliche Fehler in der Translation lag bei 0.67 cm, in der Verkippung bei 3.81° und in der Rotation in der Ebene bei 1.23° bei ausschließlicher Verwendung einer Monokamera. 91% aller Greifexperimente waren erfolgreich. Die Framerate des Systems lag bei 3-6 Hz unter Ausnutzung eines PCs für die Merkmalsberechnung und eines weiteren für die Lageschätzung.

Eine andere Variante von (Low99) mit speziellen Augenmerk auf Geschwindigkeit wird in (AAD07) für eine Humanoide-Robotik-Applikation vorgestellt - allerdings nur für Objekte mit einer dominanten ebenen Fläche. Anstatt den SIFT-Merkmalen werden nur translations- und rotationsinvariante *Harris-SIFT*-Merkmale verwendet, die aus einer Kombination von Harris-Ecken (HS88a) mit dem

SIFT-Deskriptor entstehen. Für die skaleninvariante Zuordnung wird der Deskriptor auf 3 Skalen im Abstand 0.75 voneinander berechnet, ansonsten wird das Verfahren aus (Low99) verwendet. Anstatt einer 4D-Hough-Transformation wird nur eine 3D-Hough-Transformation in der Bildebene durchgeführt, wobei durch die fehlende Skaleninformation mehrere Einträge auf verschiedenen Skalen gemacht werden. Danach wird ebenfalls mit RANSAC eine affine Tranformation oder Homographie geschätzt um weitere Ausreißer zu eliminieren und dann ein PnP-Algorithmus (LHM00) (siehe Abschnitt 2.2.2.2) für eine 6 DoF Lageschätzung verwendet. Diese ist allerdings nicht robust (Aza08) falls die Güte der Merkmalszentren nicht sehr exakt ist. Daher wird mittels Stereokamera für alle verwendeten Merkmalszentren deren 3D-Punkt bestimmt und in die Punktwolke eine Ebene hineingefittet. Auf diese werden dann die Eckpunkte des Objekts projiziert und die 6 DoF Lage ermittelt. Das System wird durch Simulation genau evaluiert, für Details sei auf (Aza08) verwiesen. Es ist sehr schnell und kann mit 23 Hz (ohne Bildaufnahme) auf einem 3 GHz Single-Core Rechner betrieben werden.

2.1.2.3 Techniken auf Basis affin-kovarianter 2D-Regionen

(PLRS04) nutzt sowohl für die Modellgenerierung in der Trainingsphase als auch die Objektdetektion rektifizierte affine Patches des affinen Harris-Detektors (HaA) (MS04) (siehe Abschnitt 3.2.2.2) unter Verwendung von geometrischen Zwangsbedingungen bzgl. mehrerer Ansichten. Sei $\check{\mathbf{R}}_{3\times3}$ eine affine Projektionsmatrix, die einen Patch $\mathcal{P}$ in seine kanonische Form $\check{\mathcal{P}}_r = \check{\mathbf{R}}\check{\mathcal{P}}$ transformiert und $\check{\mathbf{S}}_{3\times3} = \check{\mathbf{R}}^{-1} = [(\mathbf{u},0)^{\mathrm{T}},(\mathbf{v},0)^{\mathrm{T}},\check{\mathbf{c}}]$ die entsprechende inverse affine Projektionsmatrix, die einen rektifizierten Patch wieder zurück in die Bildebene transformiert. $\check{\mathbf{S}}$ enthält das Patchzentrum $\mathbf{c}$ als auch die affin verzerrten Einheitsvektoren $\mathbf{u}$ und $\mathbf{v}$ des lokalen Patch-Koordinatensystem im Bild in homogener Darstellung. Ebenfalls kann man $\check{\mathbf{S}} = \check{\mathbf{M}}\check{\mathbf{N}}$ als die Komposition einer inversen affinen Projektionsmatrix $\check{\mathbf{N}}_{4\times3} = [\mathbf{B}_{3\times3}^{\mathrm{T}},(0,0,1)^{\mathrm{T}}]^{\mathrm{T}}$ und einer affinen Projektionsmatrix $\check{\mathbf{M}}_{3\times4} = [(\mathbf{A}_{2\times3},\mathbf{b})^{\mathrm{T}},(0,0,0,1)^{\mathrm{T}}]^{\mathrm{T}}$ modellieren. $\check{\mathbf{N}}$ bildet dabei den rektifizierten virtuellen Patch auf den realen, als Ebene angenommenen physikalischen Oberflächenpatch der Szene ab und $\check{\mathbf{M}}$ projiziert den Oberflächenpatch in ein Bild.

Eine Zwangsbedingung lässt sich nun für m Bilder und n davon extrahierte korrespondierende Patches angegeben. Seien $\mathbf{R}_{ij}$ bzw. $\mathbf{S}_{ij}$ die zugehörigen Transformationsmatrizen von Bild $i \in [1,\ldots,m]$ und Patch $j \in [1,\ldots,n]$, das (gemeinsame) physikalische Koordinatensystem der Oberflächenpatches der Schwerpunkt deren Zentren und die Projektion dieses Punktes in die jeweiligen Bilder deren Koordinatenursprung. Dann folgt daraus $\mathbf{b} = \mathbf{0}$ und

$$\overline{\mathbf{S}} = \begin{pmatrix} \mathbf{S}_{11} & \ldots & \mathbf{S}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{m1} & \ldots & \mathbf{S}_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_m \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \ldots & \mathbf{B}_n \end{pmatrix} = \overline{\mathbf{A}}\,\overline{\mathbf{B}}$$

Ist $n > 2$ oder $m > 2$, ist das System überbestimmt und $\overline{\mathbf{A}}$ bzw. $\overline{\mathbf{B}}$ müssen im Sinne der kleinsten Fehlerquadrate über die SVD (Bro99) bestimmt werden.

Im Training werden nun paarweise Bilder registriert, indem zuerst korrespondierende n Patches über Korrelation im rektifizierten System ermittelt werden und dann die Matrizen $\overline{\mathbf{A}}$ bzw. $\overline{\mathbf{B}}$ bestimmt werden. Über die normierte Frobeniusnorm $||\overline{\mathbf{S}} - \overline{\mathbf{A}}\,\overline{\mathbf{B}}||_F/(\sqrt{6n})$ kann ein geometrisches Maß für die Registrierungsgüte angegeben werden. So werden sukzessive Bilder registriert, bis ein Objekt vollständig von allen Seiten erfasst ist. Danach werden alle Registrierungen in einem iterativen Minimierungsverfahren nochmals verfeinert und man erhält ein bis auf eine Skalierung definiertes 3D-Modell des Objektes, das alle Oberflächenpatches mit ihren Texturinformationen in rektifizierter Form geometrisch miteinander in Beziehung setzt. Während der Detektion werden aus einem Testbild wiederum affine Patches mit dem HaA extrahiert und über Korrelation n potentielle Zuordnungen gefunden. Aus dem Testbild sind die inversen Projektionsmatrizen $\mathbf{S}_i = \mathbf{A}\mathbf{B}_i$ bekannt und

Abbildung 2.5: Objektlokalisation mittels rektifizierter lokaler Bildpatches aus (RLSP06).

aus dem Modelltraining die Matrizen $\mathbf{N}_i$ bzw. $\mathbf{B}_i$, die die Beziehung der rektifizierten Patches zu den realen Oberflächenpatches repräsentiert. Die unbekannte Matrix $\mathbf{A}$ projiziert das Modell bzw. die verwendeten Modellpatches zurück in das Testbild und kann daher zur Lagebestimmung herangezogen werden. Sie wird mittels SVD robust durch Minimierung der kleinsten Fehlerquadrate aus $[\mathbf{B}_1,\ldots,\mathbf{B}_n]^{\mathrm{T}}\mathbf{A} = [\mathbf{S}_1,\ldots,\mathbf{S}_n]^{\mathrm{T}}$ bestimmt. Ein Gütemaß kann wiederum über die normierte Frobeniusnorm angegeben werden.

(RLSP06) verwendet anstatt der Korrelation verschiedene Beschreibungen für die rektifizierten Patches, wobei eine Kombination aus dem SIFT-Deskriptor (Low04) und 10×10 Farbhistogrammen aus dem UV-Bereich des YUV-Farbraums die besten Ergebnisse zeigten. Ebenfalls wurde neben der rein photometrischen Patchzuordnung noch eine mit RANSAC (FB81) vergleichbare geometrische Überprüfung der Zuordnungen durchgeführt, was die Robustheit des Systems weiter verbessert hat. Leider werden keine quantitativen Ergebnisse präsentiert, ein Vergleich mit dem Verfahren von (Low04) zeigt ähnliche Ergebnisse, allerdings mit einer deutlich längeren Verarbeitungszeit im Minutenbereich auf einem 3 GHz Rechner. Ein Beispiel der Objektlokalisation ist in Abb. 2.5 dargestellt.

2.1.2.4 Andere Techniken

(HCK97) stellt ein System vor, bei dem 3D-Objekte bzw. dessen Ansichten durch eine Menge von (Bild-) *Teilen* dargestellt werden. Teile sind definiert als polynomiale Oberflächen, die Approximationen von geschlossenen, nicht überlappenden Bildregionen sind, welche ein Bild auf optimale Weise im Sinne einer *Minimum Description Length*, kurz MDL (KDNS94) partitionieren. Die MDL-Minimierungsfunktion codiert dabei die Grenzen der Regionen und die statistische Beschreibung der Regioneninhalte. Teile 1 und 2 haben dieselbe *Erscheinung*, wenn der Fehler ε_{12} ihrer statistischen Beschreibungen, repräsentiert durch die Parametervektoren $\mathbf{p}_{1,2}$ bezogen auf den jeweiligen anderen Regioneninhalt $\mathbf{i}_{1,2}$ unter einem Schwellwert τ liegt:

$$\varepsilon_{12} = n_1^{-1}||\mathbf{i}_1 - \mathbf{A}_1\mathbf{p}_2|| + n_2^{-1}||\mathbf{i}_2 - \mathbf{A}_2\mathbf{p}_1|| < \tau$$

$\mathbf{i}$ ist dabei ein Vektor von Bildintensitäten der Länge n, $\mathbf{A}_{n\times m}$ eine Matrix von Basisfunktionen (z. B. Produkte von Bildkoordinatenpotenzen) und $\mathbf{p}$ ein optimaler Vektor von Regressionsparametern der Länge m, so dass $\varepsilon = ||\mathbf{i} - \mathbf{A}\mathbf{p}||$ minimal ist.

Mittels eines Drehtellers werden in einem Trainingsschritt Ansichten mit verschiedener 1D-Orientierung eines 3D-Objekts aufgenommen und eine Sammlung aller Teile derselben Erscheinung

aufgebaut. Von jeder Teilesammlung werden nun die Bildregionen bzgl. Größe und Intensität normiert, in den Eigenraum transferiert und mittels des Verfahrens (MN95) aus Abschnitt 2.1.1.1 eine über die Lage parametrisierte 1D-Mannigfaltigkeit, genannt erscheinungsbasiertes Teil (ABP[4]) aufgebaut. Dasselbe geschieht mit einer Vereinigung von Teilen bzw. Bildregionen, die über eine gemeinsame Grenze in räumlicher Beziehung stehen und als erscheinungsbasierte Beziehungen (ABR[5]) bezeichnet werden.

Für die Detektion werden die Eingabebilder über die MDL-Segmentierung in Teile partitioniert und deren Bildregionen einem der ABP's zugeordnet oder zurückgewiesen, falls der Abstand zur nächstgelegenen Mannigfaltigkeit zu groß ist. Über die Projektion auf die Mannigfaltigkeit erhält man dann für jedes Teil ebenfalls eine Objektlage. Mittels mehrerer Teile derselben Lage und desselben Objekts lässt sich über die Distanz zu den ABR's ein Gütemaß angeben. Experimente wurden nur mit matten 3D-Objekten in zwei verschiedenen Szenen mit unstrukturiertem Hintergrund und mehreren unterschiedlichen Objekten durchgeführt. Die Falschalarmrate der Objektdetektion lag bei 16% bzw. 14% und die Missklassifikationsrate bei 16% bzw. 4%. Die Güte der Lage und die Laufzeiteigenschaften wurden nicht beschrieben.

Ein System zur Erkennung und 1D-Lokalisation von 3D-Objekten mittels aktiver Stereokamera auf Basis von Konturen wird in (Kef98) vorgestellt. Als Vorverarbeitung werden durch einen *Mallatfilter* (MZ92) aus den Farbbildern auf fünf verschiedenen Skalen die Kantenstärke und -orientierung extrahiert. In einem iterativen Verfahren, das Konturkanten von anderen Kanten separiert (RPM96), wird danach jedem lokalen Kantenelement ein Konfidenzwert zugewiesen. Es bevorzugt Kanten, die Teil einer kontinuierlichen Kurve über viele Skalen sind und unterdrückt Kanten einer einzigen Skale, die vorwiegend durch Rauschen, Schattenwurf oder Textur entstehen.

In einer Trainingsphase werden von einem Objekt mittels eines Drehtellers Ansichten mit verschiedener 1D-Lage aufgenommen. Diese werden dann jeweils mit einem *gelabelten Modellgraph* beschrieben, dessen Knoten aus Kantenelementen auf der Kontur bestehen und die Kanten des Graphs aus Pixelabständen entlang der Kontur gebildet werden. Es werden nur die 25-45 Knoten mit den meisten Nachbarn verwendet um den Hintergrund auszublenden. Ebenfalls werden alle Knoten mit mehr als 6 Nachbarn eliminiert, da diese sehr sicher innerhalb des Objektes und nicht auf der Kontur liegen (KRM97). Jeder Knoten enthält weiterhin eine Beschreibung seiner Umgebung in Form von sogenannten *Jets*, die aus Kantenbeträgen und -orientierungen auf allen fünf Skalen bestehen. Die Menge aller Modellgraphen eines Objektes wird als *Multigraph* bezeichnet und dient als Objektmodell.

Eine Groblokalisation in der Detektionsphase erhält man über Blobmaxima des Konturkantenbildes gefaltet mit einem Gaußkernel. Die größten Maxima werden in beiden Stereobildern bestimmt, über eine Farbsegmentierung verifiziert und mittels Stereotriangulation deren Kamera-Koordinaten bestimmt. Der 3D-Punkt mit dem stärksten Maximum wird dann als grobe Objekthypothese angesehen und in der Stereokamera zentriert. Daraufhin wird jeder Modellgraph in der Objektdatenbank in das zentrierte Bild mittels des *Elastic Graph Matchings* (KRM97) gefittet und die beste Zuordnung zurückgegeben. Um dies zu beschleunigen werden die Graphen erst starr innerhalb eines gleitenden Fensters über das Bild gelegt und an jedem Knotenpunkt durch das normalisierte Skalarprodukt der Fehler der Jets bestimmt. An der Position mit dem kleinsten akkumulierten Fehler wird nun eine Verformung jedes Knoten innerhalb eines 9×9 großen Pixelbereichs sowie eine Skalierung des gesamten Graphes in $u-$ und $v-$Richtung innerhalb des Intervalls $[0.8, 2.0]$ erlaubt. Über die während des Trainings hinterlegte Lage der Ansicht des Modellgraphen wird dann die 1D-Orientierung zurückgegeben. Eine Interpolation zwischen den Ansichten sowie eine Behandlung der Rotation in der

[4] engl.: **A**ppearance **B**ased **P**art
[5] engl.: **A**ppearance **B**ased **R**elationship

Kameraebene erfolgt nicht.

In (Kef01) wird das *Elastic Graph Matching* auf Stereo erweitert und berücksichtigt in einem integrierten Prozess ebenfalls die Epipolareinschränkungen anstatt im linken und rechten Bild separat einen Modellgraphen einzufitten. Die Experimente wurden mit 10 eingelernten Objekten in der Modelldatenbank durchgeführt, wobei alle 1.8° eine Ansicht aufgenommen wurde. In 95.8% der Tests wurde das Objekt mit einer Lagegenauigkeit zwischen $2-3^\circ$ für Szenen mit einzelnen Objekten und homogenem Hintergrund erkannt, wobei dieser Wert auf 82.1% bei Szenen mit mehreren Objekten und strukturiertem Hintergrund fällt. Die Laufzeit lag *pro* Modellgraph in der Datenbank bei $0.04-1.2\,\mathrm{s}$ auf einem Pentium 233 MHz und einer Bildgröße von 256×256 Pixeln.

2.2 Geometriebasierte Techniken

Während ansichtsbasierte Methoden Objekte mittels visueller Ähnlichkeit erkennen, versuchen geometriebasierte Techniken den Zusammenhang zwischen Objekten und ihren Projektionen in Bildern analytisch zu modellieren. Dafür ist ein 3D-Modell der Objekte notwendig, welches bei den meisten Verfahren a priori als angepasstes CAD-Modell gegeben ist. Die meisten lokalen Verfahren verwenden Kanten oder Ecken dieser Modelle und versuchen diese in einem Registrierungsprozess mit den beobachteten Projektionen im Bild in Einklang zu bringen.

Ein grundsätzlich anderer Ansatz zur Objekterkennung verwendet anstatt Intensitätsbilder Abstandsbilder bzw. Punktwolken. Diese 3D-Daten können von Stereo- oder Laufzeit-Kameras und Laserscannern kommen. Allerdings benötigen die meisten dieser Verfahren relativ genaue bzw. dicht abgetastete Tiefeninformationen, was mit dem in dieser Arbeit verwendeten Sensorikaufbau nur schwer realisierbar ist. Dennoch werden die Ansätze hier der Vollständigkeit halber ebenfalls vorgestellt.

Eine große Herausforderung der geometriebasierten Techniken ist die automatische Modellgenerierung, da im Gegensatz zu den ansichtsbasierten Verfahren nicht einzelne Bilder bzw. Bildausschnitte verwendet werden, sondern ein komplettes 3D-Modell aufgebaut werden muss. Aus diesem Modell müssen dann je nach Verfahren weitere Merkmale abgeleitet werden, z. B. sichtbare Ecken oder Kanten.

Wie schon bei den ansichtsbasierten Verfahren lassen sich die geometriebasierten Techniken in globale und lokale Ansätze unterschieden. Einen guten Überblick bieten insbesondere für die Verfahren mit Abstandsdaten die Artikel (CF01; SMFF07).

2.2.1 Globale Ansätze

Eine der bekanntesten globalen Ansätze ist der *Iterative Closest Point Algorithmus*, welcher im folgenden vorgestellt wird. Weitere Möglichkeiten finden sich im darauf folgenden Abschnitt mit den *Extended Gaussian Images* und den *Histogrammen auf Tiefenbildern*. Der große Nachteil der globalen, geometriebasierten Ansätze ist wie bei den globalen ansichtsbasierten Ansätzen, dass sie einen Segmentierungsschritt im Vorfeld benötigen und meist Probleme mit Verdeckungen bzw. lokalen Störungen haben.

2.2.1.1 Iterative Closest Point (ICP)

Der *Iterative Closest Point* (ICP) Algorithmus (CM92; BM92) ist ein Verfahren, um eine beliebig geformte Punktmenge $\mathcal{P}$ in eine Modellform $\mathcal{X}$ zu fitten. Das Modell $\mathcal{X}$ kann in vielen verschiedenen Repräsentationen, wie eine Menge von Punkten, Linien, Dreiecken, parametrisierbare oder implizite Kurven bzw. Oberflächen vorliegen (BM92). Üblicherweise werden Punktwolken oder Dreiecksnetze und optional die Normalen der Punkte bzw. Dreiecke verwendet. Der Algorithmus liefert die 6D-Transformation, um $\mathcal{P}$ und $\mathcal{X}$ zu registrieren, ohne dass explizite Korrespondenzen gegeben sein

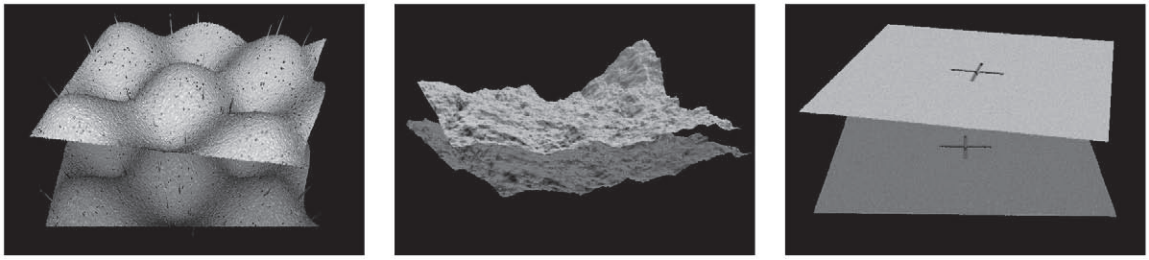

Abbildung 2.6: Registrierung von 3D-Oberflächen mit unterschiedlichen Herausforderungen für den ICP. Von links nach rechts, Wellenoberläche mit grober Struktur, gaußschem Rauschen und Ausreißern, fraktale Oberfläche mit Strukturen in jeder Auflösungsstufe und Ebene mit wenigen, aber entscheidenden Details für eine exakte Registrierung. Aus (RL01) entnommen.

müssen. Das Verfahren ist iterativ und konvergiert gegen das nächste *lokale* Minimum. Der ICP-Algorithmus eignet sich daher nur als Feinlageschätzung und ist auf eine Segmentierung als auch eine Groblageschätzung angewiesen. Im Folgenden wird erst der Grundalgorithmus (BM92) vorgestellt und danach in Anlehnung an (RL01) die wichtigsten von unzähligen Erweiterungen skizziert. Abb. 2.6 zeigt einige Registrierungsbeispiele mit unterschiedlichen Herausforderungen.

Der ICP-Algorithmus ermittelt iterativ eine Starrkörpertransformation $\mathcal{P}' = \mathbf{R}\mathcal{P} + \mathbf{t}$, um alle Punkte $\mathbf{p}' \in \mathcal{P}'$ möglichst optimal mit dem Modell $\mathcal{X}$ in Deckung zu bringen:

$$(\mathbf{R}_{\text{opt}}, \mathbf{t}_{\text{opt}}) = \underset{(\mathbf{R},\mathbf{t})}{\operatorname{argmin}} \sum_{\mathbf{p}\in\mathcal{P}} \mathrm{d}(\mathbf{R}\mathbf{p} + \mathbf{t}, \mathcal{X})^2$$

Die Metrik $\mathrm{d}(\cdot)$ gibt dabei die Distanz zwischen einem Punkt und dem Modell an und ist abhängig von der Repräsentation von $\mathcal{X}$. Die Grundform des Algorithmus kann in 7 Schritten zerlegt werden und wird beispielhaft anhand eines Punktwolkenmodells mit $\mathbf{x} \in \mathcal{X}$ beschrieben:

Schritt 0: Groblageschätzung Initialisiere die Iterationsvariable $k = 0$ und mittels eines Verfahrens zur Groblageschätzung die Parameter $\mathbf{R}^{[0]}$ und $\mathbf{t}^{[0]}$ der Starrkörpertransformation.

Schritt 1: Selektion von Punkten Ist die Anzahl $|\mathcal{P}|$ groß, ist es aus Effizienzgründen meist sinnvoll die Punktmenge $\mathcal{P}$ auszudünnen und nur eine Teilmenge $\mathcal{P}^{[k]} \subseteq \mathcal{P}$ zu betrachten. Im einfachsten Fall genügt jedoch $\mathcal{P}^{[k]} = \mathcal{P}$.

Schritt 2: Korrespondenzzuweisung Zur Berechnung der Metrik $\mathrm{d}(\mathbf{p}', \mathcal{X})$ mit $\mathbf{p}' = \mathbf{R}^{[k]}\mathbf{p} + \mathbf{t}^{[k]}$ und $\mathbf{p} \in \mathcal{P}^{[k]}$ ist es nötig, dem Punkt $\mathbf{p}'$ ein geeignetes Element $\mathbf{x} \in \mathcal{X}$ aus dem Modell zuzuweisen. Im Falle einer Punktewolkenmodells kann dies über $\mathrm{d}(\mathbf{p}', \mathcal{X}) = \min_{\mathbf{x}\in\mathcal{X}} |\mathbf{p}' - \mathbf{x}|$ realisiert werden. Es wird dann jeweils der nächste Punkt des Modells einem Punkt der transformierten Punktmenge $\mathcal{P}'$ zugewiesen. Dieser Schritt ist das Herzstück des Algorithmus, hat den größten Einfluss auf die Güte und Performance und gibt dem Verfahren seinen Namen.

Schritt 3: Gewichtung der Korrespondenzen Existieren Gütekriterien, um die gefundenen Korrespondenzen zu bewerten, kann man diese für die nachfolgende Minimierung gewichten. Im einfachsten Fall werden alle Korrespondenzen gleich gewichtet.

Schritt 4: Ausreißereliminierung Optional kann die Menge der Korrespondenzen auf Ausreißer überprüft werden und diese gegebenenfalls verworfen werden.

Schritt 5: Berechnung der Transformationsparameter Mittels der Menge an Korrespondenzen können nun die neuen Transformationsparameter $\mathbf{R}^{[k+1]}$ und $\mathbf{t}^{[k+1]}$ so berechnet werden, dass die Distanzen der Korrespondenzen minimiert werden. Im einfachsten Falle von 3D-3D-Korrespondenzen existieren mehrere analytisch geschlossene Lösungen (Hor87; AHB87; HHN88), die in Abschnitt 2.2.2.1 behandelt werden.

Schritt 6: Überprüfung des Abbruchkriteriums Es kann entweder eine festen Anzahl an Iterationsschritten $k_{\max}$ verwendet werden oder überprüft werden, ob die Summe der Fehlerresiduen $e^{[k+1]}$ aus der Distanzminimierung in Schritt 5 bzgl. des vorangegangen Iterationsschritts kleiner als ein Schwellwert $\tau > e^{[k]} - e^{[k+1]}$ ist. Nach (BM92) konvergiert der Algorithmus meist in unter $k_{\max} = 20$ Iterationsschritten. Ist das Abbruchkriterium nicht erfüllt, wird k erhöht und zu Schritt 1 gesprungen. Ansonsten bricht der Algorithmus mit der aktuellen Lösung ab.

Der ICP-Algorithmus ist bei entsprechender Grobregistrierung robust gegenüber Punktmengen $\mathcal{P}$, die nur Teilmengen des Modells $\mathcal{X}$ beschreiben. Er generalisiert weiterhin zu Dimensionen größer 3 und kann auch zur Registrierung von 2D Kurven herangezogen werden. Der Algorithmus ist zwar in seiner Grundform nicht sehr performant, im Folgenden werden allerdings für alle obigen Ausführungsschritte Erweiterungen vorgestellt, die sowohl die Konvergenzgeschwindigkeit und Laufzeit als auch die Güte des Ergebnisses deutlich verbessern. Ebenfalls können viele Schritte des ICP auf einfache Weise parallel implementiert werden.

Schritt 0: Varianten der Groblageschätzung Eine einfache Form der Groblageschätzung ergibt sich über die ersten zwei zentralen Momente von $\mathcal{X}$ für *segmentierte* Punktmengen, falls $\mathcal{P}$ annähernd denselben Bereich wie das Modell beschreibt. Die Groblage wird so ermittelt, dass die Schwerpunkte und Hauptachsen der beiden Mengen übereinander liegen. Die Hauptachsen definieren sich aus den drei Eigenvektoren der Streu- bzw. Kovarianzmatrizen und werden über die Größe ihrer zugehörigen Eigenvektoren $\lambda_1 \geq \lambda_2 \geq \lambda_3$ identifiziert. Dies ist daher nur für Modelle eindeutig möglich, in denen die Verhältnisse $r_1 = \lambda_1/\lambda_2$ bzw. $r_2 = \lambda_2/\lambda_3$ der Eigenvektoren deutlich größer als eins sind. Da die Hauptachsen allerdings keine Richtungsinformation aufweisen, ergeben sich immer noch vier mögliche Rotationen, die mittels des ICP überprüft werden müssen um eine globale Registrierung zu erreichen. Der Hauptachsenansatz wird z. B. in (BM92; DWJ97) verwendet.

Eine andere Möglichkeit der Groblageschätzung ergibt sich über Punktkorrespondenzen von $\mathcal{P}$ und $\mathcal{X}$, die über eine Ähnlichkeitsbeschreibung gefunden werden (JH97). Diese können herangezogen werden, um eine Registrierungstransformation beider Mengen zu finden. Sie sind allerdings aufgrund der Lokalität der Informationen selten so genau, dass sie ohne ICP Verfeinerung der Transformation auskommen. Ihr großer Vorteil ist allerdings, dass sie meist ebenfalls das Segmentierungsproblem lösen und mit Verdeckungen bzw. lokalen Störungen umgehen können. Diese Verfahren werden im Detail in Abschnitt 2.2.2.3 behandelt.

Wie in (BM92) dargelegt wird, hat die Güte der initialen Transformation einen entscheidenden Einfluss auf die Konvergenzgeschwindigkeit.

Schritt 1: Varianten der Punktselektion (BM92) nimmt immer alle Punkte aus $\mathcal{P}$. Diese Methode wird allerdings bei großen Punktmengen sehr ineffizient, daher verwendet (Pul99) mit dem *Random Sampling* eine einfache aber performante Möglichkeit der Ausdünnung. $\mathcal{P}^{[k]}$ ergibt sich aus zufälligen Ziehen einer festen Anzahl an Punkten aus $\mathcal{P}$. Für schwierige Punktmengen, bei denen die korrekte Registrierung von wenigen, kleinen Strukturen abhängt (vgl. Abb. 2.6 rechts), besteht dabei allerdings die Gefahr, die entscheidenden Details zu übersehen. (RL01) schlägt daher vor, den zufälligen Selektionsprozess so zu steuern, dass die Normalenrichtungen aller gezogenen Punkte möglichst gleichverteilt ist. Dafür wird die Einheitssphäre in Zellen tesseliert, dort die Punkte in Abhängigkeit

ihrer Normalenrichtungen eingetragen und daraus gleichverteilt gezogen. Dies führt dazu, dass die Punkte der essentiellen Details häufiger gezogen werden, da sie meist andere Normalenrichtungen aufweisen, als der weniger relevante Rest von $\mathcal{P}$. (Wei97) verwendet weitere Informationen wie die Farbe oder Intensität, um bewusst saliente Punkte auswählen zu können, für die im nachfolgenden Schritt 2 das Korrespondenzproblem leichter gelöst werden kann.

Es zeigt sich in (RL01), dass die Varianten ohne Intensitätsinformationen bei gut strukturierten Punktmengen eine vergleichbare Konvergenzgeschwindigkeit besitzen. Müssen allerdings in einer großen Punktmenge vor allem wenige feine Details berücksichtigt werden, ist der Ansatz der gleichverteilten Normalen wesentlich robuster.

Schritt 2: Varianten der Korrespondenzzuweisung (BM92) verwendet für die Korrespondenz von $\mathbf{p}' \in \mathcal{P}'$ den nächsten Punkt bzw. nächsten Nachbar (NN) $\mathbf{x}$ im Modell mittels einer ausführlichen Suche. Dies kann mittels einer effizienten Datenstruktur wie einem Oct-Tree bzw. kd-Tree (Sim96) deutlich beschleunigt werden. Allerdings benötigt deren Aufbau eine nicht zu vernachlässigende Zeit. Der NN-Ansatz kann erweitert werden, indem innerhalb eines gewissen Radius um $\mathbf{x}$ kompatible Modellpunkte gesucht werden, für die weitere Informationen übereinstimmen. (Pul99) erlaubt z. B. nur Korrespondenzen, deren Normalen einen Winkel kleiner 45° aufweisen, (Wei97) verwendet Intensitäts- und (GRB94) Farbübereinstimmungen.

Andere Möglichkeiten ergeben sich für Modelle $\mathcal{X}$ mit einer dichten Oberflächenrepräsentation (z. B. Dreiecksnetze) durch Projektion von $\mathcal{P}$ auf $\mathcal{X}$. (CM92) verfolgt daher für jeden transformierten Punkt $\mathbf{p}'$ dessen Normale bis sie auf die Oberfläche von $\mathcal{X}$ trifft. (BL95) benutzt die reverse Kalibrierung, eine perspektivische Projektion von $\mathcal{P}'$ auf $\mathcal{X}$ mit demjenigen Projektionszentrum, das ebenfalls während der Modellgenerierung benutzt wurde.

Nach (RL01) ist die Verwendung von kompatiblen Punkten effizienter und robuster als die gleichen Ansätze ohne Berücksichtigung weiterer Informationen. Die projektiven Verfahren sind zwar schneller als die NN-Ansätze, allerdings bei fein strukturierten Punktmengen (vgl. Abb. 2.6 Mitte) weniger robust.

Schritt 3: Varianten der Gewichtung (BM92; Pul99) verwenden konstante Gewichte für alle Korrespondenzen. (RL01) verwendet entweder den Normalenunterschied $w = \mathbf{n}_{\mathbf{p}'}^{\mathrm{T}} \mathbf{n}_{\mathbf{x}}$ zwischen Modell und aktuell transformierter Punktmenge $\mathcal{P}'$ oder den Abstand zwischen $\mathbf{p}'$ und $\mathbf{x}$ mittels $w = 1 - |\mathbf{p}' - \mathbf{x}|/d_{\max}$ bzgl. des maximalen Abstands $d_{\max}$ aller Punktkorrespondenzen. Andere Möglichkeiten ergeben sich über Farbunterschiede (GRB94) oder der Messunsicherheit der Punkte (RL01).

Es zeigt sich allerdings in (RL01), dass die unterschiedlichen Varianten praktisch keinen Einfluss auf die Konvergenzgeschwindigkeit haben.

Schritt 4: Varianten der Ausreißereliminierung Eine einfache Möglichkeit der Ausreißereliminierung erfolgt über einen Schwellwertvergleich der Gewichte aus Schritt 3. (DWJM98) schließt Korrespondenzen aus, deren Abstand d ein Vielfaches der Standardabweichung über alle Abstände überschritt. (Pul99) verwirft die schlechtesten $l = 10\%$ Korrespondenzen bzgl. eines Qualitätskriteriums wie z. B. d. Weiterhin werden Korrespondenzen verworfen, deren Punkte auf Objektkanten liegen. Ähnlich geht (TL94) vor, der alle Punkte an den Grenzen von $\mathcal{P}$ verwirft, um stärkere Unsicherheiten an den Rändern zu vermeiden. (DWJ98) überprüft die Nachbarschaft von Korrespondenzen und ob deren Punkte kompatibel zu dem Modell sind.

Nach (RL01) konvergieren restriktivere Varianten etwas langsamer, könnten aber einen positiven Einfluss auf die Genauigkeit des Endergebnisses haben.

Schritt 5: Varianten der Transformationsberechnung Erhält man aus den obigen Schritten 1-4 eine Menge von 3D-3D-Punktkorrespondenzen, können die analytisch geschlossenen Verfahren aus 2.2.2.1 benutzt werden (BM92; CM92). Da sie alle dieselbe Gleichung 2.2, die Summe der quadrierten Abstände der korrespondierenden Punkte minimieren, haben sie keinen unterschiedlichen Einfluss auf den ICP-Algorithmus. (ESK99) erweitert obige Ansätze und verwendet ebenfalls Abstände von Farbinformationen. (BM92) stellt ebenfalls eine schnell konvergierende Variante durch Extrapolation der Transformationsparameter über *mehrere* Iterationsschritte vor. Dabei besteht allerdings die Gefahr über das eigentliche Minimum des ICP hinaus zu extrapolieren. (DWJM98) benutzt für Modellrepräsentationen über eine Menge von Ebenen (z. B. Dreiecksnetze) als Metrik den lotrechten Abstand von $\mathbf{p}'$ zur nächstgelegenen Modellebene. Dafür ist allerdings keine geschlossene Lösung möglich.

Um die endgültigen Transformationsparameter zu berechnen, wendet (Sim96) den ICP mehrmals auf leicht veränderte initiale Transformationsparameter $\mathbf{R}^{[0]}$ und $\mathbf{t}^{[0]}$ an. Damit soll die Gefahr lokaler Minima reduziert werden. (DWJM98) geht ähnlich vor, allerdings wird der ICP immer nur auf einer Teilmenge von $\mathcal{P}$ ausgeführt und alle Ergebnisse über einen Median gemittelt.

Solange der Extrapolationsansatz nicht über das Minimum hinaus läuft, besitzt er nach (RL01) die schnellste Konvergenzgeschwindigkeit. Die Ansätze mit multiplen ICP Durchläufen sind dort erwartungsgemäß am langsamsten. Die Punkt-zu-Ebene Metrik hat bei schwierigen Formen Vorteile und konvergiert robuster gegen das Minimum, da dort die Punkte während der Iteration besser auf dem Modell gleiten können.

Zusammenfassung Unter Berücksichtigung aller obigen Varianten gibt (RL01) einen schnellen ICP-Ansatz an. Er verwendet in Schritt 1 das Random Sampling aus (Pul99), nutzt die Punkt-zu-Ebene Metrik (DWJM98), die projektive Korrespondenzsuche aus (BL95), keine Gewichtung und Ausreißereliminierung sowie keine Extrapolation in Schritt 5. Der Ansatz ermöglicht die Registrierung von 10^5 Punkten in 30 ms auf einem Pentium 3 Xeon Prozessor mit 550 MHz bei Auswahl von 10% der Punkte in Schritt 1.

2.2.1.2 Andere Techniken

Eine Vielzahl weiterer globaler Möglichkeiten zur Objektdetektion und Lokalisation findet man in der Literatur. Eine weitere Modellrepräsentation die sich vor allem für Tiefenbilder bzw. Neigungsbilder eignet, sind die *Extended Gaussian Images* (EGI). Sie beschreiben die Oberfläche $\mathcal{X}$ von beliebig geformten 3-D Objekten und sind für konvexe Formen eindeutig. EGI's nutzen eine Abbildung $t : \mathcal{X} \rightarrow \mathcal{S}^2$ von Punkten $\mathbf{x} \in \mathcal{X}$ der Oberfläche auf Punkte $\mathbf{s} \in \mathcal{S}^2$ der Einheitskugel, so dass die normierten Normalen $\mathbf{n}(\mathbf{x}) = \mathbf{n}(\mathbf{s})$ übereinstimmen. Weiterhin benötigt man die gaußschen Krümmungen $\kappa(\mathbf{x}) = \kappa_{\max}(\mathbf{x})\kappa_{\min}(\mathbf{x})$ der Oberflächenpunkte. Diese werden von den Hauptkrümmungen $\kappa_{\max}$ bzw. $\kappa_{\min}$ abgeleitet. Sei $f_{\theta,\mathbf{x}}$ die Kurve, die aus dem Schnitt der Oberfläche $\mathbf{X}$ mit einer Ebene entsteht. Die Ebene enthält $\mathbf{n}(\mathbf{x})$ und wird um den Winkel θ um $\mathbf{n}(\mathbf{x})$ rotiert. Die Hauptkrümmungen ergeben sich dann aus der minimalen und maximalen Krümmung von $f_{\theta,\mathbf{x}}$ in dem betrachteten Punkt $\mathbf{x}$ über alle Winkel. Die zugehörigen Ebenen stehen senkrecht aufeinander und werden als Hauptebenen bezeichnet. Damit lässt sich das egi : $\mathcal{S} \rightarrow R$ definieren (Hor84):

$$\text{egi}(\mathbf{s}) = \sum_{\mathbf{x} \in \mathcal{X}, t(\mathbf{x}) = s} \frac{1}{|\kappa(\mathbf{x})|}$$

Das EGI enthält also an jedem Punkt $\mathbf{s}$ der Einheitssphäre die aufsummierten inversen gaußschen Krümmungen aller Punkte $\mathbf{x}$ der Oberfläche mit derselben Normalenrichtung. Für die praktische Anwendung wird anstatt des kontinuierlichen EGI's eine diskrete Variante $\hat{\text{egi}}(\mathcal{S}_i) = \sum_{\mathbf{s} \in \mathcal{S}_i} \text{egi}(\mathbf{s})$ benutzt, in dem die Oberfläche der Einheitssphäre $\mathcal{S}^2$ möglichst gleichmäßig in disjunkte Zellen $\mathcal{S}_i$ mit

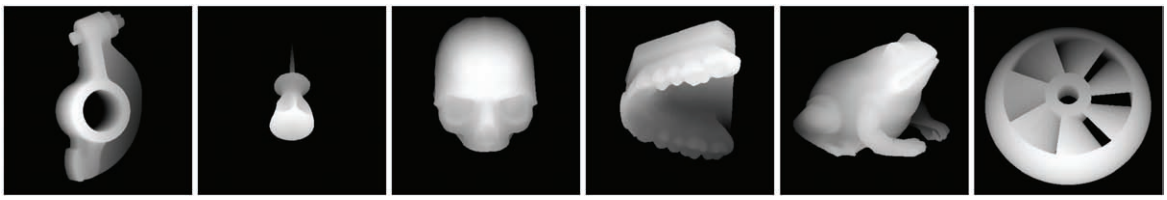

Abbildung 2.7: Synthetisch erzeugte Tiefenansichten aus (HLLS01) zum Testen der Histogramm-Technik.

$\mathcal{S} = \bigcup_i \mathcal{S}_i$ zerlegt wird. Eine interessante Eigenschaft ist, dass der Wert $\hat{\text{egi}}(\mathcal{S}_i)$ dem Flächeninhalt $\mathbf{a}_i$ entspricht, den die Flächen mit den Normalenrichtungen in $\mathcal{S}_i$ auf der Oberfläche $\mathcal{X}$ einnehmen. Insbesondere entspricht $\hat{\text{egi}}(\mathcal{S})$ dem Flächeninhalt der gesamten Oberfläche.

EGI's sind translatorisch invariante Beschreibungen von $\mathcal{X}$ und verhalten sich kovariant zu einer Rotation von $\mathcal{X}$. Sie können daher herangezogen werden, um den Rotationsunterschied zweier Oberflächen zu bestimmen, in dem die Korrelation beider EGI's über die Oberfläche der Einheitskugel minimiert wird.

(KI93) erweitert das EGI zum *Complex Extended Gaussian Image* $\text{cegi} : \mathcal{S} \rightarrow C$ welches ein komplexes Gewicht $\text{cegi}(\mathbf{s}) = a(\mathbf{s})e^{\text{id}(\mathbf{s})}$ für jede Normalenrichtung speichert. $a(\cdot)$ ist dabei wiederum der Flächeninhalt der Oberfläche mit dieser Normalenrichtung und $d(\mathbf{s}) = \mathbf{n}(\mathbf{x})^{\mathrm{T}}(\mathbf{x}-\mathbf{o})$ der Abstand der Tangente an dem Oberflächenpunkt $\mathbf{x} = t^{-1}(\mathbf{s})$ zu dem Ursprung $\mathbf{o}$ der Oberfläche. Da sich $t(\cdot)$ im allgemeinen nicht eindeutig umkehren lässt, muss in diesem Fall über alle Punkte mit unterschiedlichem Abstand komplex aufsummiert werden. Der Betrag $|\text{cegi}(\cdot)|$ ist weiterhin translationsinvariant und kann wie das EGI zur Rotationsbestimmung herangezogen werden. Über die Phase lässt sich nun allerdings bei bekannter Rotation ebenfalls der Translationsunterschied bestimmen. Sowohl das EGI als auch das CEGI sind globale Verfahren und benötigen eine Segmentierung der Punkte. Weiterhin sind sie sehr instabil gegenüber Verdeckungen und lokalen Störungen. (HID95) stellt eine alternative globale sphärische Repräsentation von Oberflächen vor. Sie ist etwas robuster gegenüber Verdeckungen, allerdings auch sehr aufwändig.

Vergleichbar zu den globalen ansichtsbasierten Histogrammtechniken aus Abschnitt 2.1.1.2 schlägt (HLLS01) die Codierung von Tiefenbildern mittels Histogrammen vor. Wie auch bei Intensitätsbildern kann man neben den Tiefenwerten $d(\mathbf{x})$ mit den Normalenrichtungen $\mathbf{n}(\mathbf{x})$ und den Krümmungen $k(\mathbf{x})$ auch die Ableitungen der Bilder als Informationsquelle heranziehen. Die Normalenrichtung $\mathbf{n} = (x,y,z)^{\mathrm{T}}$ wird dabei mit zwei Winkeln $\phi = \arctan2(z,y)$ und $\theta = \arctan2(\sqrt{y^2+z^2},x)$ codiert. Für die Krümmung wird der *Shape Index* (KD92) $k = \frac{1}{2} - \frac{1}{\pi}\arctan2(\kappa_{\max}-\kappa_{\min}, \kappa_{\max}+\kappa_{\min})$ unter Verwendung der Hauptkrümmungen an dem Punkt $\mathbf{x}$ herangezogen. Zum Vergleich zweier normalisierter Histogramme $\mathrm{h}_1(\cdot)$ und $\mathrm{h}_2(\cdot)$ verwendet (HLLS01) nicht die Histogrammschneidung aus Gl. 2.1, sondern die χ^2-Divergenz

$$d_\chi(\mathrm{h}_1,\mathrm{h}_2) = \sum_c \frac{(h_1(c)-h_2(c))^2}{h_1(c)+h_2(c)}$$

Mittels 30 synthetischer Modelle werden für jedes Modell alle 23° - 26° über die gesamte Einheitssphäre Modellansichten mit neuem Blickwinkel generiert (Beispiele siehe Abb. 2.7), so dass insgesamt 1980 Ansichten in der Modelldatenbank enthalten sind. Des Weiteren werden dazwischen 5760 Testansichten generiert und unter Verwendung unterschiedlicher Histogramme gegen die Modelldatenbank gematcht. Verwendet man nur den Shape-Index (k) oder die Normalen (n) als Histogrammeinträge, ergibt sich eine Erkennungsrate von ca. 80%, bei Verwendung der reinen Tiefenwerte (d) nur

noch ca. 43%. Kombiniert man die Informationen allerdings, ergibt sich bei (d+n) ca. 89% und bei (d+n+s) ca. 93% unter Verwendung von $4 \times 4 \times 4 \times 4 = 256$ Einträgen in $\mathbf{h}(\cdot)$. Weiterhin liefert das Verfahren auch eine grobe Lage, für die allerdings keine Ergebnisse präsentiert wurden. Die Laufzeit lag bei 100 ms auf einem Sun Blade 1000 mit 600 MHz, um eine Testansicht gegen die vollständige Modelldatenbank zu matchen. Das Verfahren ist wie bei den ansichtsbasierten Techniken auf eine Segmentierung und eine vollständige Objektansicht angewiesen.

2.2.2 Lokale Ansätze

Zu Beginn dieses Abschnitts werden Verfahren vorgestellt, die bei mehreren gegebenen Korrespondenzen zwischen Modell und Bild eine (Starrkörper-) Transformation ermitteln, die das Modell mit den beobachteten Sensordaten registriert. Je nach Art der Informationen kann es sich dabei um *3D-3D* oder *2D-3D Korrespondenzen* handeln. In den darauf folgenden Abschnitten werden grundlegende Ansätze sowie ihre Anwendungen vorgestellt. Dabei handelt es sich um Verfahren, die auf salienten *3D-Regionen* und ihren Beschreibungen operieren, der *erweiterten Hough-Transformation* und dem *geometrischen Hashing*. Der letzte Abschnitt behandelt dann mit dem *perzeptuellen Gruppieren* und ausgewählten Verfahren zur *Objektverfolgung* weitere Ansätze, die für die Objektlokalisation genutzt werden können.

Wie schon bei den lokalen ansichtsbasierten Verfahren haben die lokalen geometriebasierten Verfahren meist den Vorteil, nicht auf eine Segmentierung oder eine Groblageschätzung angewiesen zu sein. Allerdings sind sie rauschanfälliger und werden daher häufig mit einer genaueren globalen Feinlageschätzung kombiniert.

2.2.2.1 Lageschätzung mit 3D-3D-Korrespondenzen

Gegeben sei eine Menge von n Punkten $\mathbf{p}_i = (x,y,z)^{\mathrm{T}} \in \mathbb{R}^3$ in zwei unterschiedlichen Koordinatensystemen A und B. Dieser Abschnitt behandelt Verfahren, die auf Basis dieser 3D-3D-Punktkorrespondenzen eine Transformation ${}^{\mathrm{A}}\mathbf{p}_i = {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{p}_i + {}^{\mathrm{A}}\mathbf{t}$ zwischen den Koordinatensystemen schätzen. Wie in (Hor87; AHB87; HHN88) gezeigt wird, ist eine optimale geschlossene Lösung durch Minimierung der Quadrate der Fehlerresiduen $\mathbf{e}_i$ über alle Punktkorrespondenzen

$$\sum_{i=1}^{n} \|\mathbf{e}_i\|^2 = \sum_{i=1}^{n} \|\,{}^{\mathrm{A}}\mathbf{p}_i - s\,{}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{p}_i - {}^{\mathrm{A}}\mathbf{t}\|^2 \tag{2.2}$$

möglich. Optional kann auch der Skalierungsunterschied s beider Punktmengen geschätzt werden. Bei einer Starrkörpertransformation wird dagegen $s = 1$ angenommen. Sei $\overline{\mathbf{p}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i$ der Schwerpunkt aller Punkte und ${}^{\mathrm{A}}\mathbf{q}_i = {}^{\mathrm{A}}\mathbf{p}_i - {}^{\mathrm{A}}\overline{\mathbf{p}}$ bzw. ${}^{\mathrm{B}}\mathbf{q}_i = {}^{\mathrm{B}}\mathbf{p}_i - {}^{\mathrm{B}}\overline{\mathbf{p}}$ die Darstellung der Punkte im schwerpunktfreien Bezugssystems. Weiterhin wird die Skalierung s bzw. der Skalierungsfehler symmetrisch auf beide Koordinatensysteme verteilt. Dann erhält man durch Ausmultiplizieren der Quadrate für Gleichung 2.2 die Form (Hor87; HHN88)

$$\sum_{i=1}^{n} \|\mathbf{e}_i\|^2 = \frac{1}{s}\underbrace{\sum_{i=1}^{n} \|\,{}^{\mathrm{A}}\mathbf{q}_i\|^2}_{a} + s\underbrace{\sum_{i=1}^{n} \|\,{}^{\mathrm{B}}\mathbf{q}_i\|^2}_{b} - 2\underbrace{\sum_{i=1}^{n} {}^{\mathrm{A}}\mathbf{q}_i^{\mathrm{T}}\,{}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{q}_i}_{c} + n\underbrace{\|\,{}^{\mathrm{A}}\mathbf{t} - {}^{\mathrm{A}}\overline{\mathbf{p}} + s\,{}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\overline{\mathbf{p}}\|^2}_{d}}$$

Diese Gleichung wird minimal, falls a,b,d minimal und c maximal werden. Ergänzt man quadratisch, erhält man für $s^{-1}a - 2c + sb = (\sqrt{sb} - \sqrt{s^{-1}a})^2 + 2(\sqrt{ab} - c)$ und es ist leicht ersichtlich, dass mit

$$\begin{aligned} s &= \sqrt{ab^{-1}} \\ {}^{\mathrm{A}}\mathbf{t} &= {}^{\mathrm{A}}\overline{\mathbf{p}} - s\,{}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\overline{\mathbf{p}} \end{aligned}$$

die Gleichung 2.2 in Abhängigkeit der optimalen Rotation minimal wird. Diese Rotation lässt sich durch Maximieren von c entweder über eine orthogonale Matrix (AHB87; HHN88) oder ein Einheitsquaternion (Hor87) bestimmen.

Bleibt man in der Matrixschreibweise, lässt sich zeigen, dass sich c in $\mathrm{tr}\,({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\mathbf{M})$ mit $\mathbf{M} = \sum_{i=1}^{n} {}^{\mathrm{A}}\mathbf{q}_i^{\mathrm{T}}\,{}^{\mathrm{B}}\mathbf{q}_i$ umformen lässt. Die einfachste Lösung erfolgt über eine Eigenwertzerlegung (SVD) der Matrix $\mathbf{M} = \mathbf{U}\Lambda\mathbf{V}$ mit $\mathbf{U},\mathbf{V} \in O_3$ und $\Lambda \in \mathcal{S}ym_3$. Sei $\mathbf{V} = (\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3)^{\mathrm{T}}$ und $\mathbf{V}' = (\mathbf{v}_1,\mathbf{v}_2,-\mathbf{v}_3)^{\mathrm{T}}$ dann erhält man die c maximierende Rotationsmatrix nach (AHB87) durch

$$\begin{aligned} {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}} &= \mathbf{V}\mathbf{U}^{\mathrm{T}} \quad \text{falls} \quad |\mathbf{V}\mathbf{U}^{\mathrm{T}}| = 1 \\ {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}} &= \mathbf{V}'\mathbf{U}^{\mathrm{T}} \quad \text{falls} \quad |\mathbf{V}\mathbf{U}^{\mathrm{T}}| = -1 \wedge \lambda_{\min}(\mathbf{M}) = (\Lambda)_{33} = 0 \end{aligned}$$

Der Fall einer negativen Determinante tritt (fast) ausschließlich bei einer komplanaren Konfiguration der Punkte auf, bei der die Matrix $\mathbf{M}$ ein Rangdefizit von 1 hat, d. h. deren kleinster Eigenwert gleich 0 ist. Sollte der Rang der Matrix $\mathbf{M}$ kleiner als 2 sein, liegen die Punkte auf einer Linie und eine eindeutige Lösung ist nicht möglich. Ebenfalls liefert der Algorithmus keine korrekte Lösung, falls bei vollem Rang dennoch $|\mathbf{V}\mathbf{U}^{\mathrm{T}}| = -1$ ist. Dies geschieht nach (AHB87) allerdings nur bei sehr verrauschten Daten, wo schon der Ansatz der kleinsten Fehlerquadrate aus Gleichung 2.2 in Frage gestellt werden muss.

Eine äquivalente Rotation in einer effizienteren Darstellung erhält man bei Betrachtung der Rotation mit einem Einheitsquaternion $\dot{\mathbf{r}}$. Dazu muss man den Ausdruck $c = \sum_{i=1}^{n} \dot{\mathbf{r}}\,{}^{\mathrm{B}}\dot{\mathbf{q}}_i\dot{\mathbf{r}}^*\,{}^{\mathrm{A}}\dot{\mathbf{q}}_i$ maximieren (Hor87). Sei $\dot{\mathbf{q}} = \mathrm{i}x + \mathrm{j}y + \mathrm{k}z$ und

$${}^{\mathrm{A}}\mathbf{Q}({}^{\mathrm{A}}\dot{\mathbf{q}}) = \begin{pmatrix} 0 & -x & -y & -z \\ x & 0 & z & -y \\ y & -z & 0 & x \\ z & y & -x & 0 \end{pmatrix} \qquad {}^{\mathrm{B}}\mathbf{Q}({}^{\mathrm{B}}\dot{\mathbf{q}}) = \begin{pmatrix} 0 & -x & -y & -z \\ x & 0 & -z & y \\ y & z & 0 & -x \\ z & -y & x & 0 \end{pmatrix}$$

dann lässt sich c in eine Matrixschreibweise der Quaternionen-Multiplikation $\mathbf{r}^{\mathrm{T}}(\sum_{i=1}^{n} {}^{\mathrm{B}}\mathbf{Q}_i^{\mathrm{T}}\,{}^{\mathrm{A}}\mathbf{Q}_i)\mathbf{r} = \mathbf{r}^{\mathrm{T}}\mathbf{N}\mathbf{r}$ umschreiben, wenn man das Quaternion $\dot{\mathbf{r}} = r_w + \mathrm{i}r_x + \mathrm{j}r_y + \mathrm{k}r_z$ als Vektor $\mathbf{r} = (r_w,r_x,r_y,r_z)^{\mathrm{T}}$ auffasst. Der maximierende Lösungsvektor ist dann der Eigenvektor $\upsilon_{max}(\mathbf{N})$ zum größten Eigenwert der Matrix $\mathbf{N}$. Dieser lässt sich wiederum über die SVD der Matrix $\mathbf{N}$ bestimmen. Alternativ gibt (Hor87) eine effiziente Lösung zur Berechnung der Nullstellen des charakteristischen Polynoms an, inbesondere für die degenerierten Fälle bei komplanaren Punktkonfigurationen.

2.2.2.2 Lageschätzung mit 2D-3D-Korrespondenzen

Dieser Abschnitt behandelt Verfahren, die auf Basis von wenigen lokalen Punkten bekannter Lage im Objekt-Koordinatensystem und deren korrespondierenden projizierten Punkten in die Bildebene einer Kamera die Transformation zwischen Objekt und Kamera schätzen. Es wird daher auch das perspektivische n Punkte Problem, kurz PnP-Problem, genannt und kann formal wie folgt ausgedrückt werden.

Gegeben sei eine Menge von n Objektpunkten ${}^{\mathrm{O}}\mathbf{p}_i = (x,y,z)^{\mathrm{T}} \in \mathbb{R}^3$ im Objekt-Koordinatensystem O und deren Bildpunkte $\mathbf{q}_i = (u,v)^{\mathrm{T}} \in \mathbb{R}^2$. Gesucht ist dann die Starrkörper-Transformation in das Kamera-Koordinatensystem C, bestehend aus Rotation ${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}$ und Translation ${}^{\mathrm{C}}\mathbf{t}$, die den Fehler $\mathbf{e}_i = \mathbf{q}_i - \mathbf{h}({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t})$ über alle Punkte minimiert. Obwohl es analytische Lösungen (MNLF07) für das Problem gibt, sind iterative Lösungen in der Praxis aufgrund ihrer Robustheit und höheren Genauigkeit vorzuziehen.

Das wohl bekannteste Verfahren zur Lösung des PnP-Problems ist POSIT[6] (DD95). Der Ursprung

[6] engl.: **P**ose from **O**rthography and **S**caling with **IT**erations

des Objekt-Koordinatensystems wird in den ersten Punkt $\mathbf{p}_0$ gelegt, ebenso wird als Abbildungsmodell $\mathbf{h}(\cdot)$ der Kamera anstatt einer perspektivischen Projektion $u_i = f\frac{x_i}{z_i}$ bzw. $u_i = f\frac{y_i}{z_i}$ mit Brennweite f eine skalierte orthographische Projektion[7] (SOP) angenommen, für die $z_i = z_0$ im Kamera-Koordinatensystem C gilt:

$$\begin{aligned}\hat{u}_i &= f\frac{x_i}{z_0} = \frac{f}{z_0}x_0 + \frac{f}{z_0}(x_i - x_0) = u_0 + s(x_i - x_0)\\ \hat{v}_i &= f\frac{y_i}{z_0} = \frac{f}{z_0}y_0 + \frac{f}{z_0}(y_i - y_0) = v_0 + s(y_i - y_0)\end{aligned}$$

Die skalare Größe $s = \frac{f}{z_0}$ ist dabei die Skalierung der SOP. Es lassen sich dann über fundamentale Beziehung der perspektivischen und skalierten orthographischen Projektion folgende Zusammenhänge darstellen (für Beweise siehe (DD95)):

$$\begin{aligned}\frac{f}{z_0}\mathbf{p}_i^{\mathrm{T}}\mathbf{c}_x &= u_i(1+\varepsilon_i) - u_0 = \hat{u}_i - u_0\\ \frac{f}{z_0}\mathbf{p}_i^{\mathrm{T}}\mathbf{c}_y &= v_i(1+\varepsilon_i) - v_0 = \hat{v}_i - v_0\end{aligned}$$

mit $\varepsilon_i = \frac{1}{z_0}\mathbf{p}_i^{\mathrm{T}}\mathbf{c}_z$ und den Einheitsvektoren der Kamera-Koordinatenachsen $\mathbf{c}_x$, $\mathbf{c}_y$ und $\mathbf{c}_z = \mathbf{c}_x \times \mathbf{c}_y$. Ist $\varepsilon_i = 0$ gilt $\mathbf{q}_i = \hat{\mathbf{q}}_i$ und die Punkte $\mathbf{p}_0$ und $\mathbf{p}_i$ liegen in einer Ebene parallel zur Bildebene der Kamera. Man beachte weiterhin, dass ${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}} = ({}^{\mathrm{O}}\mathbf{c}_x, {}^{\mathrm{O}}\mathbf{c}_y, {}^{\mathrm{O}}\mathbf{c}_z)$ gilt, d. h. die gesuchte Rotationsmatrix enthält in den Spalten die Einheitsvektoren der Kameraachsen im Objekt-Koordinatensystem. Der gesuchte Translationsvektor ${}^{\mathrm{C}}\mathbf{t} = {}^{\mathrm{C}}\mathbf{p}_0 = s^{-1}(u_0, v_0, f)^{\mathrm{T}}$ lässt sich leicht über die Skalierung der SOP angeben.

Obige Beziehungen werden in POS ausgenutzt. Sei ε_i (vorerst) bekannt und $\hat{\mathbf{c}}_x = s\mathbf{c}_x$ bzw. $\hat{\mathbf{c}}_y = s\mathbf{c}_y$. Dann lassen sich die Skalarprodukte im Objekt-Koordinatensystem als ${}^{\mathrm{O}}\mathbf{p}_i^{\mathrm{T}}{}^{\mathrm{O}}\hat{\mathbf{c}}_{\mathbf{x}} = \hat{u}_i - u_0$ bzw. ${}^{\mathrm{O}}\mathbf{p}_i^{\mathrm{T}}{}^{\mathrm{O}}\hat{\mathbf{c}}_{\mathbf{x}} = \hat{v}_i - v_0$ ausdrücken, die Objektpunkte als auch die Bildkoordinatengrößen in einer Matrix $\mathbf{A} = ({}^{\mathrm{O}}\mathbf{p}_0, \ldots, {}^{\mathrm{O}}\mathbf{p}_{n-1})^{\mathrm{T}}$ bzw. zwei Vektoren $\mathbf{u} = (\hat{u}_0 - u_0, \ldots, \hat{u}_{n-1} - u_0)^{\mathrm{T}}$, $\mathbf{v} = (\hat{v}_0 - v_0, \ldots, \hat{v}_{n-1} - v_0)^{\mathrm{T}}$ untereinander schreiben und alles in zwei linearen Gleichungssystemen $\mathbf{A}\hat{\mathbf{c}}_x = \mathbf{u}$ bzw. $\mathbf{A}\hat{\mathbf{c}}_y = \mathbf{v}$ ausdrücken. Dies lässt sich nun im Sinne der kleinsten Fehlerquadrate durch ${}^{\mathrm{O}}\hat{\mathbf{c}}_x = \mathbf{Bu}$ bzw. ${}^{\mathrm{O}}\hat{\mathbf{c}}_y = \mathbf{Bv}$ mit der Pseudoinversen $\mathbf{B}$ der Matrix $\mathbf{A}$ lösen. $\mathbf{B}$ wird auch als *Objektmatrix* bezeichnet und ist unabhängig von den beobachteten Bildpunkten. Über die Lösungen ${}^{\mathrm{O}}\hat{\mathbf{c}}_x$ bzw. ${}^{\mathrm{O}}\hat{\mathbf{c}}_y$ kann nun die Skalierung $s = |\hat{\mathbf{c}}_x|$ sowie durch Normierung die Einheitsvektoren in Objekt-Koordinaten bestimmt werden und somit die gesuchte Rotationsmatrix ${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}$ als auch der Translationsvektor ${}^{\mathrm{C}}\mathbf{t}$.

In POS wird ε_i als bekannt angenommen, d. h. man kennt den Korrekturfaktor zwischen den bekannten perspektivischen Bildpunkten $\mathbf{q}_i$ und den benötigten SOP-Punkten $\hat{\mathbf{q}}_i = \mathbf{q}_i(1+\varepsilon_i)$. Da man aber zur Berechnung von $\varepsilon_i = \frac{1}{z_0}\mathbf{p}_i^{\mathrm{T}}\mathbf{c}_z$ die gesuchte Rotationsmatrix ${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}$ und den gesuchten Translationsvektor ${}^{\mathrm{C}}\mathbf{t}$ benötigt, muss man eine Iterationsschleife um das POS-Verfahren aufbauen, den POSIT-Algorithmus. Im ersten Schritt wird $\varepsilon_i^{[0]} = 0$ angenommen, d. h. man initialisiert die POS-Koordinaten mit den bekannten perspektivischen Punkten. Dann wird der Korrekturfaktor $\varepsilon_i^{[k]}$ zur Berechnung der SOP-Punkte mittels der neuen Translationsparameter iterativ verfeinert bis sich das Ergebnis von POS nur noch geringfügig ändert. Dies geschieht nach (DD95) schon in ca. $4-5$ Schritten.

Zur Bestimmung der Objektmatrix $\mathbf{B}$ ist die Pseudoinverse der Matrix $\mathbf{A}$ nötig, die numerisch stabil mittels der SVD bestimmt werden kann. Hat $\mathbf{A}$ allerdings nicht den vollen Rang 3, kann keine eindeutige Lösung mit obigen Verfahren berechnet werden. Geometrisch betrachtet ist dies der Fall,

[7] wird in der Photogrammetrie auch als schwache perspektivische Projektion bezeichnet.

wenn die Punkte $\mathbf{p}_i$ ungünstig im Raum verteilt liegen, entweder in einer Ebene ($\text{rang}(\mathbf{A}) = 2$) oder auf einer Geraden bzw. einem Punkt ($\text{rang}(\mathbf{A}) \leq 1$).

Im Falle von *komplanaren* Punkten gibt (ODD96) eine Erweiterung des POSIT Algorithmus an, bei dem die noch nicht verwendeten Bedingungen $|\hat{\mathbf{c}}_x| = |\hat{\mathbf{c}}_y|$ und $\hat{\mathbf{c}}_x^{\mathrm{T}}\hat{\mathbf{c}}_y = 0$ genutzt werden um das Rangdefizit auszugleichen. Eine geometrische Interpretation des linearen Gleichungssystems zeigt, dass sich die Lösungen $\hat{\mathbf{c}}_\mathbf{x}$ bzw. $\hat{\mathbf{c}}_\mathbf{y}$ jeweils durch den Schnittpunkt einer Menge von Ebenengleichungen in Normalform ergeben. Dieser ist nur bei $\text{rang}(\mathbf{A}) = 3$ eindeutig, im komplanaren Fall $\text{rang}(\mathbf{A}) = 2$ ergibt sich jeweils eine Gerade $\hat{\mathbf{c}}_\mathbf{x} = \hat{\mathbf{c}}_{x,0} + \nu\mathbf{d}$ bzw. $\hat{\mathbf{c}}_\mathbf{y} = \hat{\mathbf{c}}_{y,0} + \mu\mathbf{d}$ als Lösungsraum. Der Richtungsvektor $\mathbf{d} = \upsilon_{\min}(\mathbf{A})$ ist dabei der Eigenvektor zum kleinsten Eigenwert $\lambda_{\min}(\mathbf{A})$ von $\mathbf{A}$. Damit lassen sich jetzt die zwei Bedingungen mit den Ebenenparameter wie folgt ausdrücken:

$$\begin{aligned} -\nu\mu &= \hat{\mathbf{c}}_{x,0}^{\mathrm{T}}\hat{\mathbf{c}}_{y,0} \\ \nu^2 - \mu^2 &= |\hat{\mathbf{c}}_{y,0}|^2 - |\hat{\mathbf{c}}_{y,0}|^2 \end{aligned}$$

Eine elegante Lösung ergibt sich durch Zuhilfenahme der komplexen Zahlen. Sei $a = \nu + \mathrm{i}\mu \in \mathbb{C}$, dann ist $a^2 = \nu^2 - \mu^2 + 2\mathrm{i}\nu\mu = |\hat{\mathbf{c}}_{y,0}|^2 - |\hat{\mathbf{c}}_{y,0}|^2 - 2\mathrm{i}\hat{\mathbf{c}}_{x,0}^{\mathrm{T}}\hat{\mathbf{c}}_{y,0}$ und die Lösung ergibt sich aus der komplexen Wurzel in Polarform. Mit der Länge $\rho = |a^2|$ und dem Winkel $\theta = \frac{1}{2}\arctan2(2\hat{\mathbf{c}}_{x,0}^{\mathrm{T}}\hat{\mathbf{c}}_{y,0}, |\hat{\mathbf{c}}_{y,0}|^2 - |\hat{\mathbf{c}}_{y,0}|^2)$ ergeben sich die zwei Lösungen $\nu_\pm = \pm\rho\cos\theta$ und zugehörigem $\mu_\pm = \pm\rho\sin\theta$. Diese Mehrdeutigkeit entsteht durch die Betrachtung einer SOP anstatt einer perspektivischen Projektion und entspricht einer Spiegelung der Rotation an der Fokalebene der Kamera. Mit dem einfachen Konsistenzcheck, ob mit den erhaltenen Transformationsparameter einige der beobachteten Modellpunkte $\mathbf{p}_i$ hinter der Kamera liegen, kann man in manchen Fällen eine der Lösungen verwerfen. Dennoch würde aber über die iterative POSIT-Struktur ein ganzer Baum von Lösungen entstehen. (ODD96) zeigen experimentell, dass es genügt nur in der ersten Iteration zwischen beiden Lösungen zu unterscheiden und in den darauf folgenden Schritten nur jeweils die beste Lösung weiterzuverfolgen. Als Maß für die Güte wird die Abweichung $\sum_i |\mathbf{q}_i - \mathbf{h}({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t})|$ der in das Bild projizierten Punkte $\mathbf{p}_i$ von den tatsächlich beobachteten Bildpunkten $\mathbf{q}_i$ verwendet.

Ein alternatives Verfahren, der **O**rthogonale-**I**terations-Algorithmus (kurz OI-Algorithmus) wird in (LHM00) vorgestellt. Es hat den Vorteil, dass es auch ohne Fallunterscheidung für komplanare Punkte eine Lösung ermittelt, da im Gegensatz zu POSIT keine schwache sondern eine exakte perspektivische Abbildung modelliert wird. Dazu minimiert der Algorithmus direkt auf dem Raum der orthogonalen Matrizen iterativ kollineare Bedingungen im Objekt-Koordinatensystem durch Überführung des Problems auf einen 3D-3D-Korrespondenz-Algorithmus (vgl. Abschnitt 2.2.2.1). Sei die Brennweite $f = 1$, d. h. die Bildpunkte $\mathbf{q}_i$ sind in normalisierten Bildkoordinaten angegeben, die Rotationsmatrix ${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)^{\mathrm{T}}$ und ${}^{\mathrm{C}}\mathbf{t} = (t_x, t_y, t_z)^{\mathrm{T}}$, dann lässt sich die Beziehung zwischen Objektpunkten ${}^{\mathrm{O}}\mathbf{p}_i$ und den homogenen Bildpunkten $\mathbf{q}_i$ durch die zentrale Kollinearitätsgleichung der Photogrammetrie in der Bildebene angeben:

$$\check{\mathbf{q}}_i = \frac{1}{\mathbf{r}_3^{\mathrm{T}}\,{}^{\mathrm{O}}\mathbf{p}_i + t_z}({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t})$$

Eine direkte Sichtweise der Kollinearität im Objektraum lässt sich über eine Projektion ausdrücken, da sich kollineare Vektoren durch Projektion auf den jeweils anderen Vektor nicht verändern:

$${}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t} = \mathbf{V}_i({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t}) \qquad \text{mit} \qquad \mathbf{V}_i = \frac{\check{\mathbf{q}}_i\check{\mathbf{q}}_i^{\mathrm{T}}}{\check{\mathbf{q}}_i^{\mathrm{T}}\check{\mathbf{q}}_i}$$

Die Matrix $\mathbf{V}$ ist dabei ein Projektionsoperator, der den Objektpunkt ${}^{\mathrm{C}}\mathbf{p}_i = {}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t}$ auf den Sichtstrahl des zugehörigen Bildpunktes $\mathbf{q}_i$ projiziert. Die Abweichung von der projektiven kollinearen Bedingung kann daher herangezogen werden, um ein Fehlermaß $\mathbf{e}_i = (\mathbf{I}_{3\times3} - \mathbf{V}_i)({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}\,{}^{\mathrm{O}}\mathbf{p}_i + {}^{\mathrm{C}}\mathbf{t})$

für die gesuchten Transformationsparameter zu definieren. Drückt man den Translationsvektor $\mathbf{t_R} = {}^{\mathrm{C}}\mathbf{t}({}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}})$ mit Hilfe der Rotationsmatrix $\mathbf{R} = {}^{\mathrm{C}}\mathbf{R}_{\mathrm{O}}$ aus und definiert $\hat{\mathbf{p}}_i = \mathbf{V}_i(\mathbf{R}\,{}^{\mathrm{O}}\mathbf{p}_i + \mathbf{t_R})$, lässt sich das Minimierungsproblem alleinig über $\mathbf{R}$ wie folgt definieren:

$$\mathbf{R}_{\min} = \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\mathbf{e}_i\|^2 = \sum_{i=1}^{n} \|\mathbf{R}\,{}^{\mathrm{O}}\mathbf{p}_i + \mathbf{t_R} - \hat{\mathbf{p}}_i\|^2 \tag{2.3}$$

$$\mathbf{t_R} = \frac{1}{n}\left(\mathbf{I} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{V}_i\right)^{-1} \sum_{i=1}^{n}(\mathbf{V}_i - \mathbf{I})\mathbf{R}\,{}^{\mathrm{O}}\mathbf{p}_i \tag{2.4}$$

Diese Minimierung ähnelt sehr stark der Gleichung 2.2 des 3D-3D-Minimierungsproblems. Leider sind die 3D-Punkte $\hat{\mathbf{p}}_i$ von den gesuchten Transformationsparametern abhängig und es ist daher keine geschlossene Lösung möglich. Formuliert man den Ansatz allerdings iterativ und berechnet $\mathbf{R}^{[k+1]}$ aus den geschätzten Punkten $\hat{\mathbf{p}}_i(\mathbf{R}^{[k]})$ des letzten Iterationsschritts, erhält man nach (LHM00) schon nach $5-10$ Schritten eine global konvergente Lösung. Die Initialisierung erfolgt ähnlich wie in POS mittels einer SOP-Annahme und spielt für die Genauigkeit keine Rolle. Das 3D-3D-Minimierungsproblem wird mittels orthogonaler Zerlegung in Matrixschreibweise (siehe hierfür Abschnitt 2.2.2.1) gelöst.

In (MNLF07) wird mit dem *EPnP*-Algorithmus ein schnelles, nicht iteratives Verfahren vorgestellt, das von allen untersuchten nicht iterativen Verfahren (wie POS) das genaueste ist. Im Vergleich zu den ebenfalls untersuchten iterativen Verfahren sind diese zwar um mindestens eine Größenordnung schneller, aber auch um mindestens eine Größenordnung ungenauer. Das EPnP-Verfahren eignet sich allerdings als erste Lösung im Initialisierungsschritt obiger Verfahren und zeigt dort bessere Ergebnisse als die originalen Initialisierungen. Der OI-Algorithmus wird bei dieser Untersuchung als genauestes Verfahren bewertet, bei einer Laufzeit von ca. 8 ms bei 6 Punkten. Diese kann auf ca. 80% reduziert werden bei Verwendung von EPnP in der Initialisierung, weiterhin scheint dadurch das Verfahren robuster gegenüber Ausreißern zu werden.

Alle vorgestellten Verfahren behandeln allerdings nur eine Kamera, die Lösungen können zwar auch bei mehreren Kameras angewandt werden, aber nicht in einem integrierten Prozess. (CC04) nutzt dagegen die Information aller Korrespondenzen in einem *einzigen* Minimierungsprozess. Dieser besteht aus zwei Komponenten. Für Objektpunkte, die in mindestens zwei Kameras beobachtet werden, werden mittels Triangulation die 3D-Koordinaten im Kamerasystem bestimmt und damit das 3D-3D-Fehlermaß aus Gleichung 2.2 aufgestellt. Für die restlichen Punkte wird das Fehlermaß der Gleichung 2.3 der Kollinearität im Objektraum verwendet und durch ihre vergleichbare Struktur miteinander kombiniert. Für weitere Details sei auf (CC04) verwiesen.

2.2.2.3 Techniken auf Basis von 3D-Regionen

Vergleichbar mit den Verfahren aus Kapitel 3 für Intensitätsbilder gibt es für Punktwolken oder Dreiecksnetze das Bestreben, auf robuste Weise lokale Korrespondenzen zwischen bestimmten Punkten derselben, aber unterschiedlich betrachteten 3D-Strukturen zu finden. Diese Korrespondenzen können dann zur Segmentierung, Erkennung und Lokalisation von 3D-Objekten herangezogen werden und sind aufgrund ihrer lokalen Struktur robust gegenüber Verdeckungen und komplexen Szenen. Die Methoden und grundlegenden Ideen sind meist dieselben, wie bei den intensitätsbasierten 2D-Varianten und können in *Detektoren* (vgl. Abschnitt 3.2) zur Erkennung von geeigneten Punkten bzw. Regionen, in *Deskriptoren* (vgl. Abschnitt 3.3) zur robusten eindeutigen Codierung der Regionen und in *Matching Strategien* (vgl. Abschnitt 3.4) zum Auffinden von Korrespondenzen unterteilt werden.

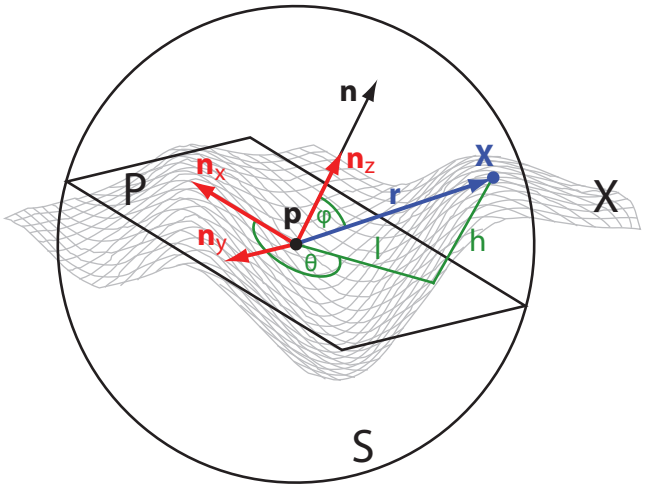

Abbildung 2.8: Geometrische Beziehungen bei der Konstruktion lokaler Beschreibungen im Einzugsbereich einer Kugel $\mathcal{S}$ mit Radius $r_{\mathcal{S}}$ um den Punkt $\mathbf{p}$ mit normalisierter Normale $\mathbf{n}$ einer 3D-Struktur $\mathcal{X}$. Anstatt einer Kugel wird manchmal auch ein an $\mathbf{n}$ ausgerichteter Zylinder verwendet. Jeder Punkt $\mathbf{x} \in \mathcal{C}$ des Supports $\mathcal{C} = \mathcal{S} \cap \mathcal{X}$ kann in verschiedenen *lokalen* Koordinatensystemen repräsentiert werden. Der Vektor $\mathbf{r} = \mathbf{x} - \mathbf{p} = (x,y,z)^{\mathrm{T}}$ beschreibt $\mathbf{x}$ relativ zu dem Ankerpunkt $\mathbf{p}$ im *globalen kartesischen Bezugssystem* (blau). Der Vektor $(n_x, n_y, n_z)^{\mathrm{T}}$ beschreibt $\mathbf{x}$ in einem *lokalen kartesischen Bezugssystem* (rot), das durch eine Ebene $\mathcal{P}$ mit Ankerpunkt $\mathbf{p}$ und Normale $\mathbf{n}_z = \mathbf{n}$ definiert ist. Eine weitere Möglichkeit besteht durch *sphärische Koordinaten* $(r, \varphi, \theta)^{\mathrm{T}}$ mit dem Mittelpunkt der Kugel in $\mathbf{p}$ und dem Nordpol in Richtung $\mathbf{n}$. Damit ergibt sich der Radius $r = |\mathbf{r}|$, der Winkel $\cos\varphi = r^{-1}\mathbf{n}^{\mathrm{T}}\mathbf{r}$ als Innenwinkel zwischen $\mathbf{n}$ und $\mathbf{r}$ und θ als die Rotation um die Normale $\mathbf{n}$ in $\mathcal{P}$. $\mathbf{x}$ kann ebenfalls in *zylindrischen Koordinaten* $(l, h, \theta)^{\mathrm{T}}$ angegeben werden, wobei der Zylinder entlang der Normalen $\mathbf{n}$ ausgerichtet ist. Die Höhe bzw. der Abstand zu $\mathcal{P}$ ergibt sich zu $h = \mathbf{n}^{\mathrm{T}}\mathbf{r}$, der Abstand zur Mittelachse $\mathbf{n}$ des Zylinders zu $l = \sqrt{r^2 - h^2}$. Der verbleibende Freiheitsgrad θ der letzten drei Koordinatensystemdefinitionen kann nicht durch den Ankerpunkt $\mathbf{p}$ und seine Normale $\mathbf{n}$ bestimmt werden und wird von jedem Verfahren unterschiedlich behandelt. Man beachte, dass die letzten drei Koordinatensysteme bei entsprechender Definition von θ translations- und rotationsinvariant konstruiert werden können.

Anhand dieser Unterteilung sollen im folgenden kurz die Methoden zum Auffinden lokaler Korrespondenzen mittels 3D-Informationen vorgestellt und im Anschluss daran konkrete Anwendungen präsentiert werden. Abbildung 2.8 und deren Beschreibung geben einen Überblick über die nachfolgend verwendeten Bezeichnungen und geometrischen Beziehungen.

Detektoren Detektoren haben die Aufgabe diejenigen Ankerpunkte $\mathbf{p} \in \mathcal{X}_\perp \subseteq \mathcal{X}$ zu bestimmen, für die der Deskriptor eine robuste, eindeutige und rotations- und translationsinvariante Beschreibung erzeugen kann. Weiterhin sollte die Menge $\mathcal{X}_\perp$ robust unter einer Starrkörpertransformation von $\mathbf{X}$ wiedergefunden werden. Während für die intenstitätsbasierten 2D-Detektoren ein großer Aufwand getrieben werden muss, um neben den Ankerpunkten auch skalierungsabhängige umschließende Regionen zu definieren, kann dies bei den 3D-Varianten sehr einfach durch Kugeln $\mathcal{S}_r[\mathbf{p}] = \{\mathbf{s} \in \mathbb{R}^3 \mid r > |\mathbf{s} - \mathbf{p}|\}$ mit beliebigen, aber fixen Radius $r_{\mathcal{S}}$ erfolgen. Die Wahl von $r_{\mathcal{S}}$ ist dabei stark von der Auflösung und Komplexität der 3D-Struktur $\mathbf{X}$ abhängig und ist ein Tradeoff zwischen Eindeutigkeit der Strukturen und Empfindlichkeit gegenüber lokalen Störungen.

Die einfachste, aber auch ineffizienteste Methode besteht darin, alle Punkte $\mathcal{X}_\perp = \mathcal{X}$ der 3D-Struktur zu verwenden (JH99; BAGC05). Diese Vorgehensweise hat den Nachteil, dass eine Vielzahl

von Regionen betrachtet werden, die aufgrund ihrer lokalen Struktur (z. B. eine Ebene) nicht unterscheidbar sind und daher zur Korrespondenzfindung völlig ungeeignet sind. (SM92) verwendet daher Punkte in der Nähe von Kanten, da sie eine saliente lokale 3D-Struktur aufweisen. Es werden allerdings nur Punkte an *inneren* Objektkanten betrachtet, da für ihre umschließende Kugel $\mathcal{S}_r[\mathbf{p}]$ die Gefahr der Überschneidung mit anderen Objekten geringer ist. (YFEB99) definiert Salienz mittels dem Simplexwinkel $\beta_\mathbf{p}$, der ein Maß für die Krümmung in der Umgebung eines Punktes $\mathbf{p}$ ist. $\beta_\mathbf{p}$ wird mittels des Simplex-Netzes berechnet, der Dualform des Dreiecksnetzes einer 3D-Struktur, die jedem Punkt $\mathbf{p}$ genau drei Nachbarn $\mathbf{x}_1$, $\mathbf{x}_2$ und $\mathbf{x}_3$ zuordnet. Diese drei Punkte definieren einen Kreis mit Radius r_{123} und alle vier Punkte eine Kugel mit Radius $r_{\mathbf{p}123}$. Damit ergibt sich der Simplexwinkel zu

$$\sin\beta_\mathbf{p} = \frac{r_{123}}{r_{\mathbf{p}123}} \operatorname{sgn}\left(\mathbf{n}^\mathrm{T}(\mathbf{x}_1 - \mathbf{x})\right) \tag{2.5}$$

der eng mit der mittleren Krümmung $\sin\beta_\mathbf{p}/r_{123}$ am Punkt $\mathbf{p}$ verwandt ist. Die Menge der detektierten Punkte ergibt sich dann mittels einem Schwellwert τ zu $\mathcal{X}_\perp = \{\mathbf{p} \in \mathbf{X} \mid \sin\beta_\mathbf{p} > \tau\}$.

Deskriptoren Deskriptoren beschreiben den lokalen Ausschnitt $\mathcal{C}_\mathbf{p} = \mathcal{S}_r[\mathbf{p}] \cap \mathcal{X}$ der 3D-Struktur für alle detektierten Ankerpunkte $\mathbf{p} \in \mathcal{X}_\perp$ auf eindeutige, robuste, translations- und rotationsinvariante Weise in einem n dimensionalen Merkmalsvektor $\mathbf{m} \in \mathbb{R}^n$. Alle hier vorgestellten Deskriptoren definieren ein lokales Koordinatensystem mit $\mathbf{p}$ als Ursprung das an der Normalen $\mathbf{n}(\mathbf{p})$ ausgerichtet ist (vgl. Abb. 2.8). Bis auf einen verbleibenden Freiheitsgrad, der Rotation θ um die Normale des Ankerpunkts können die Punkte des Supports $\mathcal{C}$ damit auf translations- und rotationsinvariante Weise dargestellt werden. Die Verfahren unterscheiden sich einerseits in Codierung als auch in der Behandlung von θ.

Spin-Images (SI) (JH99) nutzen die zylindrische Repräsentation (l,h,θ), ohne den Freiheitsgrad θ explizit zu bestimmen. Stattdessen wird ein Histogramm $\mathrm{si}(\hat{h},\hat{l})$ erzeugt, indem die durch l und h aufgespannte Ebene um die Zylinderachse bzw. Normale $\mathbf{n}$ des Ankerpunkts rotiert wird. Diese Ebene wird in k^2 gleichmäßige Zellen mit Mittelpunkt $(\hat{h},\hat{l})$ aufgeteilt und alle Punkte des zylindrischen Supports $\mathcal{C}$ werden anhand ihrer l- und h-Koordinaten binomial gesampelt in die 4 benachbarten Zellen eingetragen. Die Quantisierungsgenauigkeit $k = 15$ sollte dabei an die Auflösung von $\mathcal{X}$ angepasst werden. Über die maximalen Werte l_{max} und $h_{\max}$ des Supports kann die Lokalitätseigenschaft der Spin-Images bestimmt werden. Eine weitere Option besteht, nur Punkte $\mathbf{x} \in \mathcal{C}$ zu benutzen, deren Normale $\mathbf{n}(\mathbf{x})$ einen Innenwinkel $\cos\alpha = \mathbf{n}^\mathrm{T}\mathbf{n}(\mathbf{x}) < \cos\alpha_\tau = 60^\circ$ zur Normalen des Ankerpunkts hat. Damit sollen Probleme mit Selbstverdeckungen reduziert werden. Der k^2 dimensionale Merkmalsvektor $\mathbf{m}(\mathrm{si})$ besteht dabei aus den Einträgen der einzelnen Zellen des SI.

(BAGC05) erweitert die SI auf die *texturierten Spin-Images* (TSI) und bezieht für jeden Punkt $\mathbf{x} \in \mathcal{C}$ neben der 3D-Lage auch die Intensitätsinformation $\mathrm{i}(\mathbf{x})$ mit ein. Das Histogramm $\mathrm{tsi}(\hat{h},\hat{l},\hat{i})$ muss dabei um eine Dimension erweitert werden. Aufgrund der größeren Eindeutigkeit kann $k = 5$ reduziert und der Intensitätsbereich grob in $k_i = 4$ Zellen unterteilt werden.

(RCSM01) führt eine Transformation $\gamma : \mathbb{R}^n \to \mathcal{S}^{n-1}$ ein, um einen n dimensionalen Merkmalsvektor $\mathbf{m} = (m_1,\ldots,m_n)^\mathrm{T}$ aus dem kartesischen Raum auf die eingebettete n-1 dimensionale Oberfläche der Einheitskugel $\mathcal{S}^{n-1}$ abzubilden. γ ist definiert als

$$\gamma(\mathbf{m}) = \frac{\mathbf{m} - \mu(\mathbf{m})}{|\mathbf{m} - \mu(\mathbf{m})|} \qquad \mu(\mathbf{m}) = (\underbrace{\mu_\mathbf{m},\ldots,\mu_\mathbf{m}}_{n-\mathrm{mal}}) \qquad \mu_\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n} m_i \tag{2.6}$$

Damit kann auf effiziente Weise die Korrelation zwischen zwei Merkmalsvektoren berechnet werden (siehe nachfolgender Paragraph). (RCSM01) nutzt diese Transformation, um die SI in die *sphärischen Spin-Images* (SSI) umzuwandeln. Ein SSI ist definiert als der Vektor $\mathbf{m}(\mathrm{ssi}) = \gamma(\mathbf{m}(\mathrm{si}))$.

(SM92) nutzt ebenfalls die zylindrische Repräsentation um einen *Splash*-Deskriptor zu definieren. Ein Splash sind die Normalen $\mathbf{n}(\mathbf{x})$ eines Kreisschnittes $\mathbf{C}_l = \{\mathbf{x} = (l_\mathbf{x}, h_\mathbf{x}, \theta_\mathbf{x})^\mathrm{T} \in \mathbf{C} \,|\, l_\mathbf{x} = l\}$ um den Ankerpunkt $\mathbf{p}$ mit Radius l innerhalb des Supports $\mathbf{C}$. Die Normalen $\mathbf{n}(\mathbf{x})$ werden relativ zur Normalen $\mathbf{n}$ des Ankerpunktes mittels zweier Winkel $\cos\alpha(\mathbf{x}) = \mathbf{n}^\mathrm{T}\mathbf{n}(\mathbf{x})$ und $\cos\psi(\mathbf{x}) = \mathbf{n}_x^\mathrm{T}\mathbf{n}_{xy}(\mathbf{x})$ dargestellt. α ist dabei wiederum der Innenwinkel beider Normalen, ψ der Winkel zwischen der $\mathbf{n}_x$-Achse und der Projektion $\mathbf{n}_{xy}(\mathbf{x})$ des Vektors $\mathbf{n}(\mathbf{x})$ auf die Ebene $\mathcal{P}$. Die Punkte $\mathbf{x} \in \mathbf{C}_l$ lassen sich auch durch den Winkel θ beschreiben, so dass man mittels $\alpha(\theta) = \alpha(\mathbf{x}(\theta))$ und $\psi(\theta) = \psi(\mathbf{x}(\theta))$ die Menge der Normalen als eine Kurve im 3D-Raum mit Achsen θ, α und ψ ansehen kann. Der noch nicht bestimmte Freiheitsgrad θ wird nun bzgl. eines Referenzwinkels $\theta_\mathrm{max} = \mathrm{argmax}_\theta \sqrt{\alpha(\theta)^2 + \psi(\theta)^2}$ normiert, an dem der Abstand der Kurve zur θ-Achse maximal wird. Die Kurve wird berechnet, indem θ diskretisiert wird und an die einzelnen Punkte Polygone angefittet werden. Die Eigenschaften dieser Polygone werden dann als Merkmalsvektor $\mathbf{m}_l$ des Splashes herangezogen. Zur eindeutigen Beschreibung der lokalen Region werden k Merkmalsvektoren m_{l_i} von Splashes mit unterschiedlichem Radius l_i zu dem endgültigen Merkmalsvektor $\mathbf{m}$ kombiniert. Die Splash-Repräsentation ist ähnlich wie ein lokales EGI (siehe hierzu Abschnitt 2.2.1.2) mit dem Unterschied, dass nicht die instabilere Gaußkrümmung sondern die Normalenrichtung herangezogen wird.

Die in (CR97) beschriebene *Point Signature* (PS) verwendet ebenfalls die zylindrische Darstellung der Punkte $\mathbf{x} = (l_\mathbf{x}, h_\mathbf{x}, \theta_\mathbf{x})^\mathrm{T} \in \mathcal{C}$, allerdings wird die Normale $\mathbf{n}$ nicht lokal, sondern durch einen Ebenenfit $\mathcal{P}$ über alle Punkte innerhalb des Supports $\mathcal{C}$ bestimmt. Die Punkte werden dann anhand ihrer h und θ Koordinaten in ein Histogramm $\mathrm{ps}(\hat{h}, \hat{\theta})$ eingetragen, wobei der Freiheitsgrad θ ähnlich wie bei den Splashes bzgl. eines Referenzwinkels $\theta_\mathrm{max} = \mathrm{argmax}_\theta h(\alpha)$ normiert wird. Dieser definiert sich durch den Winkel des Punktes, der den weitesten Abstand h zur Ebene $\mathcal{P}$ hat.

Auch die *Surface Signatures* (SS) aus (YFEB99) verwenden ein Histogramm $\mathrm{ss}(\hat{r}, \hat{\varphi})$, allerdings werden hier Kugelkoordinaten (r, φ, θ) zur Repräsentation der Punkte $\mathbf{x} \in \mathcal{C}$ verwendet. Weiterhin wird in dem Histogramm nicht die Existenz von Punkten über alle Winkel θ akkumuliert, sondern der Simplex-Winkel $\beta_\mathbf{x}$ aus Gl. 2.5.

Ähnlich geht auch (FHK$^+$04) bei dem *3D Shape Context* (SC) vor, der eine Erweiterung des 2D Shape Context (BMP02) ist. Er verwendet ebenso ein Histogramm $\mathrm{sc}(\hat{r}, \hat{\varphi}, \hat{\theta})$ in Kugelkoordinaten, wobei die Winkel in k gleichmäßige Bereiche und der Radius r auf einer logarithmischen Skale in k_r Abschnitte unterteilt ist. Ebenso werden nur Punkte mit einem Radius $r > r_\mathrm{min}$ berücksichtigt. Die Vorgehensweise führt zu einer höheren Robustheit gegenüber Störungen am Rande des Supports und Fehlern in der Normalen des Ankerpunkts. Jeder Punkt wird mittels eines Gewichtes $w(\mathbf{x})$ in der Zelle akkumuliert, in der er bzgl. seiner Koordinaten fällt. Das Gewicht $w(\mathbf{x}) = (\rho(\mathbf{x})\sqrt[3]{v(\mathbf{x})})^{-1}$ berechnet sich durch die Dichte ρ um den Punkt $\mathbf{x}$ (geschätzt durch die Anzahl der Punkte in einer Kugel um $\mathbf{x}$) und dem Volumen v der Zelle, in die der Punkt $\mathbf{x}$ fällt. Anstatt den Winkel θ zu normieren, werden k um $\theta_k = \frac{2\pi}{k}$ gedrehte Versionen des Histogramms in einer Datenbank abgelegt.

Da dieses Vorgehen nicht sehr effizient ist, wird mit dem *Harmonic Shape Context* (HSC) von (FHK$^+$04) ebenso eine vollständige rotationsinvariante Repräsentation vorgestellt. Ausgangspunkt ist die Darstellung $\mathrm{sc}(\hat{r}, \hat{\varphi}, \hat{\theta})$, für die, für alle diskreten Radii $\hat{r}$, eine durch $\mathrm{sc}_r(\hat{\varphi}, \hat{\theta})$ abschnittsweise definierte Funktion $f_r(\varphi, \theta)$ nach sphärischen harmonischen Basisfunktionen $\xi_i^j(\varphi, \theta)$ entwickelt wird. Dies erfolgt durch die Transformation

$$f_r(\varphi, \theta) = \sum_{i=0}^{\infty} \sum_{j=-l}^{j=l} a_i^j \xi_i^j(\varphi, \theta)$$

Der Betrag $|a_i^j|$ der komplexen Koeffizienten ist dabei invariant bzgl. θ. Weiterhin gilt für reelle Funktionen $f(\cdot)$ $a_i^j = a_i^{-j}$. Daher werden die bandbegrenzten Beträge der Koeffizienten $|a_i^j|$ mit $i < b$ und $j \geq 0$ aller Funktionen $f_r(\cdot)$ mit verschiedenen Radien für den $n = \frac{1}{2} k_r b(b+1))$ dimensionalen

Merkmalsvektor **m** herangezogen. Man beachte, dass diese Darstellung unabhängig von der Zellenauflösung des SC in den Winkeldimensionen ist.

Die SI sind mittlerweile zu einem Standardverfahren zur lokalen Korrespondenzfindung in 3D-Strukturen geworden und dementsprechend oft werden neue Verfahren damit verglichen. In Experimenten von (FHK$^+$04) performen der SC vor dem SI am besten. Der HSC schneidet dort insbesondere bei komplexen Szenen am schlechtesten ab. Allerdings hat der SC durch seine Kovarianz bzgl. θ einen *k*-mal so hohen Aufwand während der Korrespondenzfindung. (BAGC05) zeigt, dass bei Vorhandensein von Texturinformationen die TSI selbst bei nur $5 \times 5 \times 4 = 100$ Dimensionen wesentlich eindeutiger sind als die SI mit $15 \times 15 = 225$ Einträgen. (RCSM01) stellt einen leichten Vorteil der SSI bzgl. der SI bei der Korrespondenzfindung fest.

Matching Strategien Die Matching Strategien unterscheiden sich im verwendeten Ähnlichkeitsmaß und der Datenstruktur, die zur Findung des ähnlichsten Merkmalsvektors innerhalb einer Datenbank benutzt werden. In beiden Fällen ist die Aufgabenstellung exakt dieselbe wie bei den ansichtsbasierten Varianten und die gleichen Ideen und Methoden kommen zum Einsatz. Sie werden in Abschnitt 3.4 im Detail behandelt und sollen hier nur noch aufgezählt werden.

Die meisten Methoden (JH99; BAGC05) nutzen die euklidische Distanz als Ähnlichkeitsmaß zwischen Merkmalsvektoren. Alternativ dazu verwendet (RCSM01) den Winkel zwischen zwei SSI's auf der Einheitskugel, der eng verwandt mit dem Korrelationskoeffizienten ist.

Als Datenstruktur verwenden die meisten Verfahren (JH99; RCSM01) die effiziente Datenstruktur aus (NN96) oder ein Hashing-Verfahren (SM92; FHK$^+$04). Um die Dimension des Suchraums zu reduzieren wendet (JH99) die PCA-Kompression auf SI's und (RCSM01) die Kompression mittels *Random Projection* auf SSI's an.

Anwendungen Normalerweise haben bei allen Verfahren (JH99; BAGC05; RCSM01; SM92) die gleiche Vorgehensweise. Für eine Menge an Tiefenansichten bzw. kompletten 3D-Strukturen werden mittels der Detektoren die Ankerpunkte bestimmt, mit diesen die Deskriptoren berechnet und alle Merkmalsvektoren in einer auf die NN-Suche optimierten Datenbankstruktur als Modell abgelegt. Die gleiche Vorgehensweise erfolgt für eine eingegebene Tiefenansicht bzw. 3D-Struktur. Allerdings können jetzt die Merkmalsvektoren herangezogen werden, um lokale Korrespondenzen in der Datenbank zu finden. Diese Korrespondenzen werden gruppiert und für jede Gruppe eine Starrkörpertransformation bzgl. des Modells bestimmt. Diese wird eventuell mittels eines ICP-Ansatzes auf der lokal segmentierten Punktmenge verfeinert (JH99; SM92). Diejenigen registrierten Punktmengen mit dem größten Überlappbereich mit dem Modell werden dann als erfolgreich detektiert bzw. lokalisiert zurückgegeben.

(JH99) erreicht mit 100 Testszenen à 4 Objekten eine Erkennungsrate von 90% bei Verwendung von SI bzw. 83% bei Verwendung von auf 10% PCA-komprimierten SI.

2.2.2.4 Generalisierte Hough Transformation

Die *generalisierte Hough Transformation* (GHT) (Bal81) ist eine der frühesten Algorithmen zur Detektion von beliebig geformten Kurven in Gradientenbildern. Sie ist eine Erweiterung der *Hough Transformation* (HT) (Hou62) durch Ausnutzung zusätzlicher lokaler Informationen wie die Orientierung von Gradienten. Im weiteren wird kurz die Idee der HT und GHT zur Detektion von Kurven umrissen und danach Eigenschaften, Modifikationen und Anwendungen diskutiert.

Die HT wurde ursprünglich als effiziente, diskrete Variante der *Radon Transformation* zur Detektion von Geraden entwickelt (siehe Abb. 2.9), kann aber für alle analytischen Kurven der Form $f(\mathbf{u},\mathbf{p}) = 0$ angewendet werden. Der Vektor $\mathbf{u} = (u,v)^{\mathrm{T}}$ entspricht dabei den Koordinaten im Bild und der Vektor $\mathbf{p} = (p_1,\ldots,p_k)^{\mathrm{T}}$ einem Satz von Parametern zur Definition der Kurve. Die HT nutzt

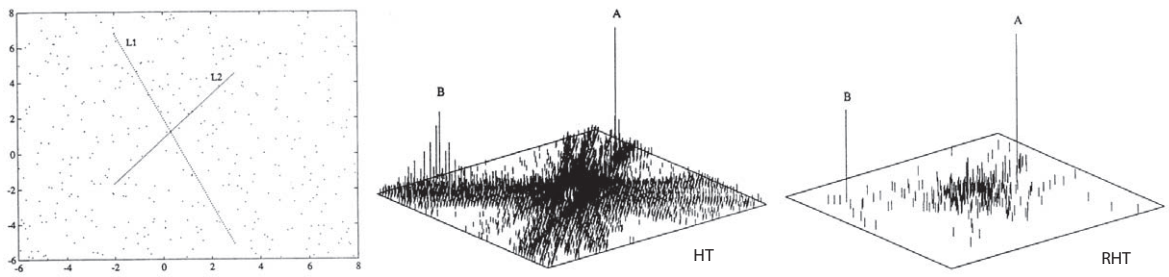

Abbildung 2.9: Eingabebild (links) mit zwei Geraden L_1 und L_2 sowie Salz-und-Pfefferrauschen. In der Mitte ist der zugehörige Hough-Raum (HT) mit den zwei Parameter-Maxima A und B abgebildet, die die Geraden beschreiben. Rechts werden die gleichen Maxima mittels der Randomized Hough Transform (RHT) wesentlich effizienter (für dieses Bsp. ca. 20 mal weniger Einträge) und mit einem besseren Verhältnis zwischen Maxima und zufälligen Überlagerungen ermittelt. Bilder adaptiert aus (XO93).

nun die Dualität zwischen der Kurve $f(\cdot)$ im Bild und deren Punkt $\mathbf{p}$ im Parameter- bzw. Hough-Raum aus. Sei $g(\mathbf{u}) = |\mathbf{g}(\mathbf{u})|$ der Gradientenbetrag bzw. die Stärke der Kante des Bildes an der Stelle $\mathbf{u}$ und

$$a[\hat{\mathbf{p}}] = \sum_{f(\mathbf{u},\hat{\mathbf{p}})=0} g(\mathbf{u})$$

ein Akkumulatorarray über den quantisierten Parameterraum, dann entsprechen Maxima von $a[\cdot]$ Ausprägungen der Kurve $f(\cdot)$ mit Parametersatz $\mathbf{p}_{\max} = \operatorname{argmax}_{\hat{\mathbf{p}}} a[\hat{\mathbf{p}}]$. $a[\cdot]$ wird mit eine Schleife über alle Bildpunkte und $k-1$ Parameter ermittelt. Der verbleibende Parameter kann dann über die Bedingung $f(\mathbf{u},\mathbf{p}) = 0$ berechnet werden. Diese Vorgehensweise ist allerdings nicht sehr effizient, da $O(nm^{k-1})$ Einträge in das Akkumulatorarray erfolgen, wenn das Bild n Pixel enthält und jede Dimension des Hough-Raums in m Zellen quantisiert wird. Ebenfalls ist aufgrund der vielen überflüssigen Einträge das Verhältnis zwischen Maxima der Kurven und zufälligen Überlagerungen im Hough-Raum ungünstig.

Die GHT nutzt daher nicht nur den Betrag des Gradienten $\mathbf{g}(\mathbf{u}) = (\mathrm{i}_{[u]}(\mathbf{u}), \mathrm{i}_{[v]}(\mathbf{u}))^{\mathrm{T}}$, sondern auch die ebenfalls messbare Orientierung $\phi(\mathbf{u}) = \arctan2(\mathrm{i}_{[v]}, \mathrm{i}_{[u]}) + \frac{\pi}{2}$ von Kantenpunkten aus. Damit lässt sich eine zusätzliche Bedingung $f_{[u]}(\mathbf{u},\mathbf{p}) = 0$ definieren, welche die Komplexität um einen weiteren Freiheitsgrad auf $O(nm^{k-2})$ einschränkt. Dadurch kann sowohl die Effizient als auch das Verhältnis zwischen Maxima und zufälligen Überlagerungen deutlich verbessert werden, allerdings zeigt (GH90b), dass für komplexe Szenen immer noch eine hohe Wahrscheinlichkeit zufälliger großer Überlagerungen in $a[\cdot]$ besteht. Dies ist u. a. darauf zurückzuführen, dass die Verwendung der Orientierung eine hohe Rauschanfälligkeit aufweist und daher Maxima der Kurve aufgeweitet werden.

Prinzipiell lässt sich die obige Vorgehensweise wiederholen und mittels der zweiten Ableitung $f_{[uu]}(\mathbf{u},\mathbf{p}) = 0$ eine weitere Bedingung definieren. Allerdings sind dafür die Krümmungen (MC88) oder relative Orientierungsinformationen (SS94) der Gradienten notwendig, welche aufgrund ihrer Rauschanfälligkeit und ihres größeren Einzugsbereichs in der Praxis nicht mehr robust aus Bildern extrahiert werden können. Einen völlig anderen Ansatz für weitere Bedingungen kann man für ausgewählte Punkte über die lokale Bildstruktur erhalten (Low99) (siehe dazu Abschnitt 2.1.2.2).

Die HT benötigt analytisch modellierbare Kurven $f(\mathbf{u},\mathbf{p}) = 0$, die je nach Bedingung nach dem Parameter p_k bzw. unter Verwendung der Orientierung auch nach p_{k-1} aufgelöst werden müssen. Sie wird daher meist nur für einfache Kurven angewendet, wovon die wichtigsten im Folgenden genannt sind.

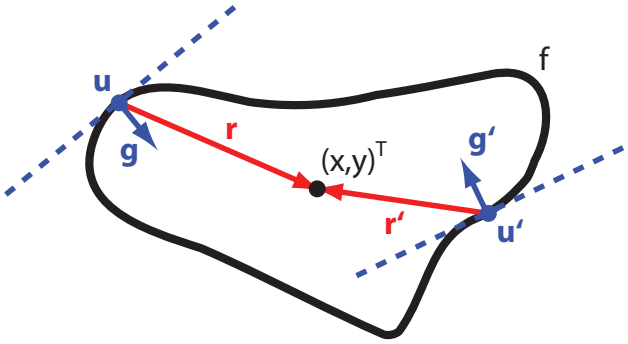

Abbildung 2.10: Parameter der generalisierten Hough Transformation für eine beliebige Kurve f mit Referenzpunkt $(x,y)^{\mathrm{T}}$. Blau sind für zwei Punkte $\mathbf{u}$ und $\mathbf{u}'$ auf der Kurve der Gradientenvektor $\mathbf{g}$ bzw. $\mathbf{g}'$ abgebildet, aus dem jeweils der Betrag $g(\cdot)$ und die Richtung $\phi(\cdot)$ abgeleitet werden, sowie in Rot die zugehörigen Vektoren $\mathbf{r}$ bzw. $\mathbf{r}'$ zum Referenzpunkt der Kurve, die in $\mathcal{R}[\phi(\mathbf{u})]$ bzw. $\mathcal{R}[\phi(\mathbf{u}')]$ eingetragen werden.

Gerade Eine Gerade wird durch die Kurve $f(\mathbf{u},\mathbf{p}) = u\cos\alpha + v\sin\alpha + t = 0$ mit dem Parametervektor $\mathbf{p} = (\alpha,t)^{\mathrm{T}}$ definiert. Anstatt der Steigung wird der Winkel α zur Parametrierung der Orientierung genutzt, da dieser im Bereich $[0,2\pi[$ liegt und äquidistant quantisiert werden kann.

Kreis Ein geschlossener Kreis mit Mittelpunkt $(x,y)^{\mathrm{T}}$ und Radius r lässt sich mittels $f(\mathbf{u},\mathbf{p}) = (u-x)^2 + (v-y)^2 - r^2 = 0$ angeben. Je nach Aufgabe kann der Hough-Akkumulator so ausgelegt werden, dass man Kreise mit fester Größe oder Radien in einem bestimmten Bereich finden kann.

Ellipse Verallgemeinert man den Kreis zu einer Ellipse mit großer und kleiner Halbachse c bzw. d und Orientierung α, ergibt sich folgende Funktion:

$$f(\mathbf{u},\mathbf{p}) = \frac{((u-x)\cos\alpha + (v-y)\sin\alpha)^2}{1 - \left(\frac{(v-y)\cos\alpha - (u-x)\sin\alpha}{d}\right)^2} - c^2 = 0$$

Der Parametervektor $\mathbf{p} = (x,y,c,d,\alpha)^{\mathrm{T}}$ hat dabei schon $k = 5$ Dimensionen und dementsprechend aufwändig ist die HT bzw. GHT.

Am Bsp. der Ellipse sieht man, dass selbst für relativ einfache Kurven die Komplexität sowohl in deren Beschreibung als auch in k schnell steigt und die HT an ihre Grenzen gelangt. Die GHT bezieht sich daher nicht auf eine analytische Modellierung, sondern approximiert $f(\cdot)$ durch eine Lookup-Tabelle $\mathcal{R}$, die für alle Punkte $\mathbf{u}$ der Kurve in Abhängigkeit ihrer quantisierten Gradientenorientierung $\phi(\mathbf{u})$ den Vektor $\mathbf{r}$ zu einem Referenzpunkt $\mathbf{x}$ enthält (siehe Abb. 2.10):

$$\mathcal{R}[\phi] = \{\mathbf{r} \,|\, \mathbf{r} = \mathbf{u} - \mathbf{x} \wedge \phi(\mathbf{u}) = \phi\}$$

Der Vektor $\mathbf{r}$ kann dabei als Vektor $\mathbf{p} = (x,y)^{\mathrm{T}}$ im Parameter- bzw. Hough-Raum angesehen werden, der in dieser einfachen Form nur die Verschiebung einer beliebigen Kurve $f(\cdot)$ modelliert. Die $\mathcal{R}$-Tabelle kann aber auf jede beliebige parametrisierbare Transformation $\mathbf{t} : \mathbb{R}^2 \to \mathbb{R}^2$ der Kurve erweitert werden. Sei $\mathbf{p} = (\mathbf{r}^{\mathrm{T}}, \mathbf{p}_{\mathbf{t}}^{\mathrm{T}})^{\mathrm{T}}$ der Vektor im Hough-Raum bestehend aus der Verschiebung und der zusätzlichen Transformation mit Parametern $\mathbf{p}_{\mathbf{t}}$, dann ergibt sich die $\mathcal{R}_{\mathbf{t}}$-Tabelle zu

$$\mathcal{R}_{\mathbf{t}}[\mathrm{t}_\phi(\phi)] = \{\mathbf{p} \,|\, \mathbf{p} = (\mathbf{t}[\mathbf{p}_{\mathbf{t}}](\mathbf{r})^{\mathrm{T}}, \mathbf{p}_{\mathbf{t}}^{\mathrm{T}})^{\mathrm{T}} \wedge \mathbf{r} \in \mathcal{R}[\phi]\}$$

Die $\mathcal{R}_\mathbf{t}$-Tabelle enthält über alle (quantisierten) Transformationsparameter $\mathbf{p_t}$ die transformierten Vektoren $\mathbf{t}[\mathbf{p_t}](\mathbf{r})^\mathrm{T}$ zu dem Referenzpunkt. Sollte die Transformation $\mathbf{t}(\cdot)$ die Kurve so verformen, dass sich die Gradientenorientierungen ϕ ändern, muss dies mittels $\mathrm{t}_\phi : \mathbb{R} \to \mathbb{R}$ berücksichtigt werden. Übliche Transformationen sind die Skalierung $\mathbf{t}[s](\mathbf{r}) = s\mathbf{r}$ mit $\mathrm{t}_\phi(\phi) = \phi$, die Rotation $\mathbf{t}[\alpha](\mathbf{r}) = \mathbf{R}[\alpha]\mathbf{r}$ mit $\mathrm{t}_\phi(\phi) = \phi + \alpha$ oder Kombinationen davon. Die Berechnung des Hough-Akkumulators erfolgt dann formal über

$$a[\hat{\mathbf{p}}] = \sum_{\hat{\mathbf{p}} \in \mathcal{R}_\mathbf{t}[\phi(\mathbf{u})]} g(\mathbf{u})$$

und wird in der Praxis über eine Schleife im Bild berechnet, für die an jeder Stelle $\mathbf{u}$ alle Parameter $\mathbf{p}$ aus dem Tabelleneintrag $\mathcal{R}_\mathbf{t}[\phi(\mathbf{u})]$ in a eingetragen werden. Die Komplexität beträgt dann $O(nl)$, wobei l die durchschnittliche oder größte Anzahl an Einträgen $|\mathcal{R}_\mathbf{t}[\phi(\mathbf{u})]|$ pro quantisierter Gradientenorientierung $\phi(\mathbf{u})$ bezeichnet. Eine feinere Quantisierung von ϕ reduziert zwar einerseits die Komplexität, da weniger Kantenpunkte in den selben Tabelleneintrag fallen, erhöht allerdings auch die Rauschanfälligkeit, da höhere Anforderungen an die Gradientenorientierung gestellt werden müssen. (Bal81) schlägt daher vor, entweder die extrahierte Orientierung und die $\mathbf{r}$-Vektoren mit einer Unsicherheit $\phi \pm \Delta_\phi$ bzw. $\mathbf{r} \pm \Delta_\mathbf{r}$ zu versehen und dementsprechend mehr Akkumulatoreinträge zu erhöhen oder vor der Maximasuche den Akkumulator z. B. mit einem Gaußkernel zu glätten.

Genauigkeit, Performance und Speicherbedarf hängen sowohl bei der HT als auch der GHT direkt von der Quantisierung des Parameter- bzw. Hough-Raums ab. Wird diese m-mal feiner quantisiert, kann $\mathbf{p}_\mathrm{max} = \mathrm{argmax}_{\hat{\mathbf{p}}}\, a[\hat{\mathbf{p}}]$ und damit die Lage der Kurve m-mal genauer bestimmt werden, allerdings steigt auch der Speicherbedarf des Hough-Akkumulators um den Faktor m^k und die Rechenzeit um m^{k-1} (HT) bzw. m^{k-2} (GHT). Dies führt schon bei Dimensionen $k > 4$ zu großen Problemen und schränkt den Anwendungsbereich der (generalisierten) Hough Transformation stark ein.

Um diese Problematik in den Griff zu bekommen, schlägt (USBE03) eine hierarchische Auswertung der GHT und eine subquantisierungsgenaue Maximabestimmung im Houghakkumulator vor. Ausgangspunkt ist eine pyramidale Darstellung des Trainings- als auch des Suchsbildes. Auf der obersten Stufe j der Pyramide erfolgt sowohl die Konstruktion der $\mathcal{R}_\mathbf{t}$-Tabelle als auch deren Auswertung analog zur GHT, allerdings mit einer geringen Quantisierung des Parameterraums. Daher sind zwar einerseits die lokalen Maxima $\mathbf{p}_{\mathrm{max},j} \pm \Delta_j$ mit einer hohen Unsicherheit versehen, andererseits ist die Auswertung effizient und benötigt einen geringen Speicherbedarf. Auf den darunter liegenden Pyramidenstufen $j-1,\ldots,0$ greifen nun 3 Mechanismen, um das Vorwissen aus der vorangegangen Stufe nutzen zu können. Erstens wird für jedes propagierte Maximum nur ein Akkumulator im Bereich der Unsicherheit konstruiert und damit eine Menge Speicher eingespart. Zweitens kann über das Vorwissen (und damit eine grobe Lagekenntnis der Kurve) eine ortsabhängige $\mathcal{R}_\mathbf{t}$-Tabelle ausgewertet werden, so dass weniger Einträge pro Gradientenorientierung ϕ vorhanden sind. Dafür wird schon während der Trainingsphase über die Kurve ein grobes quadratisches Raster gelegt und für jede Zelle des Rasters eine separate $\mathcal{R}_\mathbf{t}$-Tabelle angelegt. Als letztes können alle Kantenpixel eliminiert werden, die unter Berücksichtigung der groben Lagekenntnis nicht zur Kurve gehören können. Dafür wird ebenfalls während des Trainings die Kurve im Bereich der Unsicherheit transformiert, so dass ein verschmiertes Abbild entsteht, welches dann als Maske während der Auswertung verwendet werden kann.

Je niedriger die Pyramidenstufe, desto eingeschränkter wird der Hough-Raum und damit die Unsicherheit und desto höher kann man daher die Quantisierung wählen. Für eine genaue Lageschätzung ist dies selbst auf der untersten Stufe meist noch nicht ausreichend, daher wird im Anschluss jedes Maximum verfeinert, in dem lokal eine k-dimensionale quadratische Funktion in den Hough-Raum eingefittet wird. Der Scheitelpunkt definiert dann die subquantisierungsgenaue Lage des Maximums bzw. der Kurve.

Diese Vorgehensweise benötigt zwar ca. 5-10 mal mehr Speicherbedarf für das Modell (die $\mathcal{R}_t$-Tabellen und die Masken) als die gewöhnliche GHT. Dieser Aufwand verringert sich allerdings deutlich um den Faktor $20-1000$ für deren Auswertung. Ebenfalls sinkt die Komplexität um die gleiche Größenordnung. Dieser Ansatz war eine Grundlage für die Objekterkennung mittels *Shape-Based-Matching* der Bildverarbeitungsbibliothek Halcon (MVT10) und wird sehr erfolgreich für die 2D-Objekterkennung unter Berücksichtigung einer Ähnlichkeitstransformation eingesetzt.

(BW91) stellt einen Mechanismus vor, um mittels der GHT Objekte zu finden, die aus zwei Starrkörpern aufgebaut sind und entweder über ein rotatorisches oder lineares Gelenk miteinander verbunden sind. Untersucht wird nur eine 2D-Starrkörpertransformation (Translation und Rotation) an einfachen Haushaltsgegenständen wie eine Schere oder eine Zange. Für rotative Gelenke wird der Referenzpunkt beider Teilobjekte in den Drehpunkt gelegt, für lineare Gelenke auf dessen Verschiebungsachse. Die Auswertung erfolgt analog zur normalen GHT, allerdings wird nun bei der Maximasuche überprüft, ob es für ein Maxima des einen Teilobjekts an derselben Position ein Maxima des anderen Teilobjekts mit anderer Rotation gibt (rotatorisches Gelenk) bzw. auf der rotatorisch angepassten Verschiebungsachse ein Maxima mit gleicher Orientierung zu finden ist (lineares Gelenk). Der Algorithmus unterliegt den selben Randbedingungen wie die GHT. Es werden nur visuelle Ergebnisse präsentiert.

Um die inhärenten Schwächen der GHT zu umgehen, schlägt (FLK96) eine Kombination der GHT mit der *Randomisierte Hough Transformation* (RHT) vor. Die RHT wird in (XOK90) als Verbesserung der HT vorgestellt, um insbesondere die Problematiken der von der Quantisierung des Hough-Raums abhängigen Genauigkeit, der Beschränkung des diskreten Hough-Raums, der enorme Speicherplatz des Akkumulators und der schlechten Effizient bzw. des schlechten Verhältnis zwischen Maxima und zufälligen Überlagerungen zu beheben (vgl. Abb. 2.9). Anstatt an jedem Punkt $\mathbf{u}$ alle möglichen Parameterkombinationen $\mathbf{p}$ der Kurve $f(\mathbf{u},\mathbf{p})=0$ in den diskreten Akkumulator $a[\hat{\mathbf{p}}]$ einzutragen, wird ein effizienteres *Random Sampling* Verfahren genutzt. Es werden l-mal m unterschiedliche Punkte $\mathbf{u}_{ij}$ mit $0<i<l$ und $0<j<m$ aus der Menge der binären Kantenpunkte gleichwahrscheinlich gezogen. m ist dabei so groß, dass mittels der m Gleichungen $f(\mathbf{u}_{ij},\mathbf{p}_i)=0$ die Funktion $f(\cdot)$ nach den Parametersatz $\mathbf{p}_i$ aufgelöst werden kann. Die ermittelten Parameter $\mathbf{p}_i$ definieren l eindeutige Lagen der Kurve aus einer minimalen Anzahl m an zufälligen Punkten. Ist l groß genug, kann unter diesen Parametern nach Clustern bzw. lokalen Maxima gesucht werden, die dann die robuste Lage einer globalen Kurve beschreiben. Diese Vorgehensweise benötigt weder einen quantisierten noch einen beschränkten Houghakkumulator und hat daher eine unendliche feine Auflösung und Reichweite der Parameter. Ebenfalls ist sie wesentlich effizienter und hat ein besseres Verhältnis zwischen Maxima und zufälligen Überlagerungen, insbesondere bei großem k.

Dieses Prinzip wird in der *Randomisierten Generellen Hough Transformation* (RGHT) (FLK96) auf die GHT übertragen. Es werden für zwei gleichwahrscheinlich gezogene Punkte $\mathbf{u}_1=(u_1,v_1)^{\mathrm{T}}$ und $\mathbf{u}_2=(u_2,v_2)^{\mathrm{T}}$ zwei zufällige Vektoren $\mathbf{r}_1=(x_1,y_1)^{\mathrm{T}}$ und $\mathbf{r}_2=(x_2,y_2)^{\mathrm{T}}$ aus den zugehörigen Einträgen $\mathcal{R}[\phi(\mathbf{u}_1)]$ bzw. $\mathcal{R}[\phi(\mathbf{u}_2)]$ bestimmt. Diese definieren zwei Gleichungssysteme

$$\begin{aligned}\mathbf{p} &= (u_1,v_1,0,0)^{\mathrm{T}}+s_x(x_1,0,1,0)^{\mathrm{T}}+s_y(0,y_1,0,1)^{\mathrm{T}}\\ \mathbf{p} &= (u_2,v_2,0,0)^{\mathrm{T}}+s_x(x_2,0,1,0)^{\mathrm{T}}+s_y(0,y_2,0,1)^{\mathrm{T}}\end{aligned}$$

welche den Parametervektor $\mathbf{p}=(x,y,s_x,s_y)^{\mathrm{T}}$ mit den gezogenen Punkten verknüpft. Über diese Gleichungen können erst die zwei Skalierungsparameter s_x bzw. s_y mittels

$$s_x=\frac{u_2-u_1}{x_1-x_2} \qquad\qquad s_y=\frac{v_2-v_1}{y_1-y_2}$$

bestimmt werden und damit über den Referenzpunkt $(x,y)^{\mathrm{T}}$ der vollständige Parametervektor $\mathbf{p}$. Nach l Wiederholungen kann dann wie bei der RHT in der Menge der Parametervektoren Cluster bestimmt werden, die dann Ausprägungen der Kurve definieren. Das Verfahren kann analog zur GHT auf Rotationen erweitert werden. Es werden nur visuelle Ergebnisse präsentiert.

2.2.2.5 Geometrisches Hashing

Geometrisches Hashing (GH) baut auf lokalen Merkmalen von Objekten auf, die im Gegensatz zu den Ansätzen aus Abschnitt 2.1.2 nicht durch Metainformationen (wie der Textur) beschrieben werden, sondern alleinig durch ihre geometrischen Beziehungen untereinander. Es muss kein Korrespondenzproblem gelöst werden und eignet sich daher für Objektklassen, für die bis auf die Merkmalslagen keine weiteren robusten Informationen zur Verfügung stehen. Es wird für die Objekterkennung in Bildern erstmalig in (LW88; LSW88) angewendet. Im weiteren Verlauf wird zuerst die grundlegende Idee und deren inhärente Grenzen vorgestellt und im Anschluss daran konkrete Anwendungen. Eine gute Einführung gibt ebenfalls (WR97).

Das Prinzip des Geometrischen Hashings kann mit unterschiedlichen Objektmerkmalen angewendet werden, es lässt sich aber am einfachsten an Punkten zeigen. Da bis auf deren Lage keinerlei weiteren Informationen zur Verfügung stehen, muss diese zur Identifizierung der Punkte herangezogen werden. Die Lage, d. h. die Koordinaten eines Punktes sind abhängig von dem gewählten Koordinatensystem und der zu berücksichtigenden Transformation des Objekts zu verschiedenen Beobachtungszeitpunkten. Eine invariante Darstellung der Koordinaten lässt sich daher nur durch ein festgelegtes, objektfixes Koordinatensystem realisieren, welches aber ebenfalls die Lösung der Lageschätzung des Objektes ist. Beliebige objektfixe Koordinatensysteme lassen sich allerdings aus einer minimalen Anzahl von Basispunkten auf dem Objekt konstruieren und die restlichen Merkmalspunkte relativ dazu darstellen. Da man allerdings keine Möglichkeit der Identifikation der Basispunkte hat, werden in einem Trainingsschritt alle möglichen Kombinationen der beobachtbaren n Objektpunkte als Basis verwendet und die restlichen Punkte mittels ihrer relativen Koordinaten als Index zusammen mit der verwendeten Basis in einer Hashtabelle eingetragen (vgl. Abbildung 2.11). Sei m die minimal benötigte Anzahl von Punkten zur Definition einer Basis unter der zu berücksichtigenden Objekttransformation, dann hat dieser Schritt eine Komplexität von $O(n^{m+1})$. Während der Detektion werden die k beobachteten Punkte ebenfalls relativ zu einer gewählten Basis dargestellt, an der jeweiligen Stelle der Hashtabelle überprüft, ob ein oder mehrere Einträge vorhanden sind und gegebenenfalls für die jeweiligen Modellbasen abgestimmt[8]. Sobald für die aktuell gewählte Basis eine der Modellbasen eine genügend hohe Trefferanzahl erreicht hat, kann man annehmen, dass es sich um das gleiche Objekt und auch um die gleichen Punkte der Basis handelt. Im schlimmsten Fall, d. h. wenn alle Kombinationen zur Basiskonstruktion ausprobiert werden müssen, hat die Erkennung eine Komplexität von $O(k^{m+1})$. Aus den gefundenen Korrespondenzen der Basis und der Treffer in der Hashtabelle kann nun eine robuste Lage z. B. mit den Verfahren aus Abschnitt 2.2.2.1 bzw. 2.2.2.2 bestimmt werden.

Diese Vorgehensweise ist robust gegenüber Verdeckungen und lokalen Störungen, solange genügend Punkte ($k \gtrsim m+3$ (LSW88)) eines Objektes zu sehen sind. Ebenfalls können ohne Performanceeinbuße während der Detektion auch mehrere Objekte in der Hashtabelle eingetragen sein. Die Performance bricht allerdings aufgrund der hohen Anzahl irrelevanter Punkte bei komplexem Hintergrund ein. Gleiches gilt für mehrere Objekte in der Szene, da ohne eine Segmentierung die Wahrscheinlichkeit einer gültigen Basis, definiert durch Punkte *eines* Objekts drastisch sinkt.

[8] Im Unterschied zur (generalisierten) Hough Transformation aus Abschnitt 2.2.2.4, wo nicht für eine Objektrepräsentation mit zugehöriger Basis sondern für eine Menge von Transformationsparametern im Hough-Raum abgestimmt wird.

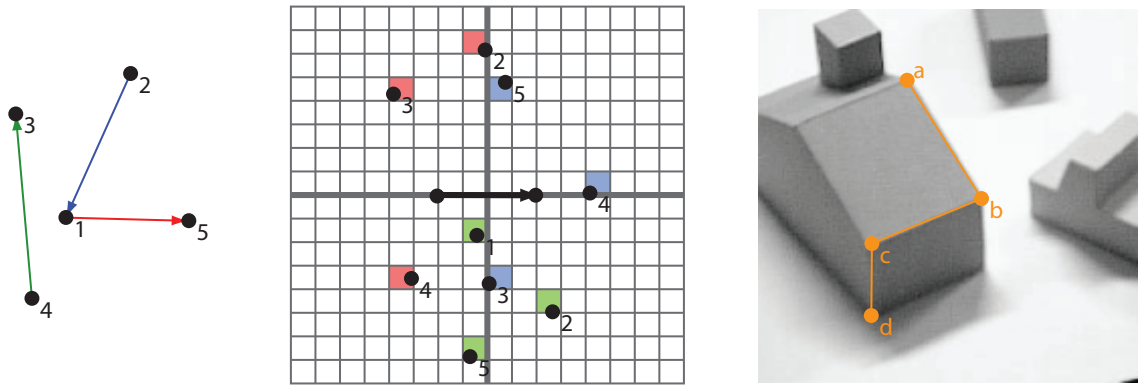

Abbildung 2.11: Ausgewählte Einträge der Tabelle beim Geometrischen Hashing für die linke Punktkonfiguration bei Verwendung einer Ähnlichkeitstransformation und unterschiedlicher Punktmengen für die Basis. Die verschiedenen Basen (roter, grüner oder blauer Vektor) werden verschoben, skaliert und gedreht auf den Vektor $(-\frac{1}{2},\frac{1}{2})^{\mathrm{T}}$ (schwarz) der mittigen Hashtabelle (Zellenseitenlänge $\frac{1}{4}$) abgebildet, die verbleibenden Punkte relativ dazu dargestellt und in der Tabelle zusammen mit der Basis und optionalen weiteren Informationen abgelegt. Rechts ist ein typisches Bauteil für das geometrische Hashing aus (PI97) mit eingezeichnetem Kantentrippel (gelb) abgebildet.

Sowohl die Komplexität des Trainings als auch der Objekterkennung hängt stark von der modellierten Objekttransformation $\mathbf{geo}(\cdot)$ bzw. von der minimalen Anzahl dafür benötigter Punkte für die Basis ab. Es soll daher im Folgenden ein Überblick über die wichtigsten Transformationen und deren Basiskonstruktion gegeben werden. Dabei sei $\mathbf{q}_i \in \mathcal{M}$ eine Menge von Merkmalspunkten und $\mathbf{p}_1,\ldots,\mathbf{p}_m \in \mathcal{M}$ eine geordnete Teilmenge von $\mathcal{M}$ an Punkten zur Basiskonstruktion.

Translation in 2D und 3D Für die Transformation $\mathbf{geo}(\mathbf{q}) = \mathbf{q}+\mathbf{t}$ genügt $m = 1$, da mit $\mathbf{q}' = \mathbf{q}-\mathbf{p}_1$ eine Basistransformation realisiert werden kann.

Ähnlichkeitstransformation in 2D Die Transformation $\mathbf{geo}(\mathbf{q}) = \mathbf{S}\mathbf{q}+\mathbf{t}$ mit $\mathbf{S} \in \mathcal{T}^{+}_{S,2}$ spielt eine wichtige Rolle, wenn eine Abbildung als skalierte orthographische Projektion (SOP) modelliert wird und Objekte bzgl. der Abbildungsebene nicht verkippen (vgl. Abschnitt 3.1.2). Für sie genügt $m = 2$. Eine Basistransformation erhält man durch $\mathbf{q}' = \mathbf{S}_2^{-1}\mathbf{q}-\mathbf{p}_1-\frac{1}{2}$, wobei $\mathbf{S}_2 = (\mathbf{p},\mathbf{p}_\perp)$ mit $\mathbf{p} = (x,y)^{\mathrm{T}} = \mathbf{p}2-\mathbf{p}1$ und $\mathbf{p}_\perp = (y,-x)^{\mathrm{T}}$. Die zusätzliche Verschiebung um $\frac{1}{2}$ führt zu ausgeglicheren Hashtabellen um den Ursprung.

Affine Transformation in 2D Handelt es sich um ebene Objekte, nimmt man eine SOP an und lässt Verkippungen zu, kann man Projektionen der Starrkörpertransformationen in 3D mit der affinen 2D-Transformation $\mathbf{geo}(\mathbf{q}) = \mathbf{A}\mathbf{q}+\mathbf{t}$ mit $\mathbf{A} \in \mathcal{T}^{+}_{A,2}$ beschreiben (vgl. Abschnitt 3.1.2). Hierfür sind $m = 3$ Punkte für die Basis notwendig, deren Transformation sich aus $\mathbf{q}' = \mathbf{A}_{23}^{-1}\mathbf{q}-\mathbf{p}_1-\frac{1}{2}$ mit $\mathbf{A}_{23} = (\mathbf{p}_2-\mathbf{p}_1,\mathbf{p}_3-\mathbf{p}_1)$ ergibt.

Homografietransformation in 2D Modelliert man anstatt einer SOP eine perspektivische Abbildung, muss man anstatt einer affinen eine Homografietransformation berücksichtigen. Hierfür sind $m = 4$ Punkte für die Basis notwendig.

Starrkörpertransformation in 3D Stehen vollständige 3D-Informationen zur Verfügung, benötigt man $m = 3$ Punkte für die Basis um die physikalische Bewegung $\mathbf{geo}(\mathbf{q}) = \mathbf{R}\mathbf{q}+\mathbf{t}$ mit $\mathbf{R} \in \mathcal{SO}_3$ eines Starrkörpers zu berücksichtigen.

(LSW88) berücksichtigt die affine Transformation, um planare Oberflächen von Werkzeugen und industriellen Bauteilen zu erkennen. Es werden unterschiedliche Merkmale wie Punkte und Linien verwendet. Die Punkte eines erfolgreichen Matches werden anschließend verwendet, um die Transformation mit einem Ansatz der kleinsten Fehlerquadrate zu verfeinern. Es werden Szenen mit mehreren Objekten untersucht, wobei leider nur visuelle Ergebnisse präsentiert werden.

(LW88) setzt geometrisches Hashing für die Erkennung von polyedrischen metallischen Bauteilen unter perspektivischer Transformation und beliebigen Blickwinkeln ein. Dafür werden drei Strategien präsentiert. Erstens die Zerlegung der Bauteile in komplanare Bereiche, die mittels einer affinen Transformation oder Homografie separat erkannt werden, zweitens das Voten in einer 3D-Hashtabelle entlang der Sichtstrahlen bei angenommener affinen Kamera und drittens das Einlernen von vielen verschiedenen Ansichten eines Objektes. Die dritte Strategie wird bevorzugt, da zwar der Trainingsschritt zum Aufbau der Hashtabelle komplexer wird, der Detektionsschritt allerdings seine 2D Komplexität behält. Da nur eine Ähnlichkeitstransformation berücksichtigt wird, werden die Ansichten alle 10° eingelernt und die Hash-Zellen erweitert, um die perspektivischen Approximationen in Zwischenansichten auszugleichen. Die Lagen erfolgreicher Matches werden wiederum mittels der kleinsten Fehlerquadrate verfeinert, ebenfalls werden Ergebnisse benachbarter Modell-Ansichten mit geringeren Votes zur Verifikation herangezogen. Es werden keine quantitativen Ergebnisse präsentiert.

(LH94) führt attributive Informationen bei den Einträgen in die Hash-Tabelle ein, um inkorrekte Treffer besser erkennen zu können. Dafür werden Eckpunkte oder Mittelpunkte von Linien als Merkmale benutzt und die Orientierung der Linien als Attribute. Weiterhin wird, abgeleitet von einer bayesschen Modellierung, eine gewichtete Abstimmung in Abhängigkeit der Koordinaten- und der Orientierungsunterschiede zwischen Punktkandidaten und Hash-Einträgen vorgenommen. An simulierten Ergebnissen zeigt sich, dass die Eckpunkte robuster sind als die Mittelpunkte von Linien, beide Ansätze aber bzgl. des Standardverfahrens Verbesserungen liefern. Da allerdings bei den Eckpunkten pro Linie zwei Merkmale generiert werden, ist die Hashtabelle durch die $O(n^3)$ Komplexität ca. 8 mal größer als bei Verwendung der Mittelpunkte. Auf realen Daten von Autos und industriellen Bauteilen war der Ansatz robust gegenüber komplexem Hintergrund bei reiner 2D-Lageschätzung mittels einer Ähnlichkeitstransformation, hatte allerdings Probleme bei kleinen Objekten aufgrund der erhöhten relativen Lageungenauigkeit der extrahierten Merkmale. Es wurden ca. 5-13 Mittelpunkte bzw. 10-26 Eckpunkte pro Objekt eingelernt und verwendet.

Der Ansatz der attributiven Merkmale wird in (Liu96) auf den *universellen* Merkmalsraum erweitert, um verschiedene Objektmerkmale und Attribute fusionieren zu können. Vergleichbare Informationen (wie Koordinaten von Eckpunkten und Multieckpunkten) werden dabei auf gleiche Bereiche des universellen Merkmalsvektors abgebildet, nicht vereinbare Informationen (wie Koordinaten von Eck- und Mittelpunkten) auf unterschiedliche Bereiche. Weiterhin wird zwischen *essentiellen* Bereichen unterschieden, die benötigt werden um eine Basis zu definieren und *annotativen* Attributen. Das geometrische Hashing wird dann inklusive gewichtetes bayessches Voten mittels des universellen Vektors durchgeführt und berücksichtigt somit auf integrierte Weise ohne weitere Modifikation verschiedenste Merkmale und Attribute. Weiterhin wird ein kd-Tree verwendet um einen effizienten Hashzugriff auch bei unbalanzierten Hashtabellen zu ermöglichen. Leider werden keine aussagekräftigen Ergebnisse präsentiert.

Geometrisches Hashing ist sensitiv gegenüber instabilen Merkmalslagen (GH90a), daher verwendet (PI97) anstatt Punktmerkmale etwas robuster extrahierbare Kantentrippels (vgl. Abb. 2.11 rechts). Sie bestehen aus 4 Punkten $\mathbf{a}, \mathbf{b}, \mathbf{c}$ und $\mathbf{d}$, die durch drei Grauwertkanten $\overline{\mathbf{ab}}, \overline{\mathbf{bc}}$ und $\overline{\mathbf{cd}}$ im Bild miteinander verbunden sind. Als Index für die Hashfunktion werden die Winkel β und γ zwischen den Kanten an den Punkten **b** und **c** verwendet. Die Winkel haben weiterhin den Vorteil, dass sie in-

variant gegenüber der Ähnlichkeitstransformation sind. Eine Basis wird implizit definiert durch die Kantenverfolgung, daher ist sowohl für das Training als auch die Detektion keine Basiswahl mehr notwendig und der Aufwand des Hashings reduziert sich auf $O(n)$ bzw. $O(k)$. Die Merkmalsextraktion ist dagegen komplexer. Experimente wurden für 15 einfarbige Polyedermodelle in Bildern unter Berücksichtigung einer Ähnlichkeitstransformation durchgeführt und das entwickelte Verfahren mit dem Standardverfahren verglichen. Die Laufzeit des Standardverfahrens lag bei ca. $31.5-38\,\mathrm{s}$ und hatte eine Erkennungsrate von ca. $26.3-32.0\%$, wohin gegen mit Kantentrippels eine deutlich geringe Laufzeit von $2.6-3\,\mathrm{s}$ und höhere Erkennungsrate von ca. $47.4-55.3\%$ erreicht werden konnte.

Geometrisches Hashing kann auch mit lokalen affinen Frames (LAF's, vgl. Abschnitt 2.1.2.1) durchgeführt werden (CM06). Zur Konstruktion der LAF's werden die Regionengrenzen von affinen MSER-Merkmalen (MCUP04) (vgl. auch Abschnitt 3.2.4.1) verwendet. Diese Grenzen können kovariant extrahiert werden, solange Intensitätsdiskontinuitäten in den Bildern erhalten bleiben. Es ist daher bzgl. Beleuchtungsschwankungen robuster, die Regionengrenzen zum Indizieren zu verwenden als die klassischen Merkmalsvektoren der Deskriptoren (vgl. Abschnitt 3.3), die im Allgemeinen nur eine affine Beleuchtungsinvarianz garantieren. LAF's haben den weiteren Vorteil, dass schon $m = 1$ LAF genügt, um eine affine Basis zu definieren und daher die Komplexität des Verfahrens gering gehalten wird. Die Grenzen benachbarter LAF's werden dann wie beim geometrischen Hashing üblich in der neuen Basis dargestellt und über ein Histogramm in Polarkoordinaten beschrieben. Daher enthalten die Hasheinträge zusätzliche Informationen (vergleichbar mit den Attributen in (Liu96)), welche Fehlmatches verhindern. Es werden nur die l nächsten LAF's derselben Skala miteinander in Beziehung gesetzt. Die Komplexität des Training- und Detektionsschritts reduziert sich daher auf $O(ln)$ bzw. $O(lk)$ und macht diesen Ansatz sehr effizient. Das Verfahren wurde experimentell eingesetzt, um komplanare Objekte in einer Bilddatenbank zu suchen und Bilder verschiedener Spektren (aber gleicher Intensitätsdiskontinuitäten) zu registrieren. Es werden allerdings nur visuelle Ergebnisse präsentiert.

2.2.2.6 Andere Techniken

(Low87) stellt ein 6 DoF Lokalisationsverfahren auf Basis von Intensitätsbildern vor, in dem *perzeptuelle Organisation* ausgenutzt wird, um Gruppen und Strukturen von Liniensegmenten zu finden, die eine hohe Wahrscheinlichkeit der Invarianz gegenüber Blickwinkeländerungen haben. Abb. 2.12 links zeigt die drei verwendeten Arten perzeptueller Gruppen: Parallelität, Kollinearität und Endpunkt Nähe.

Das Verfahren kann in fünf Schritte unterteilt werden. Zu Beginn wird ein Laplace-Kantenfilter auf ein Eingabebild angewendet. Dessen Ausgabe wird genutzt, um zusammenhängende gerade Kantenelemente zu segmentieren. Danach werden Ausprägungen der drei verschiedenen Arten perzeptueller Gruppen innerhalb der Kantensegmente gesucht (vgl. Abb. 2.12 rechts). Diese sind über einen großen Blickwinkelbereich stabil und werden genutzt um gegen ein Kantenmodell zu matchen. Um den Aufwand in Grenzen zu halten, werden nur perzeptuelle Gruppen mit mind. 3 Segmenten genutzt und nach ihrer Ausprägungsgüte sortiert. Für die Zuordnung werden mittels des Kantenmodells alle möglichen Ausprägungen von perzeptuellen Gruppen vorberechnet und in einer Datenbank abgelegt. Bei Übereinstimmung einer perzeptuellen Gruppe mit einer Modellgruppe wird die aktuelle Lage des Modells mit Hilfe einer Newton-Minimierung auf Kantenbasis berechnet. Jede der gefundenen Lagen wird verifiziert, in dem die Übereinstimmung der projizierten Modellkanten mit dem Kantenbild ermittelt werden. Diese Vorgehensweise ist robust gegenüber Verdeckungen, lokale Störungen und komplexem Szeneninhalt. Abb. 2.12 Mitte zeigt ein Ergebnis dieses Verfahrens.

Ebenfalls nützlich für die Objektlokalisation sind Verfahren zur Objektverfolgung bzw. -tracking in

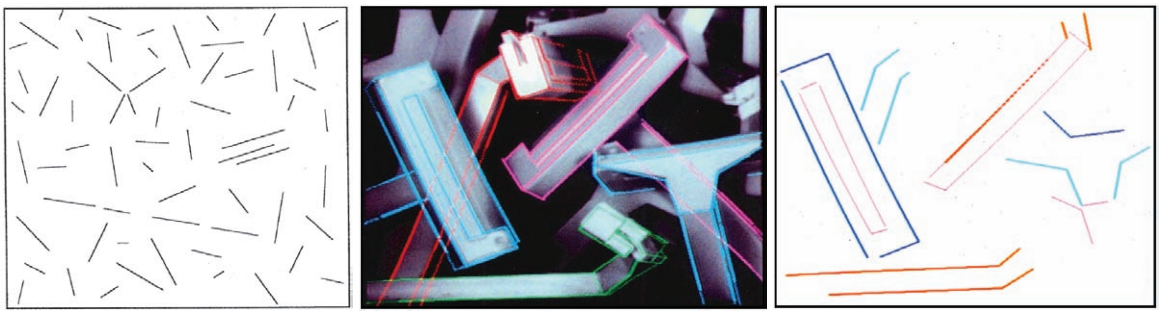

Abbildung 2.12: Links sind drei Arten perzeptueller Gruppen von Liniensegmenten abgebildet, die vom Menschen selbst in unstrukturiertem Hintergrund sofort wahrgenommen werden: Parallelität (Mitte rechts), Kollinearität (Mitte unten) und Endpunkt Nähe (oben links). In der Mitte befindet sich das Ergebnis mit überlagertem Kantenmodell der Objektlokalisation aus (Low87), welches die rechts abgebildeten perzeptuellen Gruppen zur Lageschätzung verwendet hat. Alle Bilder sind aus (Low87) entnommen.

Intensitätsbildern. Im Gegensatz zur Objektlokalisation erwarten sie eine Lage des Objekts aus einem vorherigen Bild, die dann an die neuen Intensitätsinformationen angepasst wird. Algorithmen zur Objektverfolgung können daher als Feinlageschätzung für andere Algorithmen verwendet werden.

Eines der ersten echtzeitfähigen Verfahren ist der *RAPiD-Tracker*[9] (HS90). Er verwendet ein Kantenmodell des zu verfolgenden Objekts. Mittels einer Lage aus dem vorangegangenen Frame werden die Kanten des Objekts in das aktuelle Bild projiziert. Dann wird an beliebigen, aber gleichmäßig verteilten Kontrollpunkten $\mathbf{u}_i$ auf den Modellkanten entlang der jeweiligen normierten Kanten-Normale $\mathbf{n}_i$ nach Bildkanten gesucht. Dies führt zu Tupeln $(\mathbf{u}_i, \mathbf{n}_i, l_i)$, wobei l_i dem Abstand zur nächsten Bildkante entspricht. Mindestens 6 dieser Tupel werden dann herangezogen, um die vorangegangene Lage mittels einer Newton-Minimierung anzupassen. Dies funktioniert allerdings nur, wenn die vorangegangene bzw. initiale Lage schon so genau ist, dass die projizierten Modellkanten nahe den korrespondierenden Bildkanten zu liegen kommen.

(MBCM99) verfährt ähnlich wie RAPiD um Kontrollpunkten $\mathbf{u}_i$ korrespondierende Bildpunkte $\mathbf{u}_i + l_i\mathbf{n}_i$ zuzuordnen, allerdings werden die etablierten 2D-2D Korrespondenzen genutzt, um eine affine Transformation des projizierten Modells zwischen aktuellem und vorangegangenem Frame zu schätzen. Damit können Fehlkorrespondenzen eliminiert werden. Da für die Kontrollpunkte die 3D Punkte im Modellkoordinatensystem bekannt sind, können die verbleibenden 2D-3D-Korrespondenzen mittels POSIT (DD95) (siehe Abschnitt 2.2.2.2) für eine robuste Lageschätzung verwendet werden. Diese wird anschließend mit einer ähnlichen Methode wie bei RAPiD an die aktuelle Kantenstruktur des Bildes angepasst. Ein Beispiel dieses Systems ist in Abb. 2.13 zu sehen.

(TK03) verwendet nicht nur Kanteninformationen, sondern extrahiert aus einem (gegebenem) texturierten CAD-Modell ebenfalls Farb- und Texturinformationen. Diese werden in einem integrierten Prozess mittels eines Kalman-Filters fusioniert und zum Tracking von Objekten herangezogen.

2.3 Bewertung

Dieser Abschnitt vergleicht nochmals alle Verfahren auf eine qualitative Art und Weise. Dabei kann natürlich nicht mehr auf jeden konkreten Algorithmus eingegangen werden, sondern es wird ver-

[9] **R**eal-time **A**ttitude and **P**os**i**tion **D**etermination

Abbildung 2.13: Experimente aus (MBCM99) zur Objektverfolgung eines Steckers.

sucht, die grundlegende Methodik verschiedener Verfahrensklassen zu bewerten. Insbesondere können viele Algorithmen bzw. Verfahrensklassen miteinander kombiniert werden. Daher ist in Tab. 2.1 angegeben, ob die Segmentierung, die Groblageschätzung oder die Feinlageschätzung von der Methodik abgedeckt ist (siehe dazu auch Abb. 2.1). Gerade die globalen ansichtsbasierten Methoden lassen sich mit einer ausführlichen Suche, dem *sliding Window* kombinieren und ermöglichen damit zumindest innerhalb des Bildes auch die Schätzung einer Translation. Besonders interessante Kombinationen bzw. Verfahren (AAD06; Low99; AAD07; USBE03) werden in Tab. 2.1 extra aufgeführt. Sie zeichnen sich dadurch aus, dass sie ein komplettes Verfahren zur Objektlokalisation beschreiben.

Eine Bewertung bzw. ein Vergleich, wie er in Tab. 2.1 gegeben ist, hängt natürlich stark von der Parametrierung der einzelnen Verfahren ab. Insbesondere Genauigkeit und Geschwindigkeit bzw. Trainingsaufwand verhalten sich je nach Einstellung oftmals komplementär zueinander, daher werden in diesen Fällen in Tab. 2.1 extreme Parametereinstellungen getrennt bewertet. Zum Vergleich mit dem in Kapitel 4 entwickelten Verfahren ist dieses unter *Neu* ebenfalls in der Tab. 2.1 aufgeführt.

Zusammenfassend lässt sich sagen, dass für viele praktische Anwendungen die Segmentierung eine herausragende Rolle besitzt. Hierzu bieten die globalen Methoden in den seltensten Fällen eine Lösung an. Wenn einfache Segmentierungsstrategien, wie Farbsegmentierung oder einlernbare Hintergründe nicht zu einer Lösung führen, sind die lokalen Methoden die einzige effiziente Möglichkeit für eine robuste Objekterkennung bzw. Lageschätzung. Weiterhin können sie meist besser mit lokalen Störungen bzw. Verdeckungen als auch komplexen Szenen mit vielerlei Objekten umgehen als die globalen Verfahren. Allerdings sind sie aufgrund ihrer lokalen Betrachtungsweise empfindlicher gegenüber verrauschten Daten und daher meist weniger genau. Daher ist eine erfolgreiche Strategie, lokale Verfahren zur robusten Segmentierung und Groblageschätzung zu verwenden und auf den segmentierten Daten eine globale Feinlageschätzung anzuwenden.

Ansichtsbasierte Verfahren haben den großen Vorteil, dass sie meist auf 2D Objektmodelle in Form einer Menge von Ansichten zurückgreifen, die relativ unkompliziert in einer autonomen Trainingsphase generiert werden können. Geometriebasierte Verfahren verwenden dagegen meist komplexere Kanten bzw. Volumenmodelle, die oftmals a priori durch ein vollständiges 3D-CAD-Modell gegeben sind. Diese Modelle *autonom* so zu trainieren, dass sie in der Praxis einsetzbar sind, ist eine wesentlich größere Herausforderung, da die Lokalisationsgenauigkeit dieser Verfahren normalerweise direkt mit der Modellgüte zusammenhängt. Sobald die Lage allerdings nicht mehr nur innerhalb des 2D Bildes geschätzt werden soll, benötigen auch ansichtsbasierte Verfahren geometrische Informationen um eine 3D-Lage zu ermitteln. Diese müssen allerdings nicht als vollständiges 3D Modell vorliegen, sondern können lokal in Form von Tiefenwerten oder Längeninformationen gegeben sein.

Die globalen ansichtsbasierten Korrelations-, Momente- bzw. Histogrammansätze lassen sich nur als Groblageerkennung einsetzen und besitzen auch dann nur eine mäßige Genauigkeit in Abhängigkeit der eintrainierten Ansichten. Auch lassen sich nur die momentenbasierten Verfahren heranziehen um die Rotation innerhalb von Bildern zu erkennen, so dass entweder eine sehr große Anzahl von Ansichten (AAD06) aller möglichen Rotationen eingelernt werden müssen oder der rotative Freiheitsgrad eingeschränkt werden muss. Die Genauigkeit lässt sich durch den Ansatz der parame-

Verfahren	Bereiche	Daten	Modell	Lage DoF	Robustheit	Genauigkeit	Geschw.
globale ansichtsbasierte Verfahren							
Korrelation	G	I	2D ++	1R (3D)	o L^- K^-	o	++
(AAD06)	S G	IK F D^-	2D --/++	3R 3T (3D)	o L^- K^-	+/o	++
Histogramme	G	I F	2D ++	1R (3D)	o+ L^- K^-	−o	+
Momente	G	I	2D ++	2-3R (3D)	o L^- K^-	o	++
Param. Mannigf.	G F	I	2D o	1-2R	o L^- K^-	+	+
Mustererkennung	G	I	2D o	1-2R	+	o/−	--/+
lokale ansichtsbasierte Verfahren							
Punkt basiert	G	IT	2D ++	1R (3D)	+ L^+ K^+	o	o+
Ähnliche Reg.	S G	IT	2D ++	2S 2T (2D)	+ L^+ K^+	o	o+
(Low99)	S G F	IT	2D ++	4A 2T (2D)	+ L^+ K^+	+	+
(AAD07)	S G F	IT D	2.5D o	3R 3T (3D)	+ L^+ K^+	++	o+
Affine Reg.	S G F	IT D^-	3D −	3R 3T (3D)	+ L^+ K^+	++	--
Neu (Kapitel 4)	S G F	IT D^-	2.5D o+	3R 3T (3D)	+ L^+ K^+	++	+
globale geometriebasierte Verfahren							
ICP	F	D	2.5D o	3R 3T (3D)	o L^+ K^-	++	o
PCA Ausgleich	G	D	2.5D +	3R 3T (3D)	−o L^- K^-	o	++
Histogramme 3D	G	D	2.5D o+	1-2R (3D)	o+ L^- K^-	o	o
EGI	G F	D^+	3D −	3R (3D)	o L^- K^-	+	−
lokale geometriebasierte Verfahren							
3D-3D-Kor.	G F	P	3D +	3R 3T (3D)	+	+	++
2D-3D-Kor.	G F	P	3D +	3R 3T (3D)	o	o	+
3D Regionen	S G	D^+	3D --	3R 3T (3D)	+ L^+ K^+	+	−
GHT	S G	IK	2D ++	1-4A 2T (2D)	+ L^+ K^+	−/+	+/−
(USBE03)	S G F	IK	2D +	2S 2T (2D)	+ L^+ K^+	++	++
GH (2D)	S G	P K	2D +	1-4A 2T (2D)	+ L^+	+	o/--
GH (3D)	S G	P K	3D −	3R 3T (3D)	+ L^+	+	−
Tracking	F	IK	3D −	3R 3T (3D)	o	++	++
Perz. Gruppen	S G F	IK	3D −	3R 3T (3D)	o L^+	+	o

Tabelle 2.1: Vergleich der Verfahren zur Objektlokalisation. Die spezielle Bedeutung der Spalten ist wie folgt: *Verfahren*) eingerückt sind direkt zitierte spezielle Methoden; *Bereiche*) Segmentierung S, Groblageschätzung G, Feinlageschätzung F; *Daten*) Intensität I, Farbe F, Textur T, Tiefeninformationen D (implizit durch Triangulation D^-, mit hoher Genauigkeit D^+); *Modell*) Menge von 2D Bildern, Menge von 2.5D Tiefenbildern, komplettes 3D Modell; danach geschätzter autonomer Trainingsaufwand; *Lage DoF*) geschätzte Lagefreiheitsgrade der Rotation R, Ähnlichkeitstransformation S, affine Verzerrung A und Translation T; in Klammern ausgegebene Lage in Bildkoordinaten (2D) oder Welt-Koordinaten (3D); die weiteren Spalten bewerten die *Robustheit* mit Herausforderung lokaler Störungen bzw. Verdeckungen L (große Probleme L^-, gute Eignung L^+) und komplexen Szenen bzw. Objektüberschneidungen K (große Probleme K^-, gute Eignung K^+), die *Genauigkeit* und die *Geschwindigkeit* des Verfahrens. Je nach Parameterwahl sind mit / verschiedene Extreme angegeben.

trierten Mannigfaltigkeiten (MN95; NNM96; BPPP98; CG00; HCK97) erhöhen, da dann zwischen Ansichten interpoliert werden kann. Allerdings wird dadurch das Training etwas aufwändiger und das Verfahren während der Detektion langsamer.

Die lokalen ansichtsbasierten Verfahren haben im Grunde alle dieselbe Struktur und unterscheiden sich vor allem in den verwendeten Merkmalen und den daraus gezogenen Informationen. Sie sind zwar etwas langsamer als ihre globalen Vertreter, adressieren aber auch die Segmentierung und können insbesondere mit eine Feinlageschätzung kombiniert als robustes 6D-Lokalisationsverfahren eingesetzt werden. Allerdings benötigen sie ausgeprägte Texturinformationen auf den Objekten. In diese Klasse fällt auch das neu entwickelte Verfahren aus Kap. 4, das aber im Gegensatz zu (AAD07) nicht auf ebene Bauteile beschränkt ist und im Gegensatz zu (Low99) eine vollständige 6 DoF Lage im 3D Raum liefert.

Eng verwand sind die lokalen geometriebasierten Verfahren auf Basis lokaler Beschreibungen von 3D-Strukturen. Sie benötigen hoch aufgelöste Tiefeninformationen und akkurate 3D-Modelle, können aber insbesondere mit dem ICP kombiniert ebenfalls als vollständiges 6D-Lokalisationsverfahren inklusive Segmentierung verwendet werden.

Die GHT und das GH sind weitere robuste geometriebasierte Standardverfahren der Objekterkennung auf Basis von Kanten oder Punktinformationen. Sie lassen sich relativ einfach autonom trainieren, liefern aber meist nur die Lage innerhalb eines Bildes. Auch sie haben großes Potential, insbesondere bei Kombination mit einer Feinlageschätzung. Die GHT sollte aber in der Praxis insbesondere bei komplexen Szenen einen deutlichen Geschwindigkeitsvorteil gegenüber der GH besitzen.

Für die Feinlageschätzung bieten sich die Tracking- und 2D-3D-Korrespondenzverfahren auf Intensitätsbildern oder der ICP und die 3D-3D- Korrespondenzverfahren für Tiefenbilder an. Während die Korrespondenzverfahren kein weiteres Modell benötigen, sind die Trackingalgorithmen oder der ICP auf ein geometrisches Modell angewiesen. Dafür können letztere Verfahren die Lageschätzung ohne weitere Korrespondenzinformationen durchführen.

MIR KOMMEN DIE WEGE, AUF DENEN DIE MENSCHEN ZUR ERKENNTNIS GELANGEN FAST EBENSO BEWUNDERUNGSWÜRDIG VOR WIE DIE NATUR DER DINGE SELBST.

Johannes Kepler

Kapitel 3

Interessante Bildregionen

Ein häufiges auftretendes Problem von kognitiven Bildverarbeitungsystemen ist das Finden von (physikalischen) Beziehungen zwischen verschiedenen Bildern ähnlichen Szeneninhalts. Diese Beziehungen sind ein wichtiger Schritt im kognitiven Prozess von low-level Bildintensitäten zu symbolischem Szenenverständnis, dienen der Informationsreduktion und abstrahieren von unerwünschten Einflüssen der zugrunde liegenden Abbildung. Sie lassen sich durch lokale Bildregionen modellieren, deren Beschreibungen in unterschiedlichen Bildern korrespondieren. Obwohl die Regionen und deren Beschreibungen im allgemeinen kontextabhängig sind, zeigt sich doch, dass für eine große Bandbreite von Problemstellungen, wie Stereo-Matching (Bau00; MCUP04), Indizierung in Bilddatenbanken (MS01; SZ02), Objekterkennung (Low01) oder Lokalisation (VL07) ähnliche Eigenschaften gefordert sind[1]. Neben einer Robustheit bzw. Invarianz gegenüber Rauschen, photometrischen Einfüssen und geometrischen Transformationen sollten die Beschreibungen bzw. die zugrunde liegenden Bildregionen auch möglichst diskriminativ sein und unabhängig von anderen Regionen gefunden werden.

In nahezu allen Systemen die sich mit obigen Problemstellungen beschäftigen, findet sich das in Abb. 3.1 dargestellte allgemeine Schema zum Finden lokaler Korrespondenzen wieder. In einer ersten Phase ermittelt ein Detektor relevante Regionen innerhalb des Bildes, für die obig genannten Eigenschaften gelten. Im nächsten Schritt werden die Bildinformationen der gefunden Regionen mit einem Deskriptor jeweils in einem Merkmalsvektor codiert. Dieser Merkmalsvektor beschreibt auf abstrakte Weise den Inhalt der Regionen und kann in einem weiteren Schritt von einem Matcher zum Auffinden von Korrespondenzen herangezogen werden. Sie korrespondieren dann, wenn ein auf dem Merkmalsraum definiertes Distanzmaß einen gewissen Wert unterschreitet. Sind erst einmal Korrespondenzen innerhalb von Bildern gefunden, lassen sich über die Lage der zugehörigen Regionen innerhalb der Abbildung diese Beziehung auf die physikalische Szene übertragen.

Dieses Kapitel gliedert sich in vier Bereiche. Zuerst werden theoretische Aspekte der invarianten Konstruktion von Merkmalen behandelt, sowie verschiedene Modelle für photometrische und geometrische Einflüsse von Kamerabbildungen vorgestellt. In den Unterabschnitten 3.2, 3.3 und 3.4 werden der Entwurf von verschiedenen Detektoren, Deskriptoren und Matchern beschrieben, sowie jeweils deren Performance verglichen. Die verschiedenen Verfahren lassen sich prinzipiell unabhängig voneinander verwenden, auf besonders geeignete Kombinationen wird aber an entsprechender Stelle hingewiesen.

[1] Wie in (KB01) dargestellt, gilt dies auch nach allgemeiner Überzeugung für die vordersten Verarbeitungsstufen des menschlichen visuellen Systems. Spielt zwar nach Neisser (Nei64) Kontextwissen in den höheren aufmerksamkeitsgesteuerten Stufen eine immer zentralere Bedeutung, werden in den Vor-Aufmerksamkeit-Stufen vor allem „Pop-Out“ Merkmale detektiert. Diese sind bzgl. ihrer Umgebung herausragend (salient) und bestehen aus räumlichen Diskontinuitäten, wie z. B. Farbblobs auf homogenem Hintergrund, lokalen Texturänderungen oder Ecken.

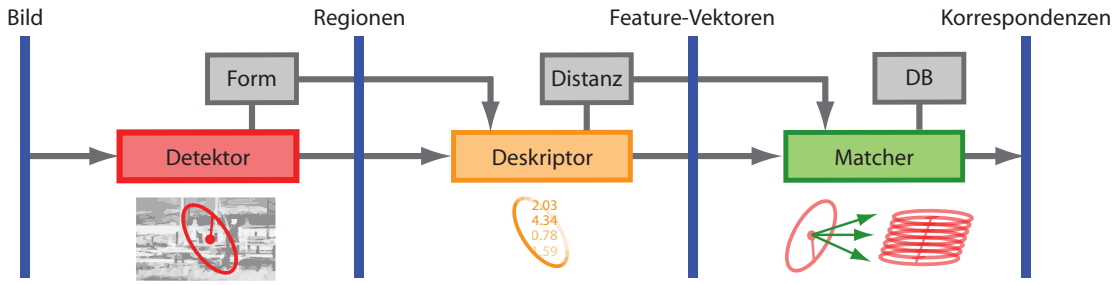

Abbildung 3.1: Schema zum Finden lokaler Bildkorrespondenzen. Schnittstellen sind blau dargestellt, die Verfahren dazwischen sind im Allgemeinen beliebig kombinierbar. Dennoch liefern einige Detektoren spezielle Regionformen, manche Deskriptoren ihr eigenes Distanzmaß und Matcher müssen auf eine für sie zugeschnittene Datenbank an Bildregionen zurückgreifen.

3.1 Invariante Merkmalskonstruktion

Ein Abbild einer physikalischen Szene $\bigcup_{\mathcal{S}} \subseteq \mathbb{R}^3$ ist ein Signal $\mathrm{i} : \mathcal{D} \subseteq \mathbb{R}^2 \rightarrow \mathcal{W} \subseteq \mathbb{R}$ welches einem Punkt $\mathbf{u} = (u,v)^{\mathrm{T}}$ der Bildebene $\mathcal{D}$ eine Intensität $\mathrm{i}(\mathbf{u})$ aus dem Wertebereich $\mathcal{W}$ zuordnet. $\mathbf{u} = \mathbf{h}(\mathbf{p})$ entspricht dabei dem Abbild des Szenenpunktes $\mathbf{p} = (x,y,z)^{\mathrm{T}} \in \bigcup_{\mathcal{S}}$ durch das geometrische Kameramodel $\mathbf{h}(\cdot)$ aus Abschnitt C.1. Es werden zwei verschiedene Arten von Einflüssen auf ein Intensitätsmuster $\mathcal{M}[\mathrm{i}(\cdot), \mathcal{R}] = \{(\mathbf{u}, \mathrm{i}(\mathbf{u})) \mid \mathbf{u} \in \mathcal{R} = \mathbf{h}(\mathcal{S})\}$, definiert durch die Abbildung $\mathrm{i}(\cdot)$ eines Bereichs der Szene $\mathcal{S}$ bzw. dessen Bildregion $\mathcal{R}$, betrachtet:

- Photometrische Einflüsse $\mathrm{i}'(\mathbf{u}) = \mathrm{photo}(\mathrm{i}(\mathbf{u}), \mathbf{u})$ bewirken eine Variation der Bildintensitäten durch andere Beleuchtungsverhältnisse oder Abbildungsparameter.
- Geometrische Einflüsse $\mathrm{i}'(\mathbf{geo}(\mathbf{u})) = \mathrm{i}(\mathbf{u})$ entstehen durch relative Lageänderung der abbildenden Kamera zu Teilen der Szene.

Für einen Merkmalsvektor $\mathbf{r}_{\mathcal{R}} = \mathbf{des}(\mathcal{M})$ eines Intensitätsmusters $\mathcal{M}$ gilt der Wunsch einer Invarianz $\mathbf{des}(\mathcal{M}[\mathrm{i}(\cdot), \cdot]) = \mathbf{des}[\mathcal{M}[\mathrm{i}'(\cdot), \cdot]]$ gegenüber diesen Einflüssen. $\mathbf{des}(\cdot)$ ist dabei ein Deskriptor, der $\mathcal{M}$ einen Merkmalsvektor zuordnet. Im Idealfall lässt sich $\mathbf{des}(\cdot)$ daher als Äquivalenzrelation auf der Menge aller möglichen Muster $\bigcup_{\{\mathcal{M}\}}$ auffassen. $\mathcal{M}_a \sim_{\mathbf{des}} \mathcal{M}_b$ gilt genau dann wenn $\mathbf{des}(\mathcal{M}_a) = \mathbf{des}(\mathcal{M}_b)$.

Definiert eine Funktion $\mathbf{g}[\mathbf{q}](\cdot)$, parametrisiert durch $\mathbf{q} = (q_1, \ldots)^{\mathrm{T}} \in \mathcal{Q}$ als dyadisches Element betrachtet eine algebraische (Transformations-) Gruppe $\mathcal{G} = \{\mathbf{g}[\mathbf{q}](\cdot) \mid \mathbf{q} \in \mathcal{Q}\}$, gibt es nach Burkhard (BS01) drei fundamentale Ansätze der Konstruktion invarianter Merkmale (bzgl. $\mathbf{g}$):

Integrale Methode Das Gruppenmittel oder auch Haar-Integral einer endlichen Transformationsgruppe $a[f](\cdot) = \frac{1}{|\mathcal{G}|} \int_{\mathcal{G}} f(\mathbf{g}(\cdot))\mathrm{d}\mathbf{g}$ normiert mit dem Volumen $\mid \mathcal{G} \mid = \int_{\mathcal{G}} \mathrm{d}\mathbf{g}$ über eine Kernfunktion $f : \cdot \rightarrow \mathbb{R}$ ist invariant bzgl. der Transformation $\mathbf{g}$ (Hur97). Ist die Kernfunktion bereits invariant, gilt $a[f](\cdot) = f(\cdot)$.

Differentielle Methode Ist auf dem Parameterraum $\mathcal{Q}$ eine Metrik $\mathrm{d} : \mathcal{Q}^2 \rightarrow \mathbb{R}$ definiert und hängt das Gruppenprodukt und inverse Element stetig von $\mathbf{q}$ ab, so wird die parametrische Gruppe $\mathcal{G}$ auch *Lie-Gruppe* genannt. Ist eine Lie-Gruppe weiterhin zusammenhängend, lässt sich jedes Element $\mathbf{g} \in \mathcal{G}$ als Produkt unendlich vieler infinitesimal kleiner Transformationen $\mathrm{d}\mathbf{g} = \mathbf{g}[\mathrm{d}\mathbf{q}]$ darstellen. Daher folgt, dass ein Merkmal m genau dann invariant gegenüber $\mathcal{G}$ ist, wenn es invariant gegenüber $\mathrm{d}\mathbf{g}$ ist. Formal ausgedrückt gilt dies für alle Lösungen $m(\cdot)$ der Differentialgleichungen $\frac{\mathrm{d}m(\mathbf{g}[\mathbf{q}])}{\mathrm{d}q_l}|_{\mathbf{q}=0} = 0$ mit $l = 1, \ldots, \dim(\mathcal{Q})$ und $\forall \mathbf{q} \in \mathcal{Q}$.

Normierung Bei der Normierung werden die Merkmale auf einer kanonischen Repräsentation $\mathcal{M}^*$ des Muster $\mathcal{M}$ berechnet. Dafür muss für alle Elemente $\mathcal{M}'$ des *Orbits* $O_{\mathcal{M}} = \{\mathbf{g}(\mathcal{M}) \mid \mathbf{g} \in \mathcal{G}\}$ ein Parameter $\mathbf{q}^*(\mathcal{M}')$ angegeben werden können, so dass $\mathcal{M}^* = \mathbf{g}[\mathbf{q}^*_{\mathcal{M}'}](\mathcal{M}')$ gilt. Weiterhin können zwei Muster $\mathcal{M}'$, $\mathcal{M}'' \in O_{\mathcal{M}}$, d.h. derselben Äquivalenzklasse, mittels $\mathbf{g}[\mathbf{q}^{**}] = \mathbf{g}[\mathbf{q}^*_{\mathcal{M}'}]^{-1}(\mathbf{g}[\mathbf{q}^*_{\mathcal{M}''}])$ bzw. dessen Inversen ineinander übergeführt werden.

Im folgenden werden nun Modelle für die photometrischen und geometrischen Einflüsse, photo$(\cdot)$ und **geo**$(\cdot)$ von (Grauwert-) Intensitätsmustern behandelt und auf konkrete invariante Konstruktionsmethoden hingewiesen.

3.1.1 Photometrisches Transformationsmodell

Eine Abbildung i$(\mathbf{u})$ einer Szene wird beeinflusst durch die Lage von Objekten, deren Reflexionseigenschaften, den Beleuchtungsverhältnissen innerhalb der Szene sowie dem Abbildungsvorgang. Da sich dies für den allgemeinen Fall nicht modellieren lässt, kann man ein abstraktes Modell $\mathrm{i}'(\mathbf{u}) = \mathrm{photo}(\mathrm{i}(\mathbf{p}), \mathbf{p})$ mit einem intensitätsabhängigen Teil (z. B. Modellierung globaler Beleuchtungsverhältnisse) und einem ortsabhängigen Teil (z. B. Modellierung von Abbildungseigenschaften wie Vignettierung) angeben. Da man aber lokale Regionen auswerten möchte, reicht es aus, dass das photometrische Modell nur lokal seine Gültigkeit besitzt. Es wird daher meist ein mathematisch leicht handhabbares, lineares Modell

$$\mathrm{photo}(\mathrm{i}(\mathbf{u}), \mathbf{u}) = \mathrm{photo}(\mathrm{i}(\mathbf{u})) = a\mathrm{i}(\mathbf{u}) + b \qquad a > 0$$

verwendet, welches nur von der Intensität abhängig ist (MTS$^+$05; MS05). Der multiplikative Anteil a verändert dabei den Kontrast, der additive Anteil b den mittleren Intensitätswert von i. Änderungen der Beleuchtungsstärke oder Einflüsse von Abbildungsparameter wie Belichtungszeit oder Gain lassen sich damit gut beschreiben, für Glanzpunkte oder nichtlineare Intensitätsschwankungen (vgl. z. B. (CJ03)) wie sie durch Änderung der Beleuchtungsrichtung entstehen ist die Modellierung natürlich ungenügend.

Die lineare Funktion photo$(\cdot)$ definiert eine unendliche Transformationsgruppe mit dem Parametervektor $\mathbf{q} = (a, b)^{\mathrm{T}} \in Q = \mathbb{R}^+ \times \mathbb{R}$. Eine einfache Möglichkeit der invarianten Merkmalskonstruktion ist die Normierung des Kontrastes und des mittleren Wertes eines Intensitätsmusters $\mathcal{M}[\mathrm{i}(\cdot), \mathcal{R}]$. Eine mögliche kanonische Form $\mathcal{M}^* = \mathcal{M}[\mathrm{i}^*(\cdot), \mathcal{R}]$ mit $\mathrm{i}^* = \frac{1}{a^*}(\mathrm{i} - b^*)$ ist durch eine lokale Schätzung des mittleren Grauwertes $b^* = \frac{1}{|\mathcal{R}|}\int_{\mathcal{R}} \mathrm{i}(\mathbf{u})\mathrm{d}\mathbf{u}$ und des lokalen Kontrastes $a^* = \sqrt{\frac{1}{|\mathcal{R}|}\int_{\mathcal{R}} (\mathrm{i}(\mathbf{u}) - b^*)^2 \mathrm{d}\mathbf{u}}$ über $\mathcal{R}$ definiert.

Ist es nicht möglich oder unerwünscht eine Region $\mathcal{R}$ anzugeben, gibt es weitere Möglichkeiten der invarianten Merkmalskonstruktion. So sind alle Merkmale basierend auf Intensitätsdifferenzen (z.B. $\mathrm{i}(\mathbf{u}_1) - \mathrm{i}(\mathbf{u}_2)$, $\mathrm{i}_{[u]}$ oder $\mathrm{i}_{[v]}$) unabhängig von der Intensitätsverschiebung b, Verhältnisse von Intensitäten unabhängig von Kontrastveränderungen a und Verhältnisse von Intensitätsdifferenzen (z.B. $\frac{\mathrm{i}_{[u]}}{\mathrm{i}_{[v]}}$) unabhängig gegen a und b.

Das lineare photometrische Modell approximiert deterministische Einflüsse der Abbildung und Szenenverhältnisse, in realen Bildern kommt es aber auch zu indeterministischen Störungen, wie das Rauschen des Bildsensors oder das Quantisierungsrauschen. In der Praxis wird eine durchaus realistische additive Überlagerung von Signal und Rauschen angenommen:

$$\tilde{\mathrm{i}}(\mathbf{p}) = \mathrm{i}(\mathbf{u}) + \tilde{r}(\mathbf{u})$$

Die Zufallsvariable $\tilde{r}(\mathbf{u})$ wird dabei als weißes ($\tilde{r}(\mathbf{u}_1)$ ist unabhängig von $\tilde{r}(\mathbf{u}_2)$), erwartungswertfreies gaußsches Rauschen ($\tilde{r}(\mathbf{u})$ verhält sich proportional zu einer Normalverteilung $\tilde{n}(0, \sigma)$) modelliert. Durch den Indeterminismus ist eine invariante Merkmalskonstruktion gegenüber Rauschen

nicht möglich, in der Praxis versucht man zumindest robuste Merkmale bzgl. $\tilde{r}$ zu konstruieren. So führt z. B. eine Mittelung $\frac{1}{|\mathcal{R}|}\int_{\mathcal{R}}\tilde{\mathrm{i}}(\mathbf{u})\mathrm{du} \approx \frac{1}{|\mathcal{R}|}\int_{\mathcal{R}}\mathrm{i}(\mathbf{u})\mathrm{du}+\mathbf{E}[\tilde{r}]$ über eine geeignet große Region $\mathcal{R}$ zu einem quasi-invarianten Merkmal, da der Erwartungswert $\mathbf{E}[\tilde{r}]$ der Rauschvariable definitionsgemäß verschwindet.

3.1.2 Geometrisches Transformationsmodell

Bewegt sich die Kamera oder Teile der Szene zwischen zwei Aufnahmen $\mathrm{i_A}$ und $\mathrm{i_B}$ relativ zueinander, kommt es zu geometrischen Veränderungen die mittels dem Modell $\mathrm{i_B}(\mathbf{geo}(\mathbf{p})) = \mathrm{i_A}(\mathbf{p})$ beschrieben werden sollen. Da dieses Modell im weiteren Verlauf der Arbeit eine essentielle Rolle spielen wird, soll es detailliert betrachtet und auch hergeleitet werden. Ausgangspunkt ist eine lokal begrenzte, zusammenhängende Teilmenge $\mathcal{S}[\mathbf{p}] = \mathbf{p}+\mathcal{S}$ der Szene. $\mathbf{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ beschreibt dabei einen geeignet gewählten festen Bezugspunkt und $\mathcal{S} = \{\mathbf{q}\}$ eine Menge lokaler Punkte $\mathbf{q} = (x,y,z)^{\mathrm{T}}$ um $\mathbf{p}$. Das exakte Abbild in Bildkoordinaten $\mathcal{R}_e$ von $\mathcal{S}[\mathbf{p}]$ in Kamera-Koordinaten berechnet sich nach Abschnitt C.1 zu

$$\mathcal{R}_e = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \mid \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{h}(\mathbf{p}+\mathbf{q}) = \mathbf{K}\mathbf{d}\left(\frac{1}{p_z+z}\begin{pmatrix} p_x+x \\ p_y+y \end{pmatrix}\right) + \begin{pmatrix} c_u \\ c_v \end{pmatrix}, \mathbf{q} \in \mathcal{S}[\mathbf{p}] \right\} \tag{3.1}$$

Für eine weiterführende Modellierung werden nun drei Annahmen getroffen:

1. Lokal lässt sich die perspektivische Projektion durch eine skalierte Parallel-Projektion (SOP[2]) beschreiben, d.h. $\frac{1}{p_z+z} \approx \frac{1}{p_z}$. Der Fehler dieser Approximation ist gering falls $p_z \gg z$ bzw. falls die Ausdehnung von $\mathcal{S}$ in der Tiefe klein bzgl. dem Abstand zur Kamera ist.

2. Die Verzerrung der Kamera verhält sich in einem lokalen Ausschnitt um einen unverzerrten Punkt $\bar{\mathbf{u}}_\mathbf{p}$ der Bildebene linear. Mittels einer zweidimensionalen Taylorreihe um $\bar{\mathbf{u}}_\mathbf{p}$ wird $\mathbf{d}(\bar{\mathbf{u}}_\mathbf{p}+\bar{\mathbf{u}}) \approx \mathbf{d}(\bar{\mathbf{u}}_\mathbf{p}) + \mathbf{D}_\mathbf{p}\bar{\mathbf{u}} + \ldots$ entwickelt und durch Abbruch nach dem Polynom erster Ordnung linearisiert. $\mathbf{D}_\mathbf{p}$ ist dabei die Jacobi-Matrix, die die Ableitungen von $d_u(\cdot)$ und $d_v(\cdot)$ an der Stelle $\bar{\mathbf{u}}_\mathbf{p}$ nach jeweils u und v enthält (vgl. Gleichung C.9). Diese Annahme trifft nur auf Systeme mit mäßig verzerrenden Objektiven sowie auf Bildregionen mit kleinen Ausdehnungen zu. Alternativ kann mit unverzerrten Bildern gearbeitet werden.

3. Die lokal begrenzte, zusammenhänge Menge $\mathcal{S}[\mathbf{p}]$ lässt sich als Ebene durch $\mathbf{p}$ darstellen, d.h. $0 \approx m_x x + m_y y + z$. Der Fehler dieser Annahme verschwindet, falls rotatorische Bewegungen nur innerhalb der Bildebene stattfinden und nimmt bei steigender Dominanz anderer Rotationen zu.

Mittels den Annahmen eins und zwei lässt sich Formel 3.1 wie folgt umformen:

$$\mathcal{R}_e \approx \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \mid \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{h}\left(\frac{1}{p_z}\begin{pmatrix} p_x \\ p_y \end{pmatrix}\right) + \mathbf{K}\mathbf{D}_\mathbf{p}\frac{1}{p_z}\begin{pmatrix} x \\ y \end{pmatrix}, \mathbf{q} \in \mathcal{S}[\mathbf{p}] \right\} = \mathcal{R}[\mathbf{u}_\mathbf{p}] \tag{3.2}$$

$$\mathcal{R}[\mathbf{u}_\mathbf{p}] = \mathbf{u}_\mathbf{p} + \frac{1}{p_z}\mathbf{D}_\mathbf{p}\mathcal{R} \tag{3.3}$$

Interpretiert man diese Formeln, so beschreiben $\mathcal{S}$ und $\mathcal{R} = \{\mathbf{u} \mid \mathbf{u} = (x,y)^{\mathrm{T}}, \mathbf{q} \in \mathcal{S}\}$ die Struktur des Szenenausschnitts bzw. das projizierte Grund-Muster der korrespondieren Region in der Bildebene. Mittels der Bezugspunkte $\mathbf{p}$ bzw. $\mathbf{u}_\mathbf{p}$ werden diese Grundmengen in der Szene bzw. der Bildebene verschoben und bilden dann den beobachteten Szenenabschnitt $\mathcal{S}[\mathbf{p}]$ bzw. die beobachtete Region $\mathcal{R}[\mathbf{u}_\mathbf{p}]$. Man beachte, dass sich aus $\mathcal{S}$ zwar $\mathcal{R}$ ohne Kenntnis der Bezugspunkte ermitteln lässt, das

[2] engl.: Scaled Orthographic Projection

tatsächlich beobachtete Muster $\mathcal{R}[\mathbf{u_p}]$ aber noch abhängig von $\mathbf{p}$ mit $\frac{1}{p_z}$ skaliert bzw. abhängig von $\mathbf{u_p}$ mit $\mathbf{D_p}$ sowie der Kameramatrix $\mathbf{K}$ affin verzerrt wird.

Um $\mathbf{geo}(\cdot)$ zu modellieren, wird im Folgenden gezeigt wie sich die beobachtete Region ${}^{\mathrm{A}}\mathcal{R}[\mathbf{u_p}]$ in ${}^{\mathrm{B}}\mathcal{R}[\mathbf{u_p}]$ verändert, wenn sich zwischen den Zeitpunkten A und B der zugrunde liegende (starre) Szenenabschnitt relativ zur Kamera mittels der Rotationsmatrix ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}} = (r_{ij})_{i,j=1,2,3}$ und dem Translationsvektor ${}^{\mathrm{B}}\mathbf{t} = (t_x, t_y, t_z)^{\mathrm{T}}$ bewegt:

$$ {}^{\mathrm{B}}\mathcal{S}[\mathbf{p}] = {}^{\mathrm{B}}\mathbf{t} + {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{S}[\mathbf{p}] = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{p} + {}^{\mathrm{B}}\mathbf{t} + {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{S} = {}^{\mathrm{B}}\mathbf{p} + {}^{\mathrm{B}}\mathcal{S} \tag{3.4} $$

Es zeigt sich, dass sich der Bezugspunkt in der Szene ${}^{\mathrm{B}}\mathbf{p} = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{p} + {}^{\mathrm{B}}\mathbf{t}$ und in der Abbildung ${}^{\mathrm{B}}\mathbf{u_p} = \mathbf{h}({}^{\mathrm{B}}\mathbf{p})$ direkt aus dem ursprünglichen Bezugspunkt ${}^{\mathrm{A}}\mathbf{p}$ ermitteln lassen. Davon unabhängig ändert sich das Grund-Muster

$$ {}^{\mathrm{B}}\mathcal{R} = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \mid \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} r_{11}x + r_{12}y + r_{13}z \\ r_{21}x + r_{22}y + r_{23}z \end{pmatrix}, \mathbf{q} \in {}^{\mathrm{A}}\mathcal{S} \right\} \tag{3.5} $$

Nutzt man nun die dritte Annahme $z \approx -m_x x - m_y y$ aus, folgt daraus

$$ {}^{\mathrm{B}}\mathcal{R} \approx \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \mid \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} r_{11} - m_x r_{13} & r_{12} - m_y r_{13} \\ r_{21} - m_x r_{23} & r_{22} - m_y r_{23} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \mathbf{q} \in {}^{\mathrm{A}}\mathcal{S} \right\} = {}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{R} \tag{3.6} $$

Man erkennt, dass für hinreichend ebene Szenenausschnitte bei Vernachlässigung der Perspektive die Grund-Muster bei beliebigen Bewegungen[3] durch eine affine Transformation

$$ {}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}} = \begin{pmatrix} r_{11} - m_x r_{13} & r_{12} - m_y r_{13} \\ r_{21} - m_x r_{23} & r_{22} - m_y r_{23} \end{pmatrix} \tag{3.7} $$

auseinander hervorgehen. Diese Transformation hängt nur von der zugrunde liegenden Rotation des Szenenausschnitts, sowie den Ebenenparameter ab. Für die beobachteten Regionen gilt nach Ausnutzung von Formel 3.3 und 3.6 folgende Umformung:

$$ {}^{\mathrm{B}}\mathcal{R}[\mathbf{u_p}] \approx {}^{\mathrm{B}}\mathbf{u_p} + \frac{{}^{\mathrm{A}}p_z}{{}^{\mathrm{B}}p_z}\,{}^{\mathrm{B}}\mathbf{K}\,{}^{\mathrm{B}}\mathbf{D_p}\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{D_p}^{-1}\,{}^{\mathrm{A}}\mathbf{K}^{-1}({}^{\mathrm{A}}\mathcal{R}[\mathbf{u_p}] - {}^{\mathrm{A}}\mathbf{u_p}) = {}^{\mathrm{B}}\mathbf{t}_u + {}^{\mathrm{B}}\mathbf{A'}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{R}[\mathbf{u_p}] \tag{3.8} $$

Vergleicht man Formel 3.4 für die Umformung in der Szene mit 3.8, der Umformung in der Bildebene, fällt sofort die ähnliche Struktur auf. Die Verschiebung ${}^{\mathrm{B}}\mathbf{t}$ innerhalb der Szene korrespondiert mit der Verschiebung ${}^{\mathrm{B}}\mathbf{t}_u = {}^{\mathrm{B}}\mathbf{u_p} + {}^{\mathrm{B}}\mathbf{A'}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{u_p}$ in der Bildebene. Eine Rotation ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}$ korrespondiert in der Bildebene mit einer affinen Transformation ${}^{\mathrm{B}}\mathbf{A'}_{\mathrm{A}} = s\,{}^{\mathrm{B}}\mathbf{K}\,{}^{\mathrm{B}}\mathbf{D_p}\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{D_p}^{-1}\,{}^{\mathrm{A}}\mathbf{K}^{-1}$ unter Berücksichtigung der Linsenverzerrung bzw. $s\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}$ im unverzerrten Fall. Die entstehende Skalierung $s = {}^{\mathrm{A}}p_z / {}^{\mathrm{B}}p_z$ ist dabei das Verhältnis der Abstände der Bezugspunkte zur Kamera.

Für $\mathbf{geo}(\cdot)$ bedeutet dies, dass (unter Berücksichtigung der Nährungen) im allgemeinen Fall ein *affines* Modell

$$ \mathbf{geo}(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{t}_u \qquad \mathbf{A} \in \mathcal{T}^{+}_{A,n} \tag{3.9} $$

ausreichend ist. Es definiert eine affine Gruppe, die durch die sechs Parameter $(a_{11}, a_{12}, a_{21}, a_{22}, u_t, v_t) \in \mathbb{R}^6$ vollständig definiert ist. Beschränkt man sich auf unverzerrte Bilder und Rotationen innerhalb der Bildebene ($r_{13} = r_{23} = 0$) lässt sich das Modell zu einer Ähnlichkeitstransformation vereinfachen.

$$ \mathbf{geo}(\mathbf{u}) = \mathbf{S}\mathbf{u} + \mathbf{t}_u \qquad \mathbf{S} = s\mathbf{R}_{\varphi} \in \mathcal{T}^{+}_{S,n} \tag{3.10} $$

Der benötigte Parametervektor $(s, \varphi, u_t, v_t) \in \mathbb{R}^{+} \times \mathbb{R}^3$ für die ähnliche Gruppe reduziert sich daher auf 4 Dimensionen. Wie der folgende Abschnitt zeigt, hat das *ähnliche* Modell für den Entwurf von Detektoren eine enorme historische aber auch praktische Bedeutung.

[3] Dies gilt natürlich nur für Bewegungen in denen der gesamte Szenenabschnitt sichtbar bleibt, d.h es weder zu Verdeckungen durch andere Szenenobjekte kommt, noch zu einer Rotation > 90° aus der Kameraebene hinaus.

3.2 Detektoren

Ein Regionen-Detektor hat die Aufgabe aus einem Bild diejenigen Regionen zu extrahieren, die zum Auffinden von Korrespondenzen geeignet sind. Dafür ist es notwendig, dass der Regionen-Deskriptor den Merkmals-Vektor invariant gegenüber dem geometrischen Modell $\mathbf{geo}(\cdot)$ konstruieren kann. Diese Invarianz wird meistens durch eine Normierung der Regionen auf ein isotropes Muster, bzw. eine kanonische Form N erreicht, das unter allen möglichen Transformationen $\mathbf{geo}(\cdot)$ denselben Bildinhalt codiert. Notwendigerweise muss der Detektor daher Regionen konstruieren, die sich *kovariant* bzgl. $\mathbf{geo}(\cdot)$ verhalten, d.h. die dieselbe Transformation unterlaufen wie das zugrunde liegende Grauwertmuster. Weiterhin müssen für die Regionen Konstruktionsmethoden verwendet werden, die invariant gegenüber den photometrischen Einflüssen $\mathrm{photo}(\cdot)$ sowie gegenüber Rauschen sind. Um möglichst eindeutige Korrespondenzen zu finden, sind ebenfalls nur Regionen geeignet, die ein genügend „interessantes“ Muster aufweisen und daher dem Deskriptor erlauben einen möglichst diskriminativen Merkmalsvektor zu berechnen.

Zwei Grundformen von Regionen haben sich in der Wissenschaft etabliert: die Gruppe der gerichteten Ellipsen $\mathcal{R}[\mathbf{u}_\perp, \mathbf{A}_\mathrm{N}] = \{\mathbf{u} \mid (\mathbf{u}-\mathbf{u}_\perp)^\mathrm{T} {\mathbf{A}_\mathrm{N}}^{-\mathrm{T}} {\mathbf{A}_\mathrm{N}}^{-1} (\mathbf{u}-\mathbf{u}_\perp) \leq 1\}$, $\mathbf{A}_\mathrm{N} \in \mathcal{T}^+_{A,n}$, abgeschlossen unter einer affinen Transformation, sowie die Gruppe der gerichteten Kreise $\mathcal{R}[\mathbf{u}_\perp, \mathbf{S}_\mathrm{N}] = \{\mathbf{u} \mid (\mathbf{u}-\mathbf{u}_\perp)^\mathrm{T} {\mathbf{S}_\mathrm{N}}^{-\mathrm{T}} {\mathbf{S}_\mathrm{N}}^{-1} (\mathbf{u}-\mathbf{u}_\perp) \leq 1\}$, $\mathbf{S}_\mathrm{N} \in \mathcal{T}^+_{S,n} \subset \mathcal{T}^+_{A,n}$, abgeschlossen unter einer Ähnlichkeitstransformation. Diese Formen korrespondieren mit dem angenommenen geometrischen Modell: affin oder ähnlich. Die Richtung bedeutet, das zur Form Ellipse bzw. Kreis noch eine charakteristische Richtung φ des zugrunde liegenden Musters angegeben wird. Diese wird im affinen Fall manchmal vernachlässigt, da die Ellipse über die Streumatrix $\mathbf{A} = \mathbf{A}_\mathrm{N}{\mathbf{A}_\mathrm{N}}^\mathrm{T} = \mathbf{A}_\mathrm{N}\mathbf{R}_\varphi\mathbf{R}_\varphi^\mathrm{T}{\mathbf{A}_\mathrm{N}}^\mathrm{T}$ unabhängig der Richtung $\mathbf{R}_\varphi$ ermittelt wird. Abkürzend werden die affin-kovarianten Detektoren auch als affine Detektoren bzw. A-Detektoren bezeichnet bzw. die ähnlich-kovarianten Detektoren als ähnliche Detektoren bzw. S-Detektoren. Nicht alle Detektoren konstruieren diese Grundformen direkt, bei manchen Algorithmen ist es notwendig die Regionen in einem zweiten Schritt zu vereinheitlichen. Dabei kann zwar Information verloren gehen, um eine allgemeine Schnittstelle für die Deskriptoren zu gewährleisten ist dies aber nicht zu vermeiden. Bei der Ermittlung der Regionen können aus Effizienzgründen nicht für alle Bildpunkte die 6 bzw. 4 Parameter der Grundformen gleichzeitig variiert werden. Es werden daher normalerweise erst Ankerpunkte bzw. Kandidaten im Bild bestimmt, um die dann eine entsprechende Region konstruiert wird.

Eine mögliche Taxonomie von Detektoren ist die in dieser Arbeit verwendete Unterscheidung durch die Konstruktionart der Ankerpunkte. Diese basieren auf einer Auswertung des Grauwertstrukturtensors, der Hesse-Matrix, von Intensitätsextrema oder der lokalen Entropie. Erst in der darunter liegenden Unterteilung wird zwischen den zwei geometrischen Modellen affine oder ähnlich unterschieden, da die Ellipsenkonstruktion meist aus einer Erweiterung der Kreiskonstruktion hervorgeht. Tab. 3.1 gibt einen Überblick über alle in diesen Abschnitt behandelten Verfahren, Abbildung 3.3 zeigt einige Beispiele.

Der weitere Abschnitt gliedert sich in allgemeine Methoden zur kovarianten Regionenkonstruktion, behandelt dann alle relevanten Detektoren, unterteilt durch obig beschriebene Taxonomie und schließt mit einer Bewertung der Detektoren ab.

3.2.1 Allgemeine Methoden zur kovarianten Regionenkonstruktion

Wie in Abschnitt 3.1 bereits erwähnt bildet der Deskriptor $\mathbf{des}(\cdot)$ im Idealfall eine Äquivalenzrelation auf $\bigcup_{\{\mathcal{M}\}}$, wobei eine Äquivalenzklasse $[\mathcal{M}]_\mathbf{des}$ unter anderem alle diejenigen Muster enthält, die durch geometrische Transformationen $\mathbf{geo}(\cdot)$ untereinander hervorgehen. Damit der Deskriptor dies leisten kann, müssen zwei Muster in verschiedenen Bildern $\mathcal{M}_a[i_a(\cdot), \mathcal{R}_a]$, $\mathcal{M}_b[i_b(\cdot), \mathcal{R}_b] \in [\mathcal{M}]_\mathbf{des}$ derselben Äquivalenzklasse auch denselben physikalischen Bildinhalt codieren. Geht Bild $i_b(\mathbf{geo}(\mathbf{u})) =$

Verfahren		**geo**	Form	Literatur
HaS	Harris-Similar	ähnlich	Kreis	(MS04)
HeS	Hessian-Similar	ähnlich	Kreis	(Mik02)
SIFT	Scale-Invariant-Feature-Transformation	ähnlich	Kreis	(Low04)
SURF	Speeded-Up-Robust-Features	ähnlich	Kreis	(BTVG06)
SaS	Salient-Similar-Regions	ähnlich	Kreis	(KB01)
HaA	Harris-Affine	affin	Ellipse	(MS04)
EBR	Edge-Based-Regions	affin	Ellipse	(TVG04)
HeA	Hessian-Affine	affin	Ellipse	(MTS$^+$05)
MSER	Maximal-Stable-Extremal-Regions	affin	Ellipse	(MCUP04)
IBR	Intensity-Based-Regions	affin	Ellipse	(TVG04)
SaA	Salient-(Affine)-Regions	affin	Ellipse	(KZB04)

Tabelle 3.1: Überblick über alle in diesem Abschnitt behandelten Detektoren. In dieser Arbeit werden aus Platzgründen durchgängig die in dieser Tabelle vorgestellten Abkürzungen der Verfahren verwendet. Diese decken sich so weit möglich mit den in der Wissenschaft üblichen Bezeichnungen.

$i_a(\mathbf{u})$ durch eine geometrische Transformation aus Bild $i_a(\cdot)$ hervor, muss daher

$$\{i_a(\mathbf{u}) \mid \mathbf{u} \in \mathcal{R}_a\} \stackrel{!}{=} \{i_b(\mathbf{u}) \mid \mathbf{u} \in \mathcal{R}_b\} = \{i_a(\mathbf{u}) \mid \mathbf{u} \in \mathbf{geo}^{-1}(\mathcal{R}_b)\}$$

gelten. Dies ist offensichtlich nur für $\mathcal{R}_b = \mathbf{geo}(\mathcal{R}_a)$ der Fall, d.h. wenn sich die Regionen *kovariant* bzgl. $\mathbf{geo}(\cdot)$ verhalten.

Für eine kovariante Konstruktion muss der Detektor charakteristische Strukturen des Musters auswerten. Dies kann für alle Parameter von $\mathbf{geo}(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{t}_u$ direkt geschehen, es ist aber sinnvoller die Verformungsmatrix $\mathbf{A}$ in ihre besser interpretierbaren Komponenten zu zerlegen. Mittels einer *Singular-Value-Decomposition*, kurz *SVD* (Bro99) lässt sich jede beliebige affine 2×2 Matrix $\mathbf{A}$ mit $\det \mathbf{A} > 0$ in folgende Einzelbestandteile zerlegen (HZ04):

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}}(\mathbf{U}\mathbf{V}^{\mathrm{T}}) = \mathbf{R}_\theta \mathbf{S}\mathbf{R}_{-\theta}\mathbf{R}_\varphi = \mathbf{S}_a\mathbf{R}_\varphi \tag{3.11}$$

Das geometrische Modell lässt sich daher durch eine Verschiebung $\mathbf{t}_u$, eine Rotation $\mathbf{R}_\varphi = \mathbf{U}\mathbf{V}^{\mathrm{T}}$ sowie eine anisotrope Skalierung $\mathbf{S}_a = \mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}} = \mathbf{R}_\theta \mathbf{S}\mathbf{R}_{-\theta}$ beschreiben. Die Skalierungsmatrix $\mathbf{S} = \mathrm{diag}(s_1, s_2)$ enthält dabei auf den Diagonalelementen die Eigenwerte von $\mathbf{A}$, die einer Skalierung entlang der Koordinatenachsen entsprechen. Mittels des letzten Parameters θ lassen sich eben diese Koordinatenachsen drehen, um eine anisotrope Skalierung in beliebiger Richtung zu ermöglichen. Die affine Transformationsgruppe lässt sich daher alternativ mit dem leicht interpretierbaren 6D-Parametervektor $(\varphi, \theta, s_1, s_2, u_t, v_t)$ beschreiben. Modelliert man nur eine isotrope Skalierung $s1 = s2 = s$, wird θ überflüssig und $\mathbf{S}_a$ reduziert sich auf $s\mathbf{I}$. Der Parametervektor der ähnlichen Gruppe wird daher (wie für Formel 3.10 schon angegeben) zu (φ, s, u_t, v_t).

Da für jede Parametergruppe Translation (u_t, v_t), Rotation (φ), isotrope Skalierung (s) sowie anisotrope Skalierung (θ, s_1, s_2) andere charakteristische Strukturen bzw. Konstruktionsmethoden verwendet werden, sollen diese unabhängig voneinander in den folgenden Unterabschnitten vorgestellt werden. Diese Unterabschnitte sind von allgemeinerer Natur und stellen die Grundlage für die später beschriebenen Detektoren dar.

3.2.1.1 Translation

Eine Verschiebung innerhalb der Bildebene lässt sich bei lokaler Betrachtung nicht für alle Bildbereiche ermitteln. Man benötigt charakteristische Strukturen, die genügend Informationsgehalt besitzen,

um eine eindeutige Verschiebung in beliebiger Richtung ermitteln zu können. Betrachtet man z.B. eine homogene Fläche lässt sich überhaupt nicht auf eine etwaige Translation schließen, bei einer optimalen Kante nur auf mögliche Translationen entlang des Gradienten. Diese Problematik ist von allgemeiner Natur und wird u.a. als Blendenproblem beim optischen Fluss (Jäh05) bezeichnet. Daher finden sich die im weitern Verlauf vorgestellten Ansätze im Kern in vielen Problemstellungen der Bildverarbeitung.

Für die Detektion von kovarianten Regionen bezeichnet man Pixel mit geeigneter charakteristischer Umgebung auch als Ankerpunkte. Fast alle Verfahren extrahieren in einem ersten Schritt diese Punkte im Bild und versuchen dann nur noch für diese Auswahl robuste kovariante Regionen zu konstruieren. Dieser zweistufige Selektionsprozess dient einerseits einer deutlichen Informationsreduktion im ersten Schritt, mit dementsprechendem Performancegewinn, ist andererseits durch die unabhängige Parameterbestimmung von Translation und Form suboptimal. Viele Detektoren verfeinern daher bei der eigentlichen Regionenkonstruktion nochmals iterativ den Ankerpunkt.

Alle in dieser Arbeit behandelten Verfahren zur Ankerpunktbestimmung lassen sich auf folgendes einfaches Schema reduzieren:

1. Extrahiere für jeden Bildpunkt $\mathbf{u} \in \mathcal{D}$ ein Merkmal $m(\mathbf{u}) \in \mathbb{R}$, welches die Eignung der zugrunde liegenden Bildstruktur als Ankerpunkt beschreibt.

2. Bestimme die Menge aller Bildpunkte $\mathcal{D}_\tau = \{\mathbf{u} \mid m(\mathbf{u}) > \tau, \mathbf{u} \in \mathcal{D}\}$ deren Merkmal größer eines Schwellwertes τ ist. τ wird meist anwendungspezifisch a priori festgelegt, es ist aber auch eine global adaptive bzw. lokal dynamische Anpassung denkbar.

3. Bestimme die Menge aller Ankerpunkte $\mathcal{D}_\perp = \{\mathbf{u} \mid \forall \mathbf{v} \in \mathcal{U}[\mathbf{u}] : m(\mathbf{u}) > m(\mathbf{v}), \mathbf{u} \in \mathcal{D}_\tau\}$, d.h. alle Punkte deren Merkmal innerhalb einer Umgebung $\mathcal{U}$ um den Punkt maximal ist. Während auch bei $\mathcal{U}$ eine adaptive Anpassung denkbare wäre, verwenden nahezu alle unten beschriebenen Detektoren die 4- oder 8-Nachbarschaft auf diskreten Bildern.

Für die Wahl des Merkmals *m* haben sich im Bereich der kovarianten Regionendetektoren vier dominante Ansätze herausgebildet: Auswertung des Grauwertstrukturtensors, der Hessematrix, der direkten Bildintensitäten sowie der Entropie. Sie werden in den folgenden Paragraphen vorgestellt.

Merkmale auf Basis des Grauwertstrukturtensors Der wohl bekannteste Ansatz zur Detektion markanter Punkte in Grauwertbildern ist 1988 von C. Harris und M. Stephens vorgestellt worden (HS88a). Er basiert auf dem *Grauwertstrukturtensor* $\mathbf{G}$:

$$\mathbf{G}(\mathbf{u}) = w[\sigma_i] * * \begin{pmatrix} \mathrm{i}^2_{[u]}(\mathbf{u}) & \mathrm{i}_{[u]}\mathrm{i}_{[v]}(\mathbf{u}) \\ \mathrm{i}_{[u]}\mathrm{i}_{[v]}(\mathbf{u}) & \mathrm{i}^2_{[v]}(\mathbf{u}) \end{pmatrix} \tag{3.12}$$

Der Grauwertstrukturtensor bildet das zweite Moment des Bildgradienten $\nabla \mathrm{i} = (\mathrm{i}_{[u]}, \mathrm{i}_{[v]})^{\mathrm{T}}$ bzw. beschreibt die Form der Autokorrelation des Bildes i über eine Fensterfunktion *w* um den Punkt $\mathbf{u}$. Die Ausdehnung der Fensterfunktion wird mit dem Parameter σ_i beschrieben, wobei *w* zur Vermeidung von Aliasing-Effekten an den Rändern auf 0 abfallen sollte. Für ein normiertes Ergebnis der über die Faltung realisierten Integration muss weiterhin $\int w[\sigma_i](\mathbf{x})\mathrm{d}\mathbf{x} = 1$ gelten. Ein weit verbreitetes Beispiel für *w* ist die zweidimensionale Normalverteilung $n[\mathbf{u}, \sigma_i \mathbf{I}]$ um den Mittelwert $\mathbf{u}$ und der Kovarianzmatrix $\sigma_i \mathbf{I}_{2\times 2}$.

Die Eigenwerte $\lambda_1(\mathbf{G})$ und $\lambda_2(\mathbf{G})$ des Grauwertstrukturtensors sind ein Maß für die Dominanz bestimmter Gradientenrichtungen innerhalb des Fensters bzw. für die Änderung der Autokorrelationsfunktion in diese Richtungen. Sind beide Eigenwerte groß, besitzt die betrachtete Struktur zwei bevorzugte, orthogonale Richtungen bzw. die Autokorrelation ändert sich in alle Richtungen stark.

Man bezeichnet die Struktur in diesem Zusammenhang auch allgemein als *Grauwertecke*. Ist dagegen nur ein Eigenwert groß, existiert nur eine dominante Richtung und die Autokorrelation hat die Form eines Grates. Verschiebungen entlang des Grates ändern die Autokorrelation nur schwach. Die Struktur ist daher in Richtung des Grates selbstähnlich und man spricht ganz allgemein von einer *Grauwertkante*. Sind dagegen beide Eigenwerte klein, handelt es sich um eine (annähernd) homogene Struktur die in alle Richtungen selbstähnlich ist. Da selbstähnliche Strukturen nur schwer lokalisierbar sind, zeigt sich, dass vor allem Grauwertecken für viele Anwendungen, insbesondere auch für die Detektion von Ankerpunkten die nötige Struktur bzw. einen herausragenden Informationsgehalt besitzen (SM97; ST94). Aus der konkreten Definition der Grauwertecke leiten sich zwei bekannte Merkmale m ab:

Harris-Ecken Man erhält diese Strukturen durch Verwendung des in (HS88a) vorgeschlagenes Merkmals

$$m(\mathbf{u}) = \det \mathbf{G}(\mathbf{u}) - k \operatorname{tr}^2 \mathbf{G}(\mathbf{u})$$

Es begründet sich empirisch aus $\det \mathbf{G} = \lambda_1 \lambda_2$ und $\operatorname{tr} \mathbf{G} = \lambda_1 + \lambda_2$ und vermeidet eine aufwändige Berechnung des Eigensystems. Der Wert ist groß für Ecken, negativ für Kanten und klein for homogene Flächen. Der Faktor k wird meist auf 0.04 festgelegt (Roc03).

Shi-Tomasi-Merkmale J. Shi und C. Tomasi schlagen in (ST94) die direkte Auswertung der Eigenwerte vor:

$$m(\mathbf{u}) = \min(\lambda_1(\mathbf{G}), \lambda_2(\mathbf{G}))$$

Sie liefern ähnliche Strukturen wie die Harris-Ecken und sind im Bereich des Trackings weit verbreitet.

Die richtige Wahl des Schwellwerts τ sollte von der Aufnahmecharakteristik des Bildsensors abgeleitet werden. Er sollte signifikant größer als der Wert des Merkmals unter Verwendung eines realen, d.h. verrauschten homogenen Bildes sein.

Merkmale auf Basis der Hessematrix Anstatt wie beim Grauwertstrukturtensor Merkmale auf Basis der ersten Ableitung des Bildes i zu konstruieren, kann man auch auf den zweiten Ableitungen in Form der Hessematrix aufbauen (MTS$^+$05; Low04):

$$\mathbf{H}(\mathbf{u}) = \begin{pmatrix} \mathrm{i}_{[uu]}(\mathbf{u}) & \mathrm{i}_{[uv]}(\mathbf{u}) \\ \mathrm{i}_{[uv]}(\mathbf{u}) & \mathrm{i}_{[vv]}(\mathbf{u}) \end{pmatrix}$$

Ankerpunkte auf dieser Basis liegen im Zentrum von blobartigen Strukturen (vgl. (Lin98)) und lassen sich normalerweise robust lokalisieren. Zwei Merkmale m liegen nahe:

Laplace-Merkmal Die Laplace-Funktion wertet die Spur der Hessematrix aus:

$$m(\mathbf{u}) = \operatorname{tr} \mathbf{H}(\mathbf{u})$$

Sie lässt sich vergleichsweise schnell berechnen, da nur die zweifachen Ableitungen auf der Diagonale berechnet werden müssen. Nachteilig wirkt sich allerdings aus, dass das Laplace-Merkmal auch auf extrem lange, dünne Strukturen reagiert, die sich entlang ihrer Ausprägung schlecht lokalisieren lassen.

Determinanten-Merkmal Wertet man dagegen die volle Information der Hesse-Matrix in Form der Determinante aus

$$m(\mathbf{u}) = \det \mathbf{H}(\mathbf{u})$$

erhält man zwar kompaktere, robuster lokalisierbare Blob-Strukturen, muss aber auch eine zweifache Ableitung mehr berechnen.

Merkmale durch direkte Auswertung der Bildintensitäten Anstatt die erste oder zweite Ableitung des Bildes zu betrachten kann man auch robuste Ankerpunkte auf Basis der ursprünglichen Bildintensitäten bestimmen (TVG04; MCUP04). Um lokale Extrema ermitteln zu können, genügen die sehr einfachen Merkmale m:

Intensitätsmaxima $m(\mathbf{u}) = \mathrm{i}(\mathbf{u})$

Intensitätsminima $m(\mathbf{u}) = -\mathrm{i}(\mathbf{u})$

Sie sind einerseits schnell zu berechnen, andererseits aber auch sehr sensitiv gegenüber Beleuchtungsschwankungen. Eine intelligente Wahl der Umgebung $\mathcal{U}$ sowie ein dynamischer Schwellwert τ sind essentiell für eine robuste Berechnung dieser Ankerpunkte.

Merkmale auf Basis der Entropie Ankerpunkte auf Basis der Entropie leiten sich aus der Fragestellung nach herausragenden bzw. *salienten* Strukturen in Bildern ab (KB01). Begründet auf der Annahme das lokale Signal-Komplexität in Bildern selten ist, wird in (Gil98) Salienz als lokale Shannon-Entropy über eine Region $\mathcal{R}$ definiert:

$$\rho[\mathcal{W},\mathcal{R}] = -\sum_{d\in\mathcal{W}} p_{\mathcal{R}}(d)\mathrm{ld}\, p_{\mathcal{R}}(d)$$

Die Wahrscheinlichkeitsverteilung $P(d\in\mathcal{W}) = P(\mathrm{i}=d) = |\mathcal{R}|^{-1}H(\mathrm{i}(\mathbf{v})=d, \mathbf{v}\in\mathcal{R})$ über alle Werte $d\in\mathcal{W}$ des Bildes i wird dabei durch das normierte lokale Grauwerthistogramm H über die Region $\mathcal{R}$ geschätzt. $p(d)$ ergibt sich daher mittels des Kronecker-Symbols δ zu $h(d) = |\mathcal{R}|^{-1}\sum_{\mathbf{v}\in\mathcal{R}}\delta_{d,\mathrm{i}(\mathbf{v})}$. Für das Merkmal m gilt daher:

Entropymaxima $m(\mathbf{u}) = \rho[\mathcal{D},\mathcal{R}_{\mathbf{u}}]$

Da diese Definition der Salienz weder in allen Fällen ausreichend ist (z.B. haben stark texturierte Objekte ebenfalls eine hohe Entropie) noch in allen Fällen robuste Ankerpunkte gewährleistet werden können, müssen weitere Nachbearbeitungsschritte folgen.

3.2.1.2 Isotrope[4] Skalierung

Ein fundamentales Problem der Bildverarbeitung ist die Handhabung unterschiedlich großer bzw. skalierter Grauwertstrukturen im Bild, die durch verschieden entfernte Objekte innerhalb der Szene hervorgerufen werden. Im Bereich der kovarianten Regionenkonstruktion manifestiert sich dies unter anderem beim Bestimmen der Ankerpunkte, deren Verfahren direkt oder indirekt eine Nachbarschaft um die Pixel auswerten. Es stellt sich für die Algorithmen die Frage, welche Größe bzw. Skalierung dieser Umgebung für die vorliegende Aufgabe sinnvoll ist. Da nur in den seltensten Fällen die richtige Skalierung a priori bekannt ist, wurden in der Wissenschaft Strukturen erschaffen, die es ermöglichen Bildoperatoren auf multiplen Skalen durchzuführen. Diese Multiskalenstrukturen haben die Eigenschaft feine Bildmerkmale in kleinen Skalen vorzuhalten und grobe Merkmale in größeren Skalen. Beispiele dafür sind Pyramiden, Skalen-Räume oder Diffusions-Ansätze (Jäh05). Aufgrund des immensen Informationsgehalts von Bild-Skalen-Räume müssen auswertende Systeme sehr früh eine Informationsreduktion durch Einschränkung der betrachteten Skalierungen durchführen. Während viele expertenbasierten Systeme diese Auswahl a priori festlegen, benötigen automatische Systeme Methoden, um für jedes Bild diejenigen Skalen herauszufiltern, in denen sich interessante Bildmerkale verbergen. Im Bereich der kovarianten Regionendetektion ist hierfür ein Verfahren von Lindeberg

[4] Gleichmäßige Skalierung in alle Richtungen wie sie im ähnlichen Modell verwendet wird.

(Lin98) weit verbreitet. Im folgenden wird nur die einfachen Darstellung mit einer isotropen Skalierung s beschrieben, auf die allgemeinere anisotrope Formulierung wird im Folgenden Unterabschnitt 3.2.1.3 eingegangen.

Ausgangspunkt ist die *Skalenraum-Repräsentation* $\mathrm{l} : \mathcal{D} \times \mathbb{R}^+ \to \mathcal{W}$ eines Bildes $\mathrm{i} : \mathcal{D} \subseteq \mathbb{R}^2 \to W \subseteq \mathbb{R}$, definiert durch die Faltung des Bildes mit einem Gaußkernel[5] variabler Breite $t = \sigma^2$:

$$\begin{aligned} \mathrm{l}(u,v,t) &= \mathrm{g}(u,v,t) * * \mathrm{i}(u,v) \\ \mathrm{g}(u,v,t) &= \frac{1}{2\pi t} e^{\frac{-(u^2+v^2)}{2t}} \end{aligned}$$

Eng verbunden mit dem Skalenraum ist der Gaußche Ableitungsoperator der Ordnung n:

$$\mathrm{l}_{x^n}(\cdot,t) = (\partial_{x^n}\mathrm{l})(\cdot,t) = \partial_{x^n}(\mathrm{g}(\cdot,t) * * \mathrm{i}(\cdot)) = (\partial_{x^n}\mathrm{g}(\cdot,t)) * * \mathrm{i}(\cdot) = \mathrm{g}(\cdot,t) * * (\partial_{x^n}\mathrm{i}(\cdot))$$

mit dessen Hilfe eine gradientenbasierte Auswertung des Skalenraums möglich ist. Es zeigt sich, dass der Gaußkernel unter den linearen Transformationen der einzige Kernel ist, der alle Eigenschaften erfüllt um einen Skalenraum zu erzeugen. Diese Eigenschaften sind Linearität, Verschiebungsinvarianz und verschiedene Formulierungen zur Vermeidung von Scheinstrukturen bei der Vergröberung der Skalierung. Interessanterweise zeigt sich eine enge Verwandschaft dieser Multiskalenauswertung zum visuellen System von Säugetieren (vgl. (Lin98; LG94)).

Eine naive Anwendung des gaußchen Gradientenoperators führt bei Skalen-Räumen allerdings zu einem Problem, da dessen Wert - durch eine Abnahme der räumlichen Amplitude des Signals - mit steigender Skalierung sinkt. Dies hängt mit dem Minimum-Maximum-Prinzip zusammen, welches besagt, dass mit zunehmender Glättung bzw. Diffusion Amplitudenmaxima nur schrumpfen bzw. Amplitudenminima nur zunehmen können. Deshalb führt Lindeberg einen *gamma-normalisierten* Gradienten bzw. Ableitunsoperator

$$\mathrm{l}_{x,\gamma}(\cdot,t) = t^{\frac{\gamma}{2}}\mathrm{l}_x(\cdot,t)$$

ein. Dessen Motivation begründet sich u.a. auf theoretischen Untersuchen (Lin98) einer Sinusschwingung $\mathrm{i}(u) = \sin\omega u$ bzw. deren assoziierten Skalenraum $\mathrm{l}(u,t) = e^{-\omega^2\frac{t}{2}}\sin\omega u$. Für den gamma-normalisierten Gradient n-ter Ordnung ergeben sich (insbesondere für $\gamma = 1$) bei Betrachtung seiner Amplitude $\mathrm{l}_{x^n,\gamma,\max}(t) = t^{n\frac{\gamma}{2}}\omega^n e^{-\omega^2\frac{t}{2}}$ über die Skalierung zwei wichtige Eigenschaften:

Kovariante Lage des Amplituden-Maximums Die Amplitude $\mathrm{l}_{x^n,\gamma,\max}(t)$ steigt mit zunehmender Skalierung t zuerst an, fällt dann ab und erreicht ihr einziges Maximum bei $t_{\max,\gamma,n} = \gamma\frac{n}{\omega^2}$. Durch Einführung der Wellenlänge $\lambda = \frac{2\pi}{\omega}$ und des Skalenparameters $s = \sqrt{t}$ zeigt sich, dass sich die Lage des Maximum bzgl. der Skalierung s proportional zur Wellenlänge λ des Signals verhält:

$$s_{\max,\gamma,n} = \frac{\sqrt{\gamma n}}{2\pi}\lambda \overset{\gamma=1}{=} \frac{\sqrt{n}}{2\pi}\lambda$$

Die Lage des Maximums entspricht der größten Korrelation des Gradienten-Operators mit dem Signal i und kann als charakteristische Länge (bzw. Größe bei Betrachtung als Grauwertmuster) von i interpretiert werden. Es ändert sich kovariant mit einer Reskalierung von $\mathrm{i}(\cdot)$.

Konstanter Wert des Amplituden-Maximums Betrachtet man den Wert des Maximums

$$\mathrm{l}_{x^n,\gamma,\max}(s^2_{\max,\gamma,n}) = \left(\frac{n\gamma}{e}\right)^{\frac{\gamma n}{2}} \omega^{(1-\lambda)n} \overset{\gamma=1}{=} \left(\frac{n}{e}\right)^{\frac{n}{2}}$$

zeigt sich, dass dieser für $\gamma = 1$ unabhängig der Frequenz ω bzw. der Skalierung $s^2_{\max,\gamma,n}$ ist. Interpretiert man den Wert des Amplituden-Maximums als Stärke des Signals, kann man dieses über verschiedene Skalierungen direkt miteinander vergleichen.

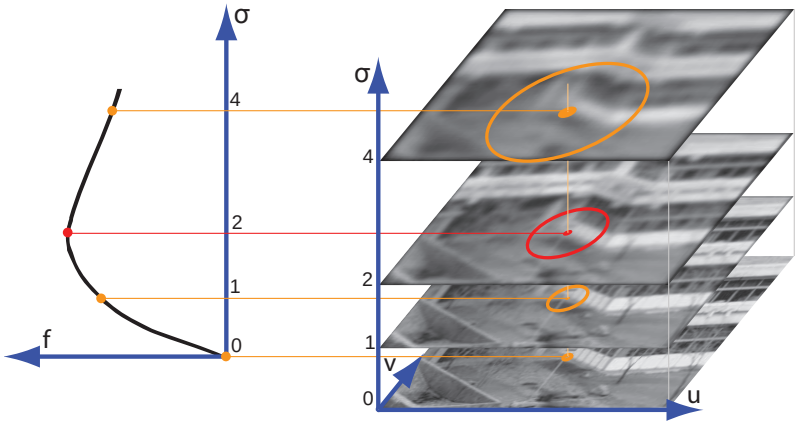

Abbildung 3.2: Darstellung des Skalenraums und dem Prinzip der automatischen Skalenselektion für eine beliebige Kombination f von normalisierten Ableitungen. Bilder des Skalenraums aus (Jäh05).

Diese zwei Eigenschaften führen dazu, dass man die Größe von Signalmustern bei Betrachtung eines *einzelnen* räumlichen Punktes durch die Änderung von beliebig kombinierten Gradienten-Merkmalen n-ter Ordnung in der Skalen-Dimension bestimmen kann und weiterhin unterschiedlich skalierte Ausprägung des Musters objektiv miteinander vergleichen kann. Nach Lindeberg (Lin98) beschränken sich diese Auswertungsmöglichkeiten nicht auf Sinus-Schwingungen, sondern lassen sich auf beliebige Signalmuster anwenden (übersetzt):

> *Ohne weitere Anhaltspunkte kann ein Skalierungslevel, an der eine beliebige (möglicherweise nichtlineare) Kombination normalisierter Ableitungen ein Maximum über die Skalierung erreicht, so behandelt werden, als würde es einer charakteristischen Länge eines korrespondierenden Musters in den Daten entsprechen.*

Von diesem Prinzip der automatischen Skalen-Selektion (siehe auch Abb. 3.2) machen viele Detektoren (MTS+05; MS04; Low04) gebrauch. Die konkrete Anwendung dieses Prinzips wird in den entsprechenden Unterabschnitten 3.2.2 und 3.2.3 der ableitungsbasierten Detektoren behandelt. Sie verwenden entweder den angepassten Grauwertstrukturtensor $\mathbf{G}_l$ mit einem Gaußkernel als Fensterfunktion

$$\mathbf{G}_{\mathrm{l}}(\mathbf{u},\sigma_i,\sigma_t) = \sigma_t^2 g(\mathbf{u},\sigma_i) * * \begin{pmatrix} \mathrm{l}^2_{[u]}(\mathbf{u},\sigma_t) & \mathrm{l}_{[u]}\mathrm{l}_{[v]}(\mathbf{u},\sigma_t) \\ \mathrm{l}_{[u]}\mathrm{l}_{[v]}(\mathbf{u},\sigma_t) & \mathrm{l}^2_{[v]}(\mathbf{u},\sigma_t) \end{pmatrix} \tag{3.13}$$

oder die angepasste Hesse-Matrix

$$\mathbf{H}_{\mathrm{l}}(\mathbf{u},\sigma_t) = \sigma_t^2 \begin{pmatrix} \mathrm{l}_{[uu]}(\mathbf{u},\sigma_t) & \mathrm{l}_{[uv]}(\mathbf{u},\sigma_t) \\ \mathrm{l}_{[uv]}(\mathbf{u},\sigma_t) & \mathrm{l}_{[vv]}(\mathbf{u},\sigma_t) \end{pmatrix} \tag{3.14}$$

σ_i beschreibt die Größe über die $\mathbf{G}$ integriert wird und σ_t die lokale Skale auf der $\mathbf{G}$ bzw. $\mathbf{H}$ berechnet wird. Der Vorfaktor σ_t^2 ergibt sich durch die Verwendung des 1-normierten Gradienten.

3.2.1.3 Anisotrope[6] Skalierung

Das Prinzip der automatischen Skalierungsselektion auf der (isotropen) Skalenraum-Repräsentation hat Nachteile bei Verfahren, die anisotrope Muster in den Signalen auswerten. Durch die isotrope

[5] Alternativ lässt sich der Skalenraum $\mathrm{l}(u,v,t)$ auch als Lösung der Differentialgleichung $\partial_t \mathrm{l} = \frac{1}{2}\nabla^2 \mathrm{l} = \frac{1}{2}\partial_{uu}\mathrm{l} + \partial_{vv}\mathrm{l}$ mit Anfangsbedingung $\mathrm{l}(\cdot,0) = \mathrm{i}(\cdot)$ definieren. Diese Darstellung hat für analytische Betrachtung oftmals Vorteile, ist aber im Bereich der Bildverarbeitung weniger intuitiv als die Definition mit einem Gaußkernel.

[6] Unterschiedliche Skalierung in orthogonalen Richtungen wie sie im affinen Modell verwendet wird.

Glättung des Gauß-Kernels wird bei gröberer Skalierung die Anisotropie immer weiter unterdrückt und führt daher zu einer Unterschätzung der Scherung und der orthogonalen Skalierungsunterschiede. Ein weiteres Problem ist die Nicht-Abgeschlossenheit der isotropen Skalenraum-Repräsentation bzgl. der affinen Gruppe. Geht Bild bzw. dessen Skalenraum $\mathrm{l}_2(\mathbf{u}_2,t_2)=\mathrm{l}_1(\mathbf{u}_1,t_1)$ durch eine geometrische Transformation $\mathbf{u}_2=\mathbf{geo}(\mathbf{u}_1)$ hervor, so ist der Skalenraum l zwar bzgl. der Ähnlichkeitsgruppe $\mathbf{geo}_\mathrm{S}(\mathbf{u})=s\mathbf{R}\mathbf{u}+\mathbf{t_u}$ durch den rotationssym. Gauß-Kernel abgeschlossen: $\mathrm{l}_2(\mathbf{geo}_\mathrm{S}^{-1}(\mathbf{u}_2),s^{-2}t_2)=\mathrm{l}_1(\mathbf{u}_1,t_1)$. Für die allgemeinere affine Gruppe $\mathbf{geo}_\mathrm{A}(\mathbf{u})=\mathbf{A}\mathbf{u}+\mathbf{t_u}$ lässt sich aber keine Transformation (des Skalenparameters t_2) angeben, damit $\mathrm{l}_2(\cdot)=\mathrm{l}_1(\cdot)$ erfüllt ist. Eine optimale *kovariante* Skalenselektion ist daher für das affine Transformationsmodell nicht gegeben. Daher wird in (LG94) die *affine* Skalenraum-Repräsentation mit einem *adaptiven* Gauß-Kernel mit Kovarianzmatrix Σ eingeführt[7]:

$$\begin{aligned}\mathrm{l}(\mathbf{u},\Sigma) &= \mathrm{g}(\mathbf{u},\Sigma)**\mathrm{i}(\mathbf{u})\\ \mathrm{g}(\mathbf{u},\Sigma) &= \frac{1}{2\pi\sqrt{\det\Sigma}}e^{-\frac{1}{2}\mathbf{u}^\mathrm{T}\Sigma_t^{-1}\mathbf{u}}\end{aligned}$$

Diese Repräsentation ist über $\mathrm{l}_2(\mathbf{geo}_\mathrm{A}^{-1}(\mathbf{u}_2),\mathbf{A}^{-1}\Sigma_2\mathbf{A}^{-\mathrm{T}})=\mathrm{l}_1(\mathbf{u}_1,\Sigma_1)$ ebenfalls bzgl. der affinen Transformationsgruppe abgeschlossen. Die Glättung muss in diesem Fall nicht in alle Richtungen gleichmäßig erfolgen, sondern kann an jedem Punkt adaptiv an die lokale Anisotropie angepasst werden, um diese über größere Skalierungen zu erhalten. Seien konkret $\lambda_{1,2}$ die Eigenwerte von Σ^{-1} mit zugehörigen Eigenvektoren $\upsilon_{1,2}$, dann erfolgt die Glättung entlang υ_1 proportional zu λ_1 und separabel davon entlang υ_2 proportional zu λ_2.

Für eine adaptive Glättung muss in einem ersten Schritt die Anisotropie bzw. affine Störung in einem Bildpunkt $\mathbf{u}$ gemessen werden. Dazu bedient sich (LG94) dem aus Formel 3.12 bekannten Grauwertstrukturtensor $\mathbf{G}$, der auf den affinen Skalenraum angepasst wurde und als Fensterfunktion ebenfalls den Gauß-Kernel nutzt:

$$\mathbf{G}(\mathbf{u},\Sigma_i,\Sigma_t)=\det\Sigma_t\,\mathrm{g}(\mathbf{u},\Sigma_i)**\begin{pmatrix}\mathrm{l}^2_{[u]}(\mathbf{u},\Sigma_t) & \mathrm{l}_{[u]}\mathrm{l}_{[v]}(\mathbf{u},\Sigma_t)\\ \mathrm{l}_{[u]}\mathrm{l}_{[v]}(\mathbf{u},\Sigma_t) & \mathrm{l}^2_{[v]}(\mathbf{u},\Sigma_t)\end{pmatrix} \tag{3.15}$$

Die Kovarianzmatrix Σ_i beschreibt dabei die Form und Größe der integrativen Fensterfunktion, Σ_t die lokale, adaptiv geglättete Skale auf Basis derer der Strukturtensor berechnet wird. Da man allerdings zur Berechnung von $\mathbf{G}$ schon Σ kennen muss, dieses sich aber erst mittels $\mathbf{G}$ bestimmen lässt, benötigt man ein iteratives Schema um sich schrittweise dem richtigen Ergebnis nähern zu können.

In seiner allgemeinsten Form muss man für jeden Bildpunkt $\mathbf{u}$ sechs Parameter für die Kovarianzmatrizen Σ_i und Σ_t bestimmen. Es ist in der Praxis allerdings ausreichend, beide Matrizen über eine Grundform Σ_0 miteinander zu koppeln: $\Sigma_i=i\Sigma_0$ bzw. $\Sigma_t=t\Sigma_0$. Daher reduziert sich das Problem auf 4 zu bestimmende Parameter. Um die Anisotropie zu erhalten, ist es nun wünschenswert entlang ausgedehnter Strukturen (z.B. Kanten) stärker zu glätten als entlang gestauchter Strukturen (z.B. starke Gradienten). Daher sollte die Grundform $\Sigma_0=\mathbf{G}(\mathbf{u},\cdot)$ proportional zum inversen Strukturtensor gewählt werden. Die Größe i des Integrationsfensters wird in der Praxis oftmals konstant gewählt, es kann aber ebenfalls adaptiv so gewählt werden, dass eine gewünschte Kombination von Ableitungsoperatoren (z.B. $\det\mathbf{G}$) über i maximal werden. Die Stärke t der anisotropen Glättung im Skalenraum wird so gewählt, dass eine minimale Glättung in beliebiger Richtung garantiert werden kann. Dadurch soll verhindert werden, dass entlang gestauchter Strukturen weniger geglättet wird, als dies z.B. bei isotropen Strukturen der Fall wäre und daher die Anisotropi noch verstärkt wird. Erreicht wird dies durch eine Normierung der Grundform mit ihrem kleinsten Eigenwert: $t=\lambda_{\min}(\Sigma_0)$. Zusammenfassend lässt sich die Iterationsvorschrift nach (LG94) so angeben:

[7] Analog zum isotropen Skalenraum kann der affine Skalenraum ebenfalls durch einen Diffusionsprozess erzeugt werden. Dieser Prozess ist jetzt allerdings anisotrop und passt den Diffusions-Fluss den lokalen Bildverhältnissen an. Für weiterführende Grundlagen sei auf (Jäh05), Kap. 17.3 verwiesen.

1. Wähle die Startbedingung so annahmslos wie möglich: $\Sigma_0^{[0]} = \mathbf{I}_{2\times 2}$, $i,t \in \mathbb{R}^+$
2. Bestimme die Glättungsstärke $t^{[k]} = t\lambda_{\min}(\Sigma_0^{[k]})$
3. Bestimme die Integrationsskale $i^{[k]} = i$ oder optional so, dass eine Kombination von Ableitungsoperatoren $f(\mathrm{l}(\mathbf{u},\cdot))$ maximal wird: $i^{[k]} = \max_j f(\mathrm{l}(\mathbf{u}, jt^{[k]}\Sigma_0^{[k]}))$.
4. Bestimme die neue Grundform $\Sigma_0^{[k+1]} = \mathbf{G}^{-1}(\mathbf{u}, i^{[k]}\Sigma_0^{[k]}, t^{[k]}\Sigma_0^{[k]})$.
5. Springe zu 2. oder breche ab, falls eine feste Anzahl von Iterationen durchlaufen wurde bzw. ein anwendungsabhängiges Abbruchskriterium erfüllt wurde.

Betrachtet man noch einmal zwei Skalenräume $\mathrm{l}_2(\mathbf{u}_2,\Sigma_{i,2},\Sigma_{t,2}) = \mathrm{l}_1(\mathbf{u}_1,\Sigma_{i,1},\Sigma_{t,1})$ die durch eine affine geometrische Transformation $\mathbf{u}_2 = \mathbf{geo}_{\mathrm{A}}(\mathbf{u}_1) = \mathbf{A}\mathbf{u}_1 + \mathbf{t}_\mathbf{u}$ auseinander hervorgehen, so erfüllen Punkte $\mathbf{u}$, deren Grauwertstrukturtensor G mittels obigen Verfahren berechnet wurde zwei Eigenschaften:

Invarianz Eigenschaft fixer Punkte Erfüllt $\mathbf{u}_1$ bestimmte affin invariante räumliche Eigenschaften in $\mathrm{l}_1(\mathbf{u}_1,\Sigma_{i,1},\Sigma_{t,1})$, wie z.B. ein räumliches Maximum von $\det\mathbf{G}$, so tut dies auch der transformierte Bildpunkt $\mathbf{u}_2$ in $\mathrm{l}_2(\mathbf{u}_2,\Sigma_{i,2},\Sigma_{t,2})$.

Kovariante Eigenschaft des Grauwertstrukturtensors Für die beiden Strukturtensoren gilt folgender Zusammenhang: $\mathbf{G}_1(\mathbf{u}_1,\Sigma_{i,1},\Sigma_{t,1}) = \mathbf{A}^{\mathrm{T}}\mathbf{G}_2(\mathbf{geo}(\mathbf{u_1}),\mathbf{A}\Sigma_{i,1}\mathbf{A}^{\mathrm{T}},\mathbf{A}\Sigma_{t,1}\mathbf{A}^{\mathrm{T}})\mathbf{A}$. Der Strukturtensor verhält sich also bis auf einen Rotationsanteil *kovariant* zu $\mathbf{A}$.

Aufgrund dieser Eigenschaften kann die affine Skalenraum-Repräsentation herangezogen werden, um den anisotropen Skalierungsanteil $\mathbf{S}_a$ des affinen Modells kovariant zu bestimmen. Die Detektoren in Abschnitt 3.2.2.2 und 3.2.3.2 machen davon Gebrauch.

3.2.1.4 Rotation

Hat man für einen Ankerpunkt $\mathbf{u}_\perp$ eine elliptische bzw. kreisförmige Region $\mathcal{R}[\mathbf{u}_\perp]$ konstruiert, verbleibt noch ein weiterer Freiheitsgrad: die Rotation φ in der Bildebene des zugrundeliegenden Intensitäts-Musters innerhalb von $\mathcal{R}$. Während dies für kreisförmige Regionen intuitiv klar ist, müssen elliptische Regionen $\mathcal{R}[\mathbf{u}_\perp,\mathbf{A}] = \{\mathbf{u} \mid (\mathbf{u}-\mathbf{u}_\perp)^{\mathrm{T}}\mathbf{A}(\mathbf{u}-\mathbf{u}_\perp = 1\}$ erst auf ihre isotrope, kreisförmige Form $\mathbf{A} = \mathbf{I}$ normiert werden. Dies erfolgt durch eine Koordinatensystemtransformation ${}^{\mathrm{N}}\mathbf{u} = \sqrt{\mathbf{A}}\mathbf{u}$, die man auf das Bild und die Region anwenden muss. Aus der elliptischen Darstellung $\mathbf{u}^{\mathrm{T}}\sqrt{\mathbf{A}}^{\mathrm{T}}\sqrt{\mathbf{A}}\mathbf{u} = \mathbf{u}^{\mathrm{T}}\sqrt{\mathbf{A}}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\sqrt{\mathbf{A}}\mathbf{u} = (\mathbf{R}\sqrt{\mathbf{A}}\mathbf{u})^{\mathrm{T}}\mathbf{R}\sqrt{\mathbf{A}}\mathbf{u} = 1$ folgt auch der verbleibende rotative Freiheitsgrad $\mathbf{R}$ in der normalisierten Darstellung (Bau00).

Beschränkt man sich bei der Wahl des Deskriptors auf rotationsinvariante Verfahren, muss man diesen Freiheitsgrad nicht bestimmen, da sich der physikalische Bildinhalt innerhalb der Region nicht verändert (vlg. z.B. (SM97)). Da diese Einschränkung im Allgemeinen aber nicht erwünscht ist, werden im folgenden Verfahren vorgestellt um eine charakteristische Richtung des Musters zu bestimmen. Damit lässt sich eine gerichtete, vollständig kovariante Konstruktion der Regionen erreichen. Ausgangspunkt ist immer ein Muster $\mathcal{M}[\mathrm{i}(\cdot),\mathcal{R}[\mathbf{u}_\perp]]$ einer kreisförmigen Region um den Ankerpunkt $\mathbf{u}_\perp$.

Richtungbestimmung mittels Gradienten-Orientierungs-Histogramm In (Low04) wird ein empirisches Verfahren zur Richtungsbestimmung auf Basis von Gradienten-Vorzugsrichtungen beschrieben. Das Verfahren besteht aus mehreren Schritten:

1. Als erstes werden für alle Punkte $\mathbf{u} \in \mathcal{R}$ der Betrag $m(\mathbf{u})$ und die Richtung $\varphi(\mathbf{u})$ des Gradienten mittels Pixeldifferenzen ermittelt:

$$\begin{aligned} m(\mathbf{u}) &= \sqrt{(\mathrm{i}(\mathbf{u}+\mathbf{e}_u)-\mathrm{i}(\mathbf{u}-\mathbf{e}_u))^2+(\mathrm{i}(\mathbf{u}+\mathbf{e}_v)-\mathrm{i}(\mathbf{u}-\mathbf{e}_v))^2} \\ \varphi(\mathbf{u}) &= \arctan\frac{\mathrm{i}(\mathbf{u}+\mathbf{e}_v)-\mathrm{i}(\mathbf{u}-\mathbf{e}_v)}{\mathrm{i}(\mathbf{u}+\mathbf{e}_u)-\mathrm{i}(\mathbf{u}-\mathbf{e}_u)} \end{aligned}$$

2. In einem weiteren Schritt wird ein Richtungshistogramm mit k Einträgen über den gesamten Winkelbereich von 2π gebildet. Dort wird jeder Punkt $\mathbf{u} \in \mathcal{R}[\mathbf{u}_\perp]$ gemäß seiner Gradientenrichtung $\varphi(\mathbf{u})$, gewichtet mit dem Betrag $m(\mathbf{u})$ und einer gaußchen Fensterfunktion mit Mittelwert $\mathbf{u}_\perp$ eingetragen.

3. Peaks in dem Richtungshistogramm entsprechen Vorzugsrichtungen des Musters und eignen sich zur kovarianten Konstruktion. Sollten mehrere dominante Peaks über $p\,\%$ des Hauptpeaks in dem Histogramm enthalten sein, wird in (Low04) für jeden weiteren Peak eine eigene Region gebildet.

4. Abschließend wird für jede Region an die drei größten Einträge des zugehörigen Peaks im Richtungshistogramm eine Parabel angefittet. Der Scheitelpunkt an der Stelle φ entspricht dann der interpolierten, endgültigen Vorzugsrichtung.

Die Anzahl k der Histogramm-Einträge wird in (Low04) mit 36 angeben, die Schwelle p für die multiplen Richtungen mit 80%.

Richtungbestimmung mittels Grauwert-Halbachsen In (TVG04) wird die große Grauwert-Halbachse durch den Ankerpunkt $\mathbf{u}_\perp = (u_\perp, v_\perp)^\mathrm{T}$ als charakteristische Richtung herangezogen. Diese wird lokal über $\mathcal{R}[\mathbf{u}_\perp]$ mittels modifizierter Momente

$$m_{pq} = \iint_{\mathcal{R}[\mathbf{u}_\perp]} \mathrm{i}(\mathbf{u})(u-u_\perp)^p(v-v_\perp)^q \mathrm{d}\mathbf{u}$$

bestimmt, die nicht mit dem Grauwert-Schwerpunkt zentriert sind, sondern mit $\mathbf{u}_\perp$. Dadurch erreicht man eine photometrisch invariante Konstruktion. Der Winkel φ der großen Halbachse ergibt sich als eine Lösung von $\tan^2\varphi + \frac{m_{20}-m_{02}}{m_{11}}\tan\varphi - 1 = 0$. Durch trigonometrische Umformung mittels der Doppelten-Winkel-Identität des Tangens erhält man

$$\varphi = \frac{1}{2}\arctan\frac{2m_{11}}{m_{20}-m_{02}}$$

3.2.2 Konstruktion auf Basis des Grauwertstrukturtensors

Alle hier beschriebenen Verfahren bestimmen ihre räumliche Lage im Bild, genauer ihres Ankerpunktes durch eine Auswertung des Grauwertstrukturtensors (siehe auch 3.2.1.1). Es sind *Harris-Similarity-Regions* (HaS), *Harris-Affine-Regions* (HaA) und *Edge-Based-Regions* (EBR). Abbildung 3.3 zeigt Beispiele einiger Verfahren.

3.2.2.1 HaS: Harris-Similar

Ausgangspunkt für die Harris-Similarity-Regions (MS04), in dieser Arbeit kurz *HaS* genannt, sind Harris-Ecken auf Basis des Grauwertstrukturtensors $\mathbf{G}$. Um diese wird ein orientierter Kreis konstruiert, der sich kovariant bzgl. des Ähnlichkeitsmodells $\mathbf{geo}_\mathrm{S}(\cdot)$ verhält. Es muss daher noch eine

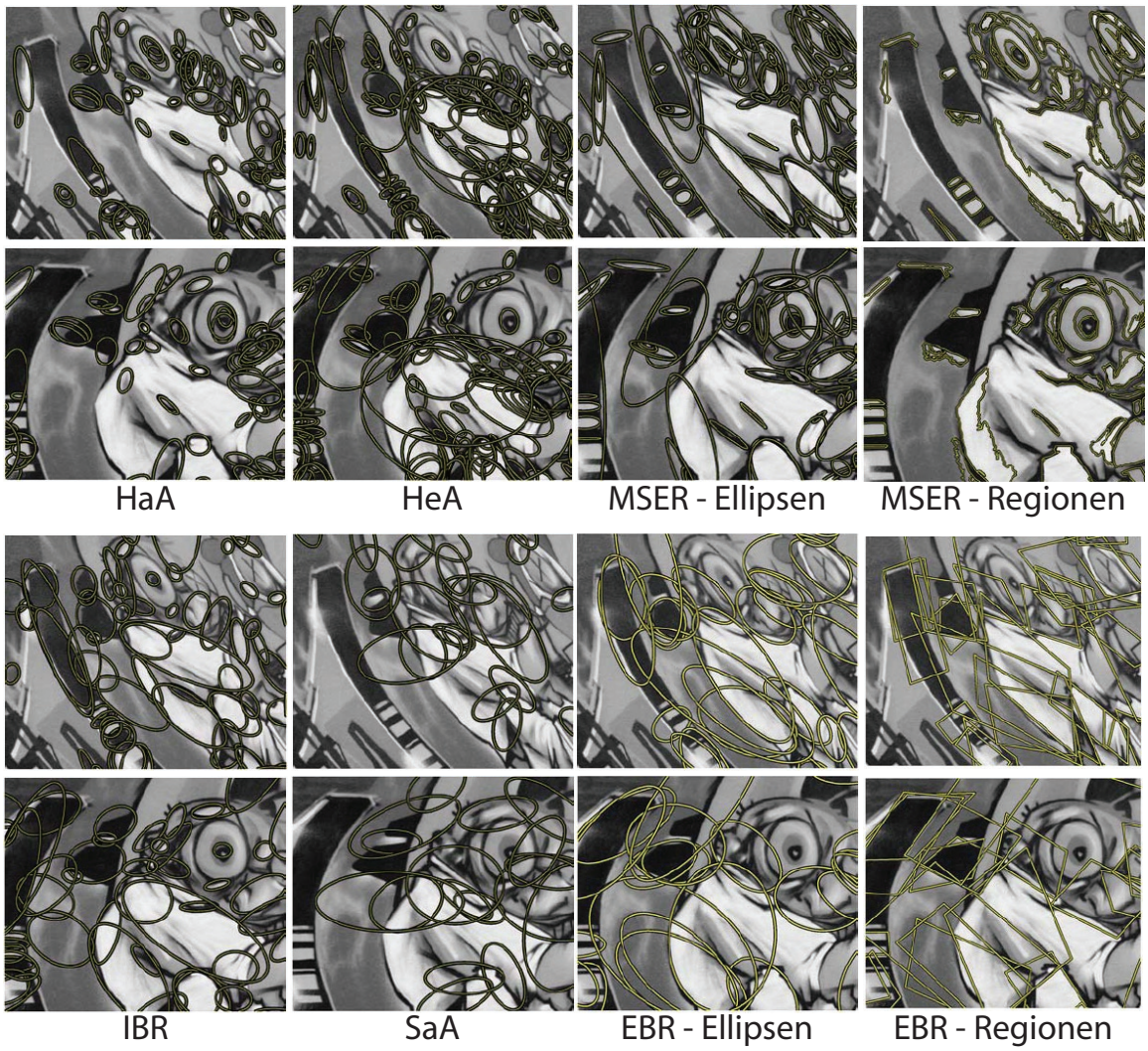

Abbildung 3.3: Beispiele affiner Regionen aus (MTS$^+$05). Für die MSER und EBR sind die tatsächlich konstruierten Regionen und die nachträglich angefitten Ellipsen dargestellt.

charakteristische Skale s und eine Orientierung φ gefunden werden. Zur Bestimmung der Skale wird das in Abschnitt 3.2.1.2 beschriebene Verfahren mittels 1-normierter Gradienten auf Basis des Skalenraums verwendet (Lin98). Harris-Ankerpunkte werden daher mit dem angepassten Grauwertstrukturtensor $\mathbf{G}_1(\mathbf{u},\sigma_i,\sigma_t)$ aus Gl. 3.13 berechnet: $m_{\text{Harris}}(\mathbf{u},\sigma_i,\sigma_t) = \det \mathbf{G}_1(\mathbf{u},\sigma_i,\sigma_t) - k\,\text{tr}^2\mathbf{G}_1(\mathbf{u},\sigma_i,\sigma_t)$. Nach (MS01) ist m_{Harris} allerdings nur zur räumlichen Maximumsbestimmung geeignet, da sich über die Skalierung keine vernünftigen Maximas ausprägen. Daher wird die Bestimmung der charakteristischen Skale mittels der angepassten Laplace-Funktion $m_{\text{LoG}} = \text{tr}\,\mathbf{H}_1(\mathbf{u},\sigma_t)$ (siehe Gl. 3.14) durchgeführt. Diese Herangehensweise mit dem *Laplacian-of-Gaußians*-Kernel kann als Matched-Filter (DHS00) verstanden werden, die nicht nur für Blob-Strukturen funktioniert, sondern auch für viele andere ausgeprägten Muster, wie Kanten, Kreuzungen o. ä. Für die Maximumssuche im Skalenraum schlägt (MS04) zwei konkrete Methoden vor:

Iterative Bestimmung des Maximums im Skalenraum (HaS) Zuerst wird der Skalenraum für diskrete Skalen $\sigma_n = \xi^n\sigma_0$ aufgebaut und dort nach räumlichen Maxima $\mathcal{D}_\perp$ von $m_{\text{Harris}}(\sigma_i,\sigma_t)$ mittels

der 8-Nachbarschaft gesucht (vgl. Abschnitt 3.2.1.1). Die Parameter $\sigma_i = \sigma_n$ und $\sigma_t = t\sigma_n$ mit $t = 0.7$ werden bzgl. der diskreten Skalen σ_n gesetzt. Für die räumlichen Ankerpunkte $\mathbf{u}_\perp \in \mathcal{D}_\perp$ wird nun iterativ die Skale und deren räumliche Position verfeinert:

1. Initialisiere den räumlichen Punkt mit $\mathbf{u}^{[0]} = \mathbf{u}_\perp$ und der assoziierten Skale $\sigma^{[0]} = \sigma_n$.
2. Finde Maxima über die Skale $\sigma^{[k+1]} = \max_{\xi^{-1} < i < \xi} m_{LoG}(\mathbf{u}^{[k]}, \sigma^{[k]})$ mittels des Laplace-Kernels. Sollte das Maximum zu schwach ausgeprägt sein, ignoriere es.
3. Verfeinere $\mathbf{u}^{[k]}$ durch Bestimmung des nächsten Maximus $\mathbf{u}^{[k+1]}$ von $m_{\text{Harris}}(\mathbf{u}, \sigma^{[k+1]}, t\sigma^{[k+1]})$ auf der neuen Skale.
4. Brich ab falls $\sigma^{[k+1]} = \sigma^{[k]}$ und $\mathbf{u}^{[k+1]} = \mathbf{u}^{[k]}$ oder nach einer festen Anzahl an Iterationen.

Da das räumliche Maximum innerhalb der Iteration noch einmal verfeinert wird, genügt es den initialen Skalenraum grob mittels $\xi = 1.4$ zu samplen. Da mehrere initialen Punkte aus $\mathcal{D}_\perp$ zu derselben Struktur konvergieren können, müssen in einem Nachbearbeitungschritt alle Duplikate entfernt werden.

Schnelle approximierte Bestimmung des Maximums im Skalenraum (HaSF) Da innerhalb der Iteration der Skalenraum lokal neu aufgebaut werden muss, ist obiges Verfahren relativ langsam. Verzichtet man auf etwas Präzision kann man den intialen Skalenraum auch mit z.B. $\xi = 1.2$ wesentlich feiner samplen und dort die räumlichen Maximas $\mathbf{u}_\perp \in \mathcal{D}_\perp$ bestimmen. Durch die feinere Diskretisierung spart man sich die Iteration und muss nur noch überprüfen, ob m_{LoG} bzgl. der benachbarten Skalen ebenfalls ein Maximum ausprägt.

Bestimmung der Rotation für HaS und HaSF Eine Bestimmung der Rotation wird in (MS04) nicht explizit angegeben. Da man allerdings durch die gefundene Skale nun die Region kennt, kann kann jedes der in Abschnitt 3.2.1.4 beschriebene Verfahren verwendet werden.

3.2.2.2 HaA: Harris-Affine

Der Harris-Affine-Detektor (HaA) aus (MS04) ist eine Erweiterung der HaS bzw. HaSF auf eine affin invariante Konstruktion der Region. Ausgangspunkt sind die Ergebnisse von HaS bzw. HaSF, d.h. Harris-Ecken $(\mathbf{u}_\perp, \sigma_i)$ mit assoziierter Skale im Skalenraum. Mit Hilfe des affinen Skalenraums wird in einem iterativen Verfahren (siehe Abschnitt 3.2.1.3 bzw. (Bau00)) die kreisförmige Region solange transformiert, bis das Anisotropi-Maß $q_A(\mathbf{G}_\text{l}) = \frac{\lambda_{\min}(\mathbf{G}_\text{l})}{\lambda_{\max}(\mathbf{G}_\text{l})}$ des angepassten Strukturtensors $\mathbf{G}_\text{l}(\mathbf{u}, \Sigma_i, \Sigma_t)$ aus Gl. 3.15 maximal wird.

Aus praktischen Gesichtspunkten ist es besser, das Bild i$(\mathbf{u})$ bzw. einen lokalen Ausschnitt affin in eine isotrope Struktur zu verzerren als es anisotrop zu glätten. Dadurch kann man auf den Ergebnissen des letzten Iterationsschrittes aufbauen und rekursive isotrope Gaußkernel einsetzen. Die Verzerrungsmatrix für einen Iterationsschritt k ergibt sich dann zu

$$\mathbf{U}^{[k]} = \prod_{i=0}^{k} (\mathbf{G}_\text{l}^{-\frac{1}{2}})^{[i]} = (\mathbf{G}_\text{l}^{-\frac{1}{2}})^{[k]} \mathbf{U}^{[k-1]}$$

Konkret ergibt sich folgender Algorithmus zur Detektion der HaA's:

1. Initialisiere $\mathbf{U}^{[0]} = \mathbf{I}_{2\times 2}$ so als wäre die betrachtete Struktur isotrop und setze $\mathbf{u}_\perp^{[0]}$ auf einen HaS bzw HaSF.

2. Berechne das verzerrte Bild ${}^{\mathrm{D}}\mathrm{i}^{[k]}((\mathbf{U}^{[k]})^{-1})\mathbf{u}) = \mathrm{i}(\mathbf{u})$ in der Region $R(\mathbf{u}_{\perp}^{[k]}, (\mathbf{U}^{[k]})^{\mathrm{T}}\mathbf{U}^{[k]})$ im ursprünglichen Bild.

3. Ermittle die Integrationsskale $\sigma_i^{[k]}$ mittels des Verfahrens von HaS bzw. HaSF auf ${}^{\mathrm{D}}\mathrm{i}^{[k]}$.

4. Ermittle die Glättungsskale $\sigma_t^{[k]} = \max_{0.5 \leq j \leq 0.75} q_A({}^{\mathrm{D}}\mathbf{G}_{\mathrm{l}}({}^{\mathrm{D}}\mathbf{u}_{\perp}^{[k]}, \sigma_i^{[k]}, j\sigma_i^{[k]}))$ auf der die Anisotropi in ${}^{\mathrm{D}}\mathrm{i}^{[k]}$ maximal wird. Alternativ kann die Laufzeit deutlich erhöht werden, wenn $\sigma_t^{[k]} = c\sigma_i^{[k]}$ in einem konstanten Verhältnis c zueinander gesetzt werden.

5. Verfeinere die Harris-Ecke ${}^{\mathrm{D}}\mathbf{u}_{\perp}^{[k+1]}$ auf ${}^{\mathrm{D}}\mathrm{l}({}^{\mathrm{D}}\mathbf{u}, \sigma_i^{[k]}, \sigma_t^{[k]})$ durch Zuordnung des nächsten Maximum von m_{Harris} und projiziere sie zurück auf $\mathbf{u}_{\perp}^{[k+1]}$.

6. Berechne die neue Transformation $\mathbf{U}^{[k+1]} = ({}^{\mathrm{D}}\mathbf{G}_{\mathrm{l}}({}^{\mathrm{D}}\mathbf{u}_{\perp}^{[k+1]}, \sigma_i^{[k]}, \sigma_t^{[k]})^{-\frac{1}{2}})^{[k]}\mathbf{U}^{[k]}$ auf ${}^{\mathrm{D}}\mathrm{l}$) und normiere sie auf $\lambda_{\max}(\mathbf{U}^{[k+1]}) = 1$. Die Normalisierung ist von Nöten um ein geeignetes Sampeln der verzerrten Region zu ermöglichen. Sollte das Verhältnis $\frac{\lambda_{\max}}{\lambda_{\min}} > d = 6$ sein, wird die Iteration abgebrochen und keine Region konstruiert. Dies ist für wenige ungünstige Punkte der Fall an denen eine extreme Anisotropi herrscht und eine korrekte Konvergenz des Algorithmus unwahrscheinlich ist.

7. Springe zu 2. oder breche ab, falls die verzerrte Struktur $1 - q_A(\mathbf{G}_{\mathrm{l}}({}^{\mathrm{D}}\mathbf{u}_{\perp}^{[k+1]}, \sigma_i^{[k]}, \sigma_t^{[k]})) < \varepsilon = 0.05$ genügend isotrop ist.

Wie auch bei den HaS und HaSF konvergieren mehrere initiale Ankerpunkte zu denselben Regionen. Daher müssen Duplikate durch Vergleich ihrer Lage und Form eliminiert werden. Nach (MS04) werden ca. 40% der Ankerpunkte ausgeschlossen, da entweder die Anisotropie zu groß ist oder die lokale Skale σ_i nicht bestimmt werden kann. Von den verbleibenden Punkten bleiben ca. 30% nach Entfernen der Duplikate übrig, so dass ca. 20%-30% der initialen HaS bzw. HaSF zur Bestimmung der HaA genutzt werden können.

3.2.2.3 EBR: Edge-Based-Regions

Einen anderen Weg zur Regionenkonstruktion bestreiten die *Edge-Based-Regions* (EBR) bzw. *Geometry-Based-Regions* (TVG04). Ausgehend von Harris-Ecken werden mittels zwei naheliegender Kanten auf photometrisch invariante Weise (bzgl. photo$(\cdot)$, siehe Abschnitt 3.1.1) ein Parallelogramm erzeugt, das sich bzgl. des affinen geometrischen Modells $\mathbf{geo}_{\mathrm{A}}(\cdot)$ kovariant verhält. Während der Konstruktion wird dabei ausgenutzt, dass sich das Verhältnis zweier Flächeninhalte bei einer affinen Transformation nicht verändert. Zwei verschiedene Fälle sind zu unterscheiden: gekrümmte und gerade Kanten.

Konstruktion bei gekrümmten Kanten Sei $\mathbf{u}_{\perp}$ eine Harris-Ecke, die durch den Schnittpunkt zweier, durch den Canny-Edge-Detector (Can86) gefundener, gekrümmter Kanten definiert ist (vgl. Abb. 3.4). Seien $\mathbf{u}_1$, $\mathbf{u}_2$ weiterhin Punkte auf der jeweiligen Kante, so dass der Flächeninhalt $l_1(\mathbf{u}_1)$ und $l_1(\mathbf{u}_2)$:

$$l_i(\mathbf{u}_i) = \int |\det(\mathbf{u}_{[s_i],i}(s_i) \quad \mathbf{u}_{\perp} - \mathbf{u}_i(s_i))| ds_i \qquad i = 1,2$$

aller aufgespannter Parallelogramme des Tangens an die Kante $\mathbf{u}_{[s_i],i}(s_i)$ mit der Verbindung des Tangentialpunktes $\mathbf{u}_i(s_i)$ zu dem Ankerpunkt $\mathbf{u}_{\perp}$ gleich ist. s_i ist dabei ein Kurvenparameter zum Beschreiben der Kanten vom Ankerpunkt bis $\mathbf{u}_i$. Durch die Bedingung $l_1 = l_2 = l$ wird gewährleistet, dass das durch die Vektoren $\mathbf{u}_1(l) - \mathbf{u}_{\perp}$ und $\mathbf{u}_2(l) - \mathbf{u}_{\perp}$ aufgespannte Parallelogramm sich affin kovariant verhält. Dies gibt eine eindimensionale Familie Ω_l von parallelogrammförmigen Regionen,

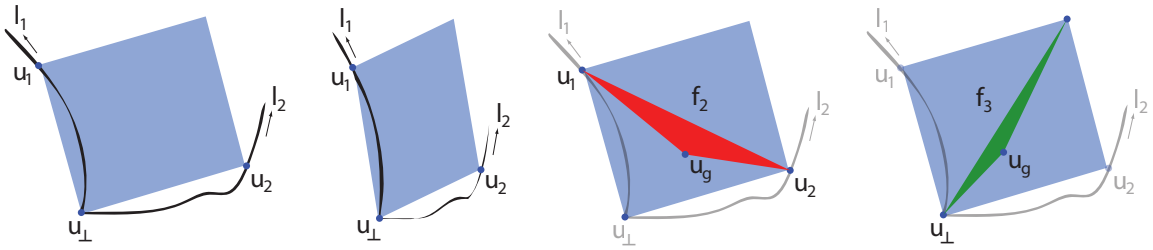

Abbildung 3.4: Konstruktionmethode der EBR, links die durch gleiche Flächeninhalte gekoppelten Eckpunkte $\mathbf{u}_1$ und $\mathbf{u}_2$ unter verschiedenen affinen Verzerungen, rechts die zwei Flächeninhalte der Funktionen f_2 und f_3. Adaptiert aus (TVG04).

für die als letzter Schritt der skalenähnliche Parameter l bestimmt werden muss. Dazu werden die Extrema von photometrisch invarianten Funktionen herangezogen:

$$\begin{aligned} f_1(\Omega) &= \frac{m^1_{00}}{m^0_{00}} \\ f_2(\Omega) &= c\left|\frac{\det(\mathbf{u}_1-\mathbf{u}_g \quad \mathbf{u}_2-\mathbf{u}_g)}{\det(\mathbf{u}_\perp-\mathbf{u}_1 \quad \mathbf{u}_\perp-\mathbf{u}_2)}\right| \\ f_3(\Omega) &= c\left|\frac{\det(\mathbf{u}_\perp-\mathbf{u}_g \quad \mathbf{u}_\perp-\mathbf{u}_g)}{\det(\mathbf{u}_\perp-\mathbf{u}_1 \quad \mathbf{u}_\perp-\mathbf{u}_2)}\right| \\ c &= \frac{m^1_{00}}{\sqrt{m^2_{00}m^0_{00}-(m^1_{00})^2}} \\ \mathbf{u}_g &= \left(\frac{m^1_{10}}{m^1_{00}},\frac{m^1_{01}}{m^1_{00}}\right)^{\mathrm{T}} \end{aligned}$$

die mittels der Momente $m^n_{pq}=\int\int_\Omega i^n(u,v)u^pv^q\mathrm{d}u\mathrm{d}v$ n-ten Ordnung und $(p+q)$-ten Grades definiert werden. f_1 repräsentiert die mittlere Helligkeit innerhalb Ω, ist sehr schnell zu berechnen, tendiert allerdings zu schlecht lokalisierbaren Extrema und erreicht auch nur seine Extrema auf photometrisch invariante Weise. f_1 bzw. f_2 beschreiben das Verhältnis zweier Flächen (siehe Abb. 3.4) die durch den Normierungsfaktor c ebenfalls photometrisch invariant berechnet werden können. Über die wesentlich robusteren Minima dieser Funktionen kann der verbleibende Faktor l bestimmt werden. Die Verwendung mehrerer Funktionen gewährleistet, dass für die meisten Ankerpunkte mind. eine Region stabil konstruiert werden kann.

Konstruktion bei geraden Kanten Da bei einer geraden Kante immer $l=0$ gilt, kann obige Konstruktionsmethode nicht verwendet werden. Man könnte stattdessen im zweidimensionalen Raum die Minima der Funktionen f_2 oder f_3 bestimmen. Diese neigen allerdings zu einer lang gezogenen Talbildung an den Minimastellen, so dass deren genaue Lage im zweidimensionalen nicht exakt lokalisierbarer ist. Es wird daher die Überlagerung beider Funktionen untersucht, der Schnitt beider Extremtäler bestimmt und damit die Lage von $\mathbf{u}_1$ bzw. $\mathbf{u}_2$. Sollten beide Täler ähnlich gerichtet sein, wird die Minimasuche ungenau und das Verfahren abgebrochen.

Optional kann an die parallelförmigen Regionen eine Ellipse angefittet werden. Dann muss allerdings auch eine charakteristische Orientierung geschätzt werden. Diese lässt sich aber leicht durch eine der beiden Parallelogrammdiagonalen in der normierten isotropen Darstellung (siehe. Abschnitt 3.2.1.4) angeben.

3.2.3 Konstruktion auf Basis der Hesse-Matrix

Im folgenden werden Verfahren beschrieben, die ihre Ankerpunkte auf Basis der Hesse-Matrix **H** bestimmen. Dazu zählen *Hessian-Similarity-Regions* (HeS), *Hessian-Affine-Regionen* (HeA) sowie aufbauend auf geschickten Approximationen von **H** *Scale-Invariant-Feature-Transform-Regions* (SIFT) und *Speeded-Up-Robust-Features* (SURF).

3.2.3.1 HeS: Hessian-Similar

Analog zu den HaS können auf exakt dieselbe Weise Hessian-Similarity-Regions (HeS) konstruiert werden ((Mik02)), indem anstatt von Harris-Ecken eine Auswertung der angepassten Hesse-Matrix $\mathbf{H}_\mathrm{l}(\mathbf{u},\sigma_t)$ zur räumlichen Maximumssuche herangezogen wird. Zur Wahl stehen die Determinante oder die Spur (LoG) der Matrix. Die Determinate hat gegenüber der Spur den Vorteil, dass sie kompakte gut lokalisierbare Strukturen bevorzugt. Dennoch existieren mit SIFT und SURF hochgradig optimierte, wesentlich schnellere Algorithmen, die allerdings eine approximierte Variante der Spur auswerten. Da man die HeS analog zu den HaS implementieren kann, wird dieser Abschnitt nicht weiter vertieft.

3.2.3.2 HeA: Hessian-Affine

Wie schon bei den HeS kann der Hessian-Affine (MTS$^+$05) Detektor (HeA) mit dem selben Algorithmus beschrieben werden wie die HaA's, indem anstatt von Harris-Ecken die Determinante der angepassten Hesse-Matrix $\mathbf{H}_\mathrm{l}(\mathbf{u},\sigma_t)$ zur räumlichen Maximumssuche herangezogen wird. Ausgangspunkt sind in diesem Fall natürlich HeS bzw. HeSF. Auf eine erneute Ausführung wird daher verzichtet und auf Abschnitt 3.2.2.2 verwiesen.

3.2.3.3 SIFT: Scale-Invariant-Feature-Transform

Der Detektor (Low04) der *Scale-Invariant-Feature-Transform* (SIFT) findet kovariante Regionen bzgl. des Ähnlichkeitsmodells. In seinem Kern besteht er aus einer approximierten Auswertung der Spur der Hesse-Matrix durch einen *Difference-of-Gaussians*-Ansatz (DoG) die sowohl für die räumliche Maximumssuche als auch die charakteristische Skaleselektion genutzt wird. Diese Herangehensweise kann auf effiziente Weise durch den Skalenraum realisiert werden. Als Nachverarbeitung werden die Maxima im Skalenraum durch Anfitten einer dreidimensionalen quadratischen Funktion subpixel-genau bestimmt, Punkte mit zu schlechtem Eigenkrümmungsverhältnissen (z.B. Kanten) eliminiert und eine Rotation geschätzt.

Mittels der Diffusions-Definition des Skalenraums $\mathrm{l}(\mathbf{u},\sigma)$ lässt sich zeigen, dass sich die Spur der angepassten Hesse-Matrix $\mathrm{tr}\,\mathbf{H}_\mathrm{l}$ (siehe Gl. 3.14) durch Differenzen des Bildes verschiedener Glättungsstufen realisieren lässt (Low04):

$$m_{DoG}(\mathbf{u},\sigma) = \mathrm{d}(\mathbf{u},\sigma) = l(\mathbf{u},\sigma) - l(\mathbf{u},k\sigma) \approx (k-1)\mathrm{tr}\,\mathbf{H}_\mathrm{l}$$

Der Faktor k beschreibt dabei den Glättungsunterschied, d. h. den Größenunterschied der beiden Gauß-Kernels. Bemerkenswert an der DoG-Approximation ist, dass der Normierungsfaktor σ^2 des 1-normierten Ableitungsoperators der angepassten Hessematrix (siehe Abschnitt 3.2.1.2) schon enthalten ist. Experimente zeigen, dass der zusätzliche Faktor $k-1$ selbst bei größeren Variationen im DoG-Raum $\mathrm{d}(\mathbf{u},\sigma)$ keinen praktischen Einfluss hat.

Die Maximasuche erfolgt auf gleiche Weise wie im Skalenraum. Da sowohl für die räumlichen Maxima als auch die Skalenselektion dieselbe Funktion m_{DoG} verwendet wird, spart man sich einen iterativen Ansatz wie bei den HaS. Ein Punkt gilt dann als Maximum im DoG-Raum, wenn er in der

$3\times3\times3$-Nachbarschaft von (u,v,σ) den größten Wert einnimmt. Für eine effiziente Implementierung wird nach jeder Oktave im Skalenraum, d.h. einer Verdopplung von σ das Bild im Skalenraum um die Hälfte unterabgetastet. Dies reduziert die Rechenzeit deutlich, hat aber auf das Ergebnis keinen Einfluss, da durch die Glättung das Abtasttheorem bereits erfüllt ist und es zu keinen Aliasing-Effekten kommt. Um eine Oktave mit s diskrete Stufen im DoG-Raum mit $k=2^{1/s}$ für die Maximussuche zu realisieren, sind deshalb $s+3$ (+1 für den Aufbau des DoG, +2 für die nachbarschaftlichen Skalen im DoG-Raum) diskrete Stufen des Skalenraums l notwendig. In (Low04) wird das Bild vor dem Aufbau des Skalenraums mit $\sigma=1.6$ geglättet, um die Maxima robuster detektieren zu können. Optional kann das Bild ebenfalls vor dem Aufbau auf die doppelte Größe expandiert werden, was zu ca. 4 mal mehr stabilen Maxima führt.

Für eine subpixel-genaue Bestimmung der gefundenen Maxima $\mathbf{x}_\perp=(\mathbf{u}_\perp,\sigma)^{\mathrm{T}}\in\mathcal{D}_\perp$ wird in einem Nachverarbeitungschritt an jedes Maximum eine drei-dimensionale quadratische Funktion angepasst. Ausgehend von der Taylor-Entwicklung des DoG-Raums um $\mathbf{x}_\perp$ bis zum quadratischen Term

$$\mathrm{d}(\mathbf{x}_\perp-\mathbf{x})=\mathrm{d}(\mathbf{x}_\perp)+\frac{\partial}{\partial\mathbf{x}}\mathrm{d}^{\mathrm{T}}(\mathbf{x}_\perp)\mathbf{x}+\frac{1}{2}\mathbf{x}^{\mathrm{T}}\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{d}(\mathbf{x}_\perp)\mathbf{x}$$

erhält man das subpixel-genaue Extremum durch Gleichsetzen der Ableitung der Entwicklung mit 0 und Lösen des erhaltenen Gleichungssystems:

$$\mathbf{x}=-\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{d}^{-1}(\mathbf{x}_\perp)\frac{\partial}{\partial\mathbf{x}}\mathrm{d}(\mathbf{x}_\perp)$$

Die Jacobi-Matrix $\frac{\partial}{\partial\mathbf{x}}\mathrm{d}$ und Hesse-Matrix $\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{d}$ werden dabei durch Pixel-Differenzen um $\mathbf{x}_\perp$ berechnet (BL02). Sollte der gefundene Offset $\mathbf{x}$ größer als 0.5 in einer Dimension sein, wird das Verfahren an dem näher gelegenen Sample-Punkt wiederholt. Der tatsächliche Wert des Extremums

$$\mathrm{d}(\mathbf{x}_\perp-\mathbf{x})=\mathrm{d}(\mathbf{x}_\perp)+\frac{1}{2}\frac{\partial}{\partial\mathbf{x}}\mathrm{d}^{\mathrm{T}}(\mathbf{x}_\perp)\mathbf{x}$$

kann herangezogen werden, um Maxima mit geringem Kontrast $|\mathrm{d}(\mathbf{x}_\perp-\mathbf{x})|<\varepsilon_{\mathrm{SIFT}}=0.03$[8] zu entfernen.

Da die Auswertung der Spur von $\mathbf{H}_\mathrm{l}$ auch auf sehr lange Blob-Strukturen mit ungünstigen Eigenkrümmungsverhältniss (z.B. Kanten) reagiert, müssen diese Maxima ebenfalls eliminiert werden. Dazu wird die Hesse-Matrix $\mathbf{H}$ an der verfeinerten Stelle $\mathbf{x}_\perp+\mathbf{x}$ im DoG-Raum berechnet und an Anlehnung an die Harris-Auswertung (HS88a) ohne explizite Bestimmung des Eigensystems das Verhältnis $r=\lambda_{\max}/\lambda_{min}$ der Eigenwerte untersucht:

$$\frac{\mathrm{tr}^2\mathbf{H}(\mathbf{x}_\perp+\mathbf{x})}{\det\mathbf{H}(\mathbf{x}_\perp+\mathbf{x})}=\frac{(r+1)^2}{r}<\frac{(\tau+1)^2}{\tau}$$

Sollte die Bedingung mit $\tau=10$ erfüllt sein oder $\det\mathbf{H}(\mathbf{x}_\perp+\mathbf{x})<0$ wird das Maximum als zu ungünstig zurückgewiesen.

Als letzten Schritt wird für die übrig gebliebenen Maxima mittels des in Abschnitt 3.2.1.4 beschriebenen Gradienten-Orientierungs-Histogramms die Rotation bestimmt.

3.2.3.4 SURF: Speeded-Up-Robust-Features

Der Detektor (BTVG06) der *Speeded-Up-Robust-Features* (SURF) motiviert sich aus einem Trade-Off zwischen Geschwindigkeit und Präzision der Regionen. Er orientiert sich an SIFT und bestimmt

[8] bei Pixelwerten zwischen 0 und 1

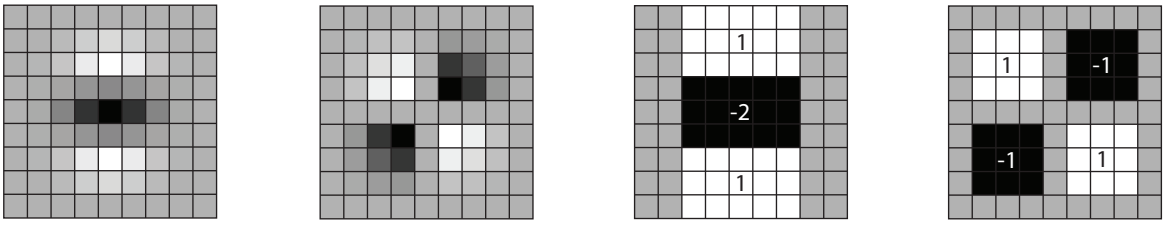

Abbildung 3.5: Diskrete Varianten (links) der gaußschen Ableitung zweiter Ordnung $l_{[uu]}$ bzw. $l_{[uv]}$ sowie approximierte Varianten (rechts) mit Box-Filtern $d_{[uu]}$ bzw. $d_{[uv]}$. Die Box-Filter werden anschließend über ihre Masken-Größe normiert. Aus (BTVG06).

ebenfalls eine gerichtete kreisförmige Region. Dazu wertet er sowohl für die räumlichen Maxima als auch für die Skalen-Selektion eine approximierte Variante der Determinate $\det \mathbf{H}_l(\mathbf{u},\sigma)$ der Hessematrix auf dem Skalenraum aus. Diese wird sehr grob mit Box-Filtern (siehe Fig. 3.5) nachgebildet, die im Gegensatz dafür mittels Integralbildern $i_\Sigma(u,v) = \sum_{j=0}^{j\leq u}\sum_{k=0}^{k\leq v} i(j,k)$ (für Details siehe (VJ01)) in beliebiger Größe in konstanter Zeit berechnet werden können. Die Determinante approximiert sich dann zu

$$\det \mathbf{H}_l(\cdot) = l_{[uu]}(\cdot)l_{[vv]}(\cdot) - l^2_{[uv]}(\cdot) \approx d_{[uu]}(\cdot)d_{[vv]}(\cdot) - c d^2_{[uv]}(\cdot)$$

wobei der Faktor $c = \frac{|l_{[uu]}(\cdot,1.2)|_F |d_{[uu](\cdot,9)}|_F}{|l_{[uu]}(\cdot,1.2)|_F |d_{[uu]}(\cdot,9)|_F} = 0.912\ldots \approx 0.9$ das relative Gewicht in dem Ausdruck zur Berechnung der Determinante erhält. Er wird mittels der Frobenius-Norm $|\cdot|_F$ auf der kleinsten Filter-Skala $\sigma_0 = 1.2$ bzw. der dazu korrespondierenden Box-Filtergröße 9×9 berechnet. Bei einem Wechsel der Skala $\sigma = s\sigma_0 = 1.2s$ der Gauß-Kernels im kontinuierlichen Fall skalieren die entsprechenden Box-Filter im approximierten Fall genauso und werden dann mittels der Masken-Größe $9s\times 9s$ repräsentiert. Ähnlich wie bei der SIFT-Auswertung wird in jeder neuen Oktave der Skale die Filtergröße sowie die räumliche Abtastung bei der Maximasuche verdoppelt. Die Maxima-Suche erfolgt dann wie bei SIFT durch Auswertung der $3\times 3\times 3$-Nachbarschaft im Box-Filter approximierten Skalenraum. Da man die Determinante auswertet, müssen die gefundenen Punkte nicht mehr auf ungünstige Eigenkrümmungsverhältnisse überprüft werden. Wie bei SIFT wird nach dem Verfahren von (BL02) (siehe Abschnitt 3.2.3.3) abschließend eine subpixel-genaue Bestimmung der Maxima vorgenommen.

Mittels eines ähnlichen Vorgehens wie das in Abschnitt 3.2.1.4 beschriebene Gradienten-Orientierungs-Histogramm wird dann eine Rotation bestimmt. Diese wurde allerdings so modifiziert, dass anstatt von Gradienten Antworten von Haar-Wavelets zur Richtungsbestimmung benutzt werden. Diese können wiederum effizient mit Integralbildern berechnet werden. Anstatt einer Histogramm-Auswertung wird das Maximum eines rotierenden Orientierungs-Fensters mit Öffnung $\frac{1}{3}\pi$ als charakteristische Richtung herangezogen.

3.2.4 Konstruktion auf Basis von Intensitätsextrema

Regionen können auch auf Basis von Ankerpunkten konstruiert werden, die sich aus Intensitätsextrema ableiten. Dazu gehören die *Maximal-Stable-Extremal-Regions* und die *Intensity-Based-Regions*.

3.2.4.1 MSER: Maximal-Stable-Extremal-Regions

Maximal-Stable-Extremal-Regions (MSER) werden in (MCUP04) vorgestellt und nutzen Intensitätsextrema als Startwerte zur Detektion von lokal binearisierten *Extremal-Regions* (ER), aus deren sich

die Teilmenge der robusten MSER ableiten lässt. MSER haben eine Reihe erwünschter Eigenschaften, wie eine invariante Konstruktion bzgl. kontinuierlicher, d.h. auch linearer Intensitätsänderungen (vgl. photometrisches Modell in Abschnitt 3.1.1), eine kovariante Konstruktion bzgl. jeglicher kontinuierlichen geometrischen Transformation, d.h. auch bzgl. $\mathbf{geo}_{\mathbf{A}}(\cdot)$, einer impliziten Multi-Skalen-Detektion, einer hohen Robustheit durch Selektion von ER's, die sich nur minimal bei Parametervariation verändern sowie eines sehr effizienten Algorithmus.

Ausgangspunkt ist ein diskretes Bild $\mathrm{i} : \mathcal{D} \subseteq \mathbb{Z}^2 \rightarrow \mathcal{W}$ mit einer (im Allgemeinen beliebigen) Menge $\mathcal{W}$ von Intensitäten, auf denen eine totale Ordnung definiert ist. Weiterhin muss eine Nachbarschaftsbeziehung $a : \mathcal{D}^2 \rightarrow \{0,1\}$ definiert sein, für die $a(\mathbf{u},\mathbf{v}) = 1$ nur gilt, falls die beiden Punkte $\mathbf{u},\mathbf{v} \in \mathcal{D}$ Nachbarn sind. In (MCUP04) ist $\mathcal{W} = \{0,1,\ldots,255\}$ und $a(\mathbf{u},\mathbf{v}) = 1$ wenn $(\mathbf{u}-\mathbf{v})^{\mathrm{T}}(\mathbf{u}-\mathbf{v}) \leq 1$ gilt (2×2-Nachbarschaft).

In einem ersten Schritt werden alle Pixel in $\mathbf{u} \in \mathcal{D}$ nach ihrem Intensitätswert $\mathrm{i}(\mathbf{u})$ sortiert. Für diskrete $\mathcal{W}$ kann dies mittels Binsort (Sed88) sehr effizient in $O(n)$, $n = |\mathcal{D}|$ erfolgen. Danach werden die Pixel in auf- (-) oder absteigender (+) Reihenfolge der Intensitätswerte in einem anfangs leerem Bild plaziert, was einer Binearisierung mit größer oder kleiner werdenden Schwellwerten $\tau_1,\ldots,\tau_i$ entspricht (vgl. Abb. 3.6). Mittels des Algorithmus zur Vereinigungs-Suche (Sed88) kann in $O(n \log\log n)$ die Menge aller zusammenhängender Regionen $\mathcal{Q} \subset \mathcal{D} = \{\mathbf{u},\mathbf{v} \in \mathcal{Q} \Rightarrow \exists \mathbf{p}_1,\ldots,\mathbf{p}_i \in \mathcal{Q} : a(\mathbf{u},\mathbf{p}_1) = a(\mathbf{p}_j,\mathbf{p}_{j+1}) = a(\mathbf{p}_i,\mathbf{v}) = 1\}$ gefunden werden die extremal sind[9]:

$$\mathcal{Q}^{\pm}_{\mathrm{ER}} \subset \mathcal{Q} = \{\mathbf{u} \mid \forall \mathbf{v} \in \partial\mathcal{Q}_{\mathrm{ER}} : \mathrm{i}(\mathbf{u}) >^{+} \mathrm{i}(\mathbf{v}) \,\mathrm{oder}\, \mathrm{i}(\mathbf{u}) <^{-} \mathrm{i}(\mathbf{v})\},$$

definiert mittels der Regionengrenze $\partial\mathcal{Q} = \{\mathbf{u} \in \mathcal{D}\backslash\mathcal{Q} : \exists \mathbf{v} \in \mathcal{Q} : a(\mathbf{u},\mathbf{v}) = 1\}$. Die Menge aller ER's bilden abhängig der Schwellwerte τ_i eine verschachtelte Struktur $\ldots, \mathcal{Q}_{\tau_{i-1}} \subseteq \mathcal{Q}_{\tau_i} \subseteq \mathcal{Q}_{\tau_{i+1}}, \ldots$ aus denen diejenigen Regionen $\mathcal{Q}_{\mathrm{MSER}} = \mathcal{Q}_{\tau^*}$ als maximal stabil angenommen werden, deren relative Größenänderung $q(i)_{\mathrm{MSER}} = |\mathcal{Q}_{\tau_{i+\Delta}} \backslash \mathcal{Q}_{\tau_{i-\Delta}}|/|\mathcal{Q}_{\tau_i}|$ bei einer Variation des Schwellwertes τ_i ein lokales Minimum τ^* hat. $\Delta = \Delta_{\mathrm{MSER}}$ ist dabei ein Parameter des Algorithmus der die geforderte Stabilität beeinflusst. Die gefundenen MSER sind noch von beliebiger Form. Es kann daher als letzten Schritt eine Ellipse mit den Parametern $\mathbf{u}_{\perp} = |\mathcal{Q}|^{-1} \sum_{\mathbf{v} \in \mathcal{Q}} \mathbf{v}$ und $\mathbf{A} = |\mathcal{Q}|^{-1} \sum_{\mathbf{v} \in \mathcal{Q}} (\mathbf{v}-\mathbf{u}_{\perp})^{\mathrm{T}}(\mathbf{v}-\mathbf{u}_{\perp})$ angefittet werden. Ebenso muss mittels der in Abschnitt 3.2.1.4 beschriebenen Verfahren eine Orientierung geschätzt werden. Alternativ kann auch der Punkt der maximalen Krümmung der Regionengrenzen herangezogen werden.

Abhängig von der Sortierreihenfolge findet der Algorithmus entweder MSER^{+} mit Intensitätsmaxima oder MSER^{-} mit Intensitätsminima als Startregionen. Um nicht beide Fälle des Algorithmus implementieren zu müssen, wird meist in einem zweiten Durchlauf das invertierte Bild übergeben.

3.2.4.2 IBR: Intensity-Based-Regions

Die *Intensity-Based-Regions* (IBR) werden in (TVG04) vorgestellt und konstruieren auf photometrisch (vgl. Abschnitt 3.1.1) invariante Weise eine affin kovariante Region durch direkte Auswertung der Bildintensitäten. Ausgehend von einem Intensitätsextrema $\mathbf{u}_{\perp}$ wird für sternenförmig abgehende Strahlen $\mathbf{u}(t)$ eine Funktion

$$f(t) = \frac{|\mathrm{i}(\mathbf{u}(t)) - \mathrm{i}(\mathbf{u}_{\perp})|}{\max\left(\int_0^t |\mathrm{i}(\mathbf{u}(t)) - \mathrm{i}(\mathbf{u}_{\perp})dt|, \varepsilon\right)}$$

untersucht (vgl. Abb. 3.7). t ist dabei ein Geraden- bzw. Strahlenparameter mit $\mathbf{u}(0) = \mathbf{u}_{\perp}$ und ε eine kleine Zahl die eine Division durch null verhindern soll. $f(t)$ ist bezogen auf den Strahl invariant bzgl. $\mathrm{photo}(\cdot)$ und $\mathbf{geo}_{\mathrm{A}}(\cdot)$, aus Gründen der Robustheit werden dennoch Punkte $\mathbf{u}(t_e)$ gewählt, an

[9] In (NS08) wird ein modifizierter Algorithmus vorgestellt, der eine linearer Zeit $O(n)$ benötigt und durch ein besseres Speichermanagement auch praktisch wesentlich effizienter ist.

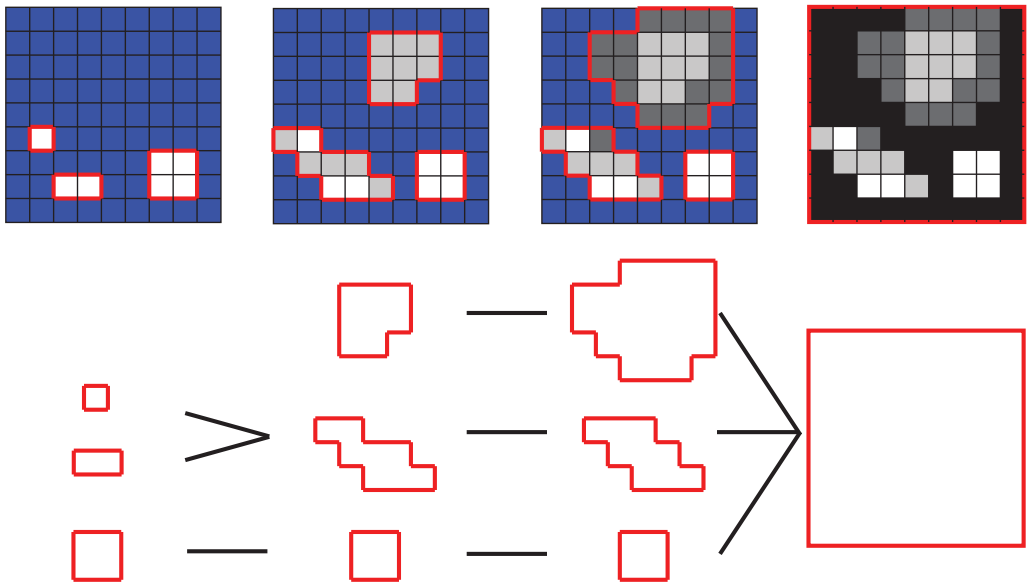

Abbildung 3.6: Konstruktionsmethode der MSER. Die obere Reihe zeigt die entstandenen extremalen Regionen Q_{ER}^{+} nach der Schwellwertoperation, wobei der Schwellwert von links nach recht von hell nach dunkel absteigt (+). Ganz rechts ist der Algorithmus am niedrigsten Schwellwert angelangt und segmentiert das gesamte Ausgangsbild. Darunter ist die verschachtelte Struktur aller ER's abgebildet, die von links nach rechts eine Baumstruktur von den Blättern bis zur Wurzel bildet. Es werden diejenigen ER's als maximal stabil ausgewählt, die entlang ihres Pfades von Blatt zu Wurzel in einer Umgebung $\pm\Delta$ eine minimale Größenänderung haben. In obigem Beispiel trifft dies auf die untere quadratische Region in idealer Weise zu.

denen $f(t)$ zusätzlich ein lokales Extremum erreicht. Typischerweise sind dies Punkte, an denen die Intensität sich dramatisch verändert bzgl. der durchschnittlichen Intensitätsänderung entlang des Strahls.

Die Verbindung all dieser Punkte ergibt eine affin kovariante Region. Sollten mehrere Extrema entlang eines Strahls gefunden werden, wird dasjenige ausgewählt, dessen Entfernung zu den benachbarten Maxima am kleinsten ist. An die gefundene Region wird anschließend eine Ellipse angefittet und deren Größe für eine bessere Eindeutigkeit der Region verdoppelt. Optional kann noch mittels der in Abschnitt 3.2.1.4 beschriebenen Verfahren eine Rotation geschätzt werden. Alternativ kann man auch den Hauptkrümmungspunkt der Regionengrenze dazu heranziehen.

3.2.5 Konstruktion auf Basis von Entropiemaxima

Bis jetzt wurden Verfahren beschrieben, die die tatsächliche geometrische Struktur von Mustern ausgewertet haben. Bei der Konstruktion der Regionen durch Auswertung von Entropieeigenschaften wird dagegen nur die lokale Grauwertverteilung von Mustern betrachtet. In diese Klasse fallen die *Salient-Similarity-Features* (SaS) und *Salient-Affine-Features* (SaA).

3.2.5.1 SaS: Salient-Similarity-Features

Wie bereits in Abschnitt 3.2.1.1 erwähnt, lassen sich saliente Strukturen in Bildern nach (Gil98) mittels der Shannon-Entropie über eine lokale Region definieren. Dies wird in (KB01) ausgenutzt um über gewichtete Maxima der Entropy in (KB01) die *Salient-Similarity-Features* (SaS) zu konstruierten. Für eine automatische Skalenselektion wird die kreisförmige Region $\mathcal{R}[\mathbf{u}, s\mathbf{I}_{2\times 2}]$ sowohl über

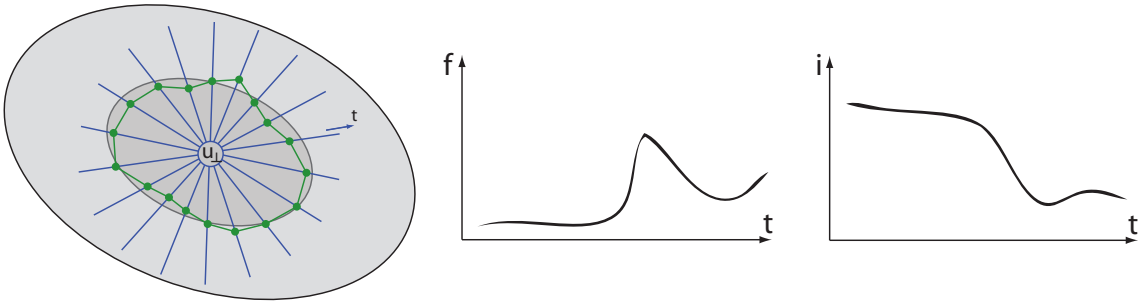

Abbildung 3.7: Konstruktionsmethode der IBR (links) und Gegenüberstellung (rechts) von Intensitätsverlauf $\mathrm{i}(\mathbf{u}(t))$ und $f(t)$ entlang eines Strahls.

den Bildbereich $\mathbf{u}$ als auch die Skale s variiert. Die lokale Entropie ergibt sich dann zu

$$\rho(\mathbf{u},s) = -\sum_{d\in\mathcal{W}} p_{\mathcal{R}(\mathbf{u},s\mathbf{I})}(d)\mathrm{ld}\, p_{\mathcal{R}(\mathbf{u},s\mathbf{I})}(d)$$

wobei die Wahrscheinlichkeitsverteilung $P(d\in\mathcal{W}) = P(\mathrm{i}=d) = |\mathcal{R}|^{-1}H(\mathrm{i}(\mathbf{v})=d, \mathbf{v}\in\mathcal{R})$ über die (hier diskrete) Wertemenge $\mathcal{W}$ des Bildes i durch das normierte lokale Grauwerthistogram H über die Region $\mathcal{R}$ geschätzt wird. $p(d)$ ergibt sich daher mittels des Kronecker-Symbols δ zu $h(d) = |\mathcal{R}|^{-1}\sum_{\mathbf{v}\in\mathcal{R}}\delta_{d,\mathrm{i}(\mathbf{v})}$. Diese Definition für Salienz ist allerdings nicht ausreichend. Betrachtet man z.B. einen Busch, so hat dieser durch seine chaotische Blätterstruktrur eine hohe Entropie, ist allerdings durch seine Selbstähnlichkeit innerhalb veschiedener Skalenstufen nicht als salient zu bezeichnen. (KB01) führt daher eine modifizierte Salienzdefinition $\gamma(\mathbf{u},s) = w(\mathbf{u},s)\rho(\mathbf{u},s)$ ein, deren Gewichtsfunktion

$$w(\mathbf{u},s) = s\sum_{d\in\mathcal{D}}\left|\frac{\partial}{\partial s}p_{\mathcal{R}[\mathbf{u},s\mathbf{I}]}(d)\right|$$

die Selbstunähnlichkeit über die Skale beschreibt. Die Maxima von $\gamma(\mathbf{u},s)$ werden auf rotations- und photometrisch invariante Weise gefunden, neigen allerdings zu starker Clusterbildung. Daher werden nur die salientesten Regionen eines Clusters ausgegeben. Um sie vollständig kovariant bzgl. des geometrischen Transformationsmodells $\mathbf{geo}_S(\cdot)$ zu machen, muss mit einem Verfahren aus Abschnitt 3.2.1.4 noch eine charakteristische Orientierung geschätzt werden.

3.2.5.2 SaA: Salient-Affine-Features

In (KZB04) erweitern die Autoren die Idee der SaA's auf affin kovariante *Salient-Affine-Features* (SaA). Der Suchraum für Maxima der modifizierten Salienzmetrik $\gamma(\cdot) = w(\cdot)\mathrm{ent}(\cdot)$ wird auf einen affinen Raum mit Parametervektor $(\mathbf{u},s,\rho,\theta)$ (ohne Rotation φ in der normalisierten Darstellung, vgl. Abschnitt 3.2.1 bzw. 3.2.1.4) erweitert. Die elliptische Region $\mathcal{R}[\mathbf{u},\mathbf{A}]$ wird dabei durch die anisotrope Skalierungsmatrix $\mathbf{A} = \mathbf{R}_\theta\mathrm{diag}(s,\rho s)\mathbf{R}_{-\theta}$ definiert. Bei der affinen Erweiterung treten allerdings zwei Probleme auf. Bei der Berechnung der Gewichtungsfunktion $w(\cdot)$ mittels Histogrammdifferenzen treten Aliasing-Effekte auf, deren Scheinstrukturen eine Maximumssuche erschweren. Für eine robuste Berechnung wird daher das Histogramm über eine Fensterfunktion h_w abhängig vom Radius r des Pixels in der normalisierten kreisförmigen Region gewichtet. Empirisch wurde in (KZB04) $h_w(r) = \frac{1}{1+r^{42}}$ gewählt. Weiterhin kann bei der Maximumssuche ein degenerativer Fall auftreten, an denen die Entropie $\rho(\mathbf{u},s,\rho,\theta)$ über mehrere Skalen s durch eine gleichbleibende Grauwertverteilung unverändert bleibt und daher $w(\cdot)$ sehr klein wird. Dies kann umgangen werden, wenn die Selbstun-

ähnlichkeit w über mehrere Skalen geglättet betrachtet wird. Realisiert wird dies über einen einfachen Durchschnittsfilter von $w(\mathbf{u}, s, \rho, \theta)$ über s.

Da eine vollständige Suche im affinen Raum zu aufwändig ist, wird eine lokale iterative Suchstrategie ähnlich zu den HaA's bzw. HeA's angewendet. Ausgehend von SaS Startpunkten wird iterativ abwechselnd zuerst über die anisotropen Parameter ρ und θ die Selbstunähnlichkeit $w(\mathbf{u}, s, \rho, \theta)$ maximiert, dann über die Skale s die Entropie $\rho(\mathbf{u}, s, \rho, \theta)$ maximiert. Abgebrochen wird nach einer festen Anzahl von Iterationsschritten oder wenn sich weder Skale noch Form weiter verändern. Wie auch bei den SaS's werden nur die besten Regionen eines Clusters ausgewählt und zusätzlich für eine vollständige kovariante Konstruktion eine charakteristische Orientierung geschätzt.

3.2.6 Bewertung der Detektoren

Dieser Abschnitt soll alle oben vorgestellten Verfahren bewerten, ihre Stärken, Schwächen und Eigenarten darstellen und zusammenfassend eine Einschätzung des Autors bzgl. eines praktischen Einsatzes geben. Die dafür notwendigen Daten wurden allesamt aus der Literatur entnommen (MS02; MS04; MTS$^+$05; BTVG06) und werden an entsprechender Stelle gekennzeichnet. Leider ist es nicht ganz einfach, Detektoren getrennt von Deskriptoren zu evaluieren, da bei vielen Arbeiten, wie z.B. (VL07) nur eine kombinierte Auswertung erfolgt, insbesondere wenn es sich um spezielle Algorithmen wie SIFT oder SURF handelt. Ebenso wird meist nur eine Untermenge der hier vorgestellten Detektoren untersucht, so dass es sehr schwierig ist einen umfassenden Vergleich anzustellen. Bei den zusammenfassenden Bewertungen werden daher fehlende Informationen durch Einschätzungen des Autors ergänzt und dementsprechend markiert.

Die wichtigsten Bewertungskriterien sind die *Reproduzierbarkeit* und *Genauigkeit* unter verschiedensten geometrischen und photometrischen Bildtransformationen, die *Eindeutigkeit* der gefundenen Regionen sowie die *Laufzeit*. Diese werden in den folgenden Paragraphen im Detail behandelt, hängen aber stark von den verwendeten Testdatensätzen ab. Eine Verallgemeinerung kann daher nur mit Einschränkungen gemacht werden. Weiterhin sind auch noch die Anzahl der Regionen und die Regionengröße von Bedeutung, auf diese wird aber an geeigneter Stelle hingewiesen.

Laufzeit Die Rechengeschwindigkeit verschiedener Detektoren ist abhängig von dem verwendeten Rechner, der Bildgröße, des Szeneninhalts, der verwendeten Parameter und der Anzahl gefundener Regionen. Eine abstrahierte Bewertung in Tab. 3.2 gibt einerseits die theoretischen Grenzen, andererseits aber auch eine Einschätzung für den praktischen Einsatz an. Damit sollte es möglich sein, einen Eindruck über die Größenordnung der Verarbeitungszeiten zu bekommen, sowie einen Vergleich zwischen den einzelnen Detektoren anzustellen. Weiterhin sind konkrete Rechenzeiten aus (MS04; MTS$^+$05; BTVG06) angegeben.

Klar zu erkennen ist, dass die ähnlichen Detektoren SIFT, SURF, HeA und HeS aufgrund der geringeren Komplexitätsklasse der Regionen wesentlich schneller sind als ihre affin-kovarianten Verwandten. Einzige positive Ausnahme sind die MSER, die sich stark von den anderen affinen Verfahren abheben und nur von den auf der approximierten Hesse-Matrix basierenden Verfahren SIFT und SURF geschlagen werden. Aufgrund der hohen Berechnungszeit ist es fraglich ob die Salienten-Regionen als auch die EBR in mittlerer Zukunft jemals praktisch nutzbar werden.

Anzumerken ist, das auch bei modernen Rechnern (Stand: Core 2 Duo 3 Ghz) mit diversen SIMD-Erweiterungen wie SSE und mehreren Cores eine Bearbeitungszeit zwischen 100 ms - 1000 ms auf der CPU anzusetzen sind. Um eine Größenordnung schnellere Zeiten sind nur auf der GPU zu erreichen, wobei dies für einzelne Verfahren schon umgesetzt wurde (Wu09).

Reproduzierbarkeit Die Reproduzierbarkeit ist ein Maß für die Robustheit der einzelnen Verfahren bzgl. einer Menge von Bildtransformationen. Sie ist definiert als die Anzahl der Regionen, deren

Algo.	**geo**	$\mathcal{M}$	Laufzeit		(MS04)		(MTS$^+$05)		(BTVG06)	
			theo.	prakt.	t [s]	#	t [s]	#	t [s]	#
HaS	S	C	$O(n+(s+k)p)$	+	7.00	1438			1.80	1664
HaSF	S	C	$O(n+sp)$	+	1.40	1624				
HeS	S	B	$O(n+(s+k)p)$	+					0.65	1979
HeSF	S	B	$O(n+sp)$	++						
SIFT	S	B	$O(n+sp)$	++	0.70	1527			0.40	1520
SURF	S	B	$O(n+sp)$	++					0.12	1418
SaS	S	E	$O(sn+p)$	-						
HaA	A	C	$O(n+(s+k)p)$	o+	36.0	1123	1.43	1791		
EBR	A	C*	$O(n+dp)$	--			164	1265		
HeA	A	B	$O(n+(s+k)p)$	o			2.73	1649		
MSER	A	F	$O(n+n\log\log n)$	+			0.66	533		
IBR	A	F	$O(n+p)$	-			10.8	679		
SaA	A	E	$O(sn+klp)$	--			2013	513		
Daten	Bildgröße in allen Fällen 800×640				P2 0.5 GHz		P4 2 GHz		P4 3 GHz	

Tabelle 3.2: Vergleich der Laufzeit verschiedener Detektoren. Neben der Kovarianz-Klasse bzgl. dem geometrischen Modell **geo**: affin (A) oder ähnlich (S) und der Art des zugrundeliegenden Musters $\mathcal{M}$: Ecken (C), Ecken u. Kanten (C*), Blobs (B), homogene Flächen (F) sowie hohe Entropie (E) ist die theoretische Performanceklasse sowie eine persönliche praktische Einschätzung angegeben. n ist dabei die Anzahl der Bildpixel, p die Anzahl der gefundenen Ankerpunkte, s die Anzahl untersuchter Skalen, k die Anzahl Iterationsschritte, p die mittlere Anzahl an Kanten in der Nähe einer Ecke sowie l die Anzahl untersuchter anisotroper Skalenvariationen. Rechts daneben sind verschiedene konkrete Zeiten t aus der Literatur angegeben, sowie die Anzahl # der dabei konstruierten Regionen. In (MS04) wurde dafür ein P2 0.5 GHz verwendet, in (MTS$^+$05) ein P4 2 GHz und in (BTVG06) ein P4 3 GHz. Die Bildgröße war in allen Fällen 800×640.

Überlappungsfehler (durch Projektion der Region in das jeweils andere Bild) zwischen zwei Bildern mit bekanntem Transformationsunterschied kleiner als 40% ist. Diese Regionen eignen sich für potentielle Korrespondenzen, da die physikalischen Bildinhalte großteils die gleichen sind. Allerdings hängt dies noch von dem Leistungsvermögen des Deskriptors ab, sowie von der Eindeutigkeit der gefundenen Regionen. Die Reproduzierbarkeit kann absolut als Anzahl an (potentiellen) Korrespondenzen als auch relativ bzgl. der kleineren Anzahl an gefundenen Regionen innerhalb beider Bilder angegeben werden.

In (MTS$^+$05) wird ein Testdatensatz vorgestellt, der Bilderserien mit verschiedenen Szeneninhalt und unterschiedlichen Transformationen enthält. Er umfasst strukturierte und texturierte Szenen mit Blickwinkeländerungen (Rotationen aus der Bildebene hinaus), Skalierungsänderungen, Schärfeänderungen, Beleuchtungsschwankungen sowie verschiedene Stufen der JPG-Kompression. In Abb. 3.8 werden eine für diese Arbeit relevante Untermenge der Ergebnisse zur Reproduzierbarkeit für ähnlich-kovariante Regionen (BTVG06) und affin-kovariante Regionen (MTS$^+$05) sowie Auschnitte der Bilderserien dargestellt. Die Untersuchungen in (BTVG06) beschränken sich nur auf relative Daten. Ideal wäre eine relative Reproduzierbarkeit von 100% mit einer Linie parallel zur Z-Achse. Ebenso sollte die absolute Anzahl der möglichen Korrespondenzen konstant bleiben. Wie sich leicht erkennen lässt, ist dies aus unterschiedlichen Gründen für keinen untersuchten Detektor gegeben. Zum einen wandern manche Regionen mit zunehmender Transformation aus der Szene hinaus oder sie können nicht mehr aufgelöst werden, wie es z.B. bei einer Skalierungsänderung mit sehr kleinen Regionen der Fall ist. Andererseits spiegelt die Reproduzierbarkeit die Güte der zugrundeliegenden

Modelle der verschiedenen Detektoren wieder, sowie die Robustheit der Algorithmen gegenüber bestimmten Transformationen. Ebenso ist die Reproduzierbarkeit stark abhängig vom Szeneninhalt, insbesondere die Unterscheidung zwischen strukturierten und texturierten Mustern. Diese Zusammenfassung konzentriert sich auf strukturierte Inhalte, da sie als repräsentativer für die zu lösende Aufgabe angesehen werden. Allgemein kann man aber sagen, dass sich die relative und absolute Reproduzierbarkeit der MSER für texurierte Muster erhöht und für EBR sinkt (für Details siehe (MTS$^+$05; BTVG06)). Da die Ergebnisse für die einzelnen Sequenzen stark schwanken, sollen sie hier im Detail diskutiert werden.

Bei einer *Änderung des Blickwinkels*, d.h. einer Rotation aus der Bildebene hinaus erkennt man sehr gut die Auswirkung der unterschiedlichen zugrundeliegenden geometrischen Modelle der ähnlich- bzw. affin-kovarianten Detektoren. Während bei kleinen Änderungen $\leq 20^\circ$ die S-Detektoren aufgrund ihrer Robustheit noch eine hohe Reproduzierbarkeit liefern, brechen sie bei größeren Transformationen sehr schnell ein und funktionieren ab 50° überhaupt nicht mehr. Dies ist auch nicht weiter verwunderlich, da mit $\mathbf{geo}_S(\cdot)$ keine affine Verformungen der kreisförmigen Regionen modelliert werden kann, wie sie bei einer Änderung des Blickwinkels entstehen. Alle ähnlichen Detektoren schneiden vergleichbar schlecht ab. SIFT hat einen leichten Vorteil, SURF dagegen einen leichten Nachteil. Auch die affin-kovarianten Detektoren brechen mit zunehmender Blickwinkeländerung ein, allerdings bei weitem nicht so schnell, so dass selbst bei 60° noch Korrespondenzen gefunden werden können. Allerdings sinkt die absolute Reproduzierbarkeit rapide ab. MSER dominiert diesen Vergleich deutlich vor den vergleichbaren HeA und IBR. Die SaA schneiden selbst bei geringen Winkeländerungen sehr schlecht ab.

Ein gegenteiliges Bild zeichnet sich bei *Skalierungs- und Rotationsänderungen* in der Bildebene ab. Dort performieren die ähnlichen Verfahren vor allem bei größeren Transformationen besser, da sie aufgrund ihrer geringen Komplexität robuster sind. Bis auf kleine Nachteile der HaS schneiden diese Detektoren alle vergleichbar gut ab. Die affinen Verfahren brechen insbesondere bei großer Skalierungsänderung ein, wobei die HeA vor den MSER und HaA liegen. Die IBR und weit abgeschlagen die SaA bilden das Schlusslicht bei diesem Vergleich. Die absolute Anzahl der Korrespondenzen bricht sehr schnell ein, insbesondere bei den Detektoren, die sehr viele kleine Regionen in dem Referenzbild finden.

Für die *Sequenz mit zunehmender Unschärfe* liegen nur Daten für die affinen Detekoren vor, die Ergebnisse sollten sich aber von den HaA bzw. HeA auf die ähnlichen Verwandten HaS bzw. HeS übertragen lassen. Bei geringer Schärfeänderung funktionieren sowohl die HeA als auch MSER am besten, allerdings brechen letztere sehr schnell ein, was wahrscheinlich auf Schwierigkeiten des Verfahrens bei weichen Regionenübergängen zurückzuführen ist. Weniger Probleme bei großer Unschärfe haben dagegen die HeA aber auch die EBR. IBR und SaA haben eine schlechte relative Reproduzierbarkeit, bei der absoluten Anzahl an Korrespondenzen sind sie allerdings wie alle anderen Algorithmen nahezu konstant. Dies ist darauf zurückzuführen, dass es sich um eine statische Szene handelt, an der nur der Fokus verändert wurde und somit keinerlei Regionen aus dem Bild herauswandern bzw. nicht mehr aufgelöst werden können.

Die gleiche Argumentation gilt auch für die *Beleuchtungssequenz* bei der nur die Blende verändert wurde. Auch hier sinkt im Vergleich zu einer Änderung des Blickwinkels bzw. der Skalierung die absolute Anzahl der Korrespondenzen nur gering. Die S-Detektoren haben eine vergleichbare Performance wie ihre affinen Verwandten. Wie auch schon bei den anderen Transformationen scheiden die Hesse-Matrix basierten Verfahren etwas besser ab als die Strukturtensor basierten Detektoren. Dominiert wird dieser Vergleich aber von den MSER, die aufgrund ihres allgemeineren Beleuchtungsmo-

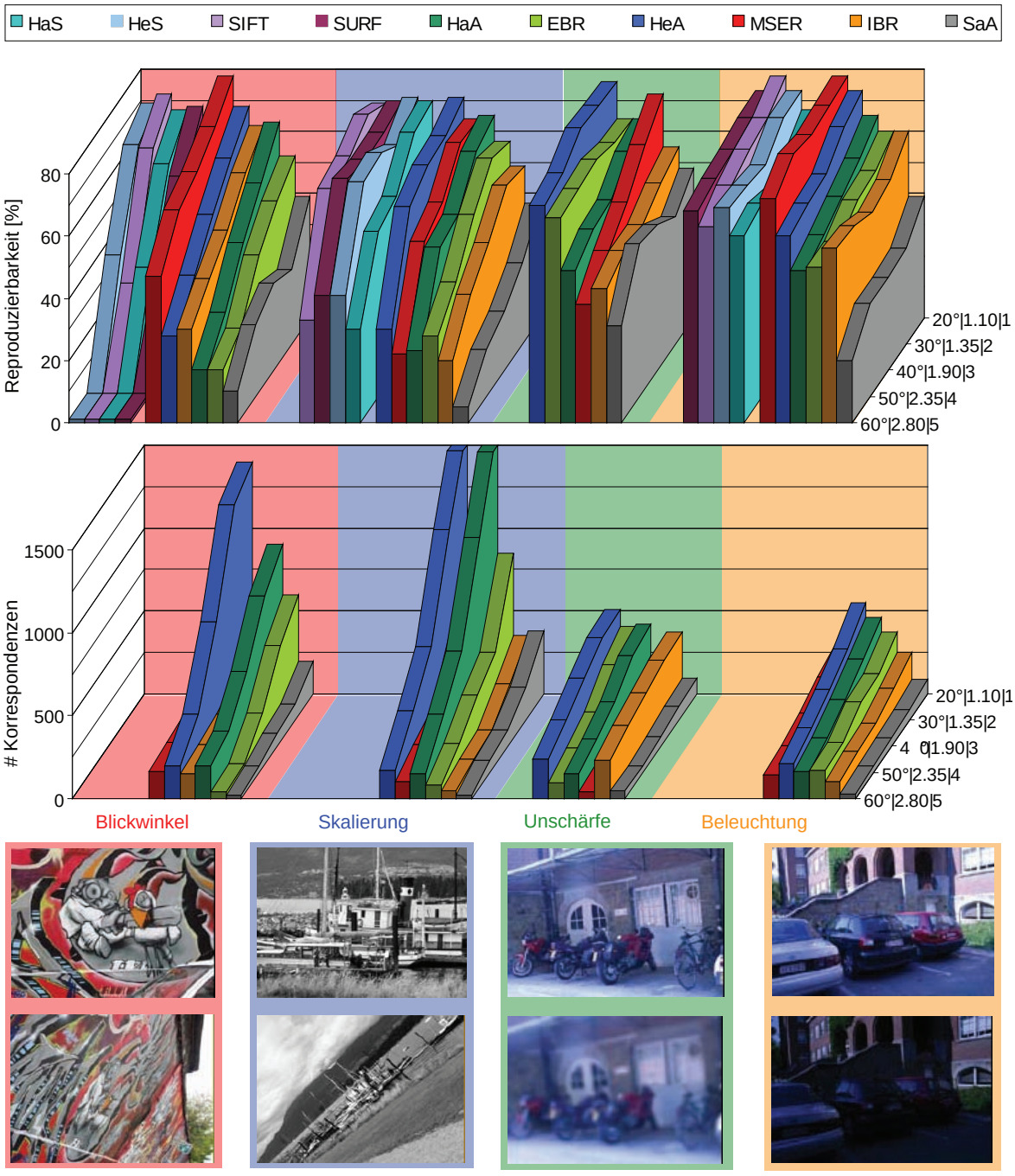

Abbildung 3.8: Vergleich der Reproduzierbarkeit ausgewählter Detektoren aus (MTS+05; BTVG06). Im unteren Drittel sind für jede der dargestellten Transformationen Blickwinkeländerung, Skalierungsänderung, Schärfevariation und Beleuchtungsschwankungen jeweils das Ausgangsbild sowie darunter das am stärksten transformierte Bild der Testsequenz abgebildet. Darüber befindet sich nach Transformation und Kovarianz-Klasse gruppiert die absolute und darüber relative Reproduzierbarkeit der untersuchten Detektoren. In der Tiefe sind die Ergebnisse der verschieden transformierten Bilder dargestellt, wobei die Transformation aus der Grafik heraus jeweils zunimmt. Für die Blickwinkeländerung und Skalierungsänderung ist rechts die entsprechende Größenordnung zu erkennen. Innerhalb der Gruppen sind die Detektoren nach relativer Reproduzierbarkeit geordnet, d.h. je weiter links sich ein Detektor befindet, desto robuster ist er im Vergleich zu den anderen Detektoren einer Kovarianz-Klasse.

dells nicht nur lineare Änderungen verkraften sondern beliebige kontinuierlichen photometrischen Transformationen. Dies erhöht die Robustheit nochmals enorm, so dass selbst bei wesentlich dunkleren Bildern fast konstant eine Reproduzierbarkeit über 70% erreicht werden kann.

Vergleicht man die Ergebnisse der Sequenzen untereinander, so erkennt man das eine Änderung des Blickwinkels oder der Skalierung die schwierigsten Transformationen darstellen. Benötigt man keine Kovarianz gegenüber großen Blickwinkeländerungen, sind die S-Detektoren aufgrund ihrer geringeren Komplexität meist etwas robuster als die affinen Verfahren. Weiterhin schlägt unter den Intensitätsextremabasierten Detektoren MSER systematisch IBR und die Hesse-Matrix basierten Detektoren die Strukturtensor basierten Verfahren. Sieht man von den Schwächen bei starker Unschärfe ab, dann ist MSER das robusteste Verfahren vor HeA und den vergleichbaren HeS, SIFT und SURF. Im Mittelfeld positionieren sich HaA, HaS, EBR und IBR, als weit abgeschlagenes Schlusslicht müssen die SaA bezeichnet werden.

Genauigkeit Ist die Reproduzierbarkeit als die Anzahl an Regionen definiert, die in beiden Bildern mind. $\kappa = 60\%$ des selben physikalischen Bildinhalts beschreiben, so wird bei der Genauigkeit untersucht, wie sich diese Anzahl bei Veränderung von κ verhält. Auswirkungen darauf haben unakkurat lokalisierte Ankerpunkte, Schwächen in der automatischen Skalenselektion oder eine unrobuste affine Adaption. Daten gibt es nur für die affinen Detektoren, es muss hier also wiederum auf die korrespondierenden, ähnlichen Detektoren geschlossen werden. Es werden hier nur die qualitativen Resultate aus (MTS$^+$05) vorgestellt, auf eine Grafik wurde verzichtet.

Erhöht man κ, d.h. fordert man eine höhere Genauigkeit der Regionen, brechen die MSER und danach die IBR am wenigsten schnell ein und liefern selbst bei einer geringen Anzahl an Regionen die exaktesten Korrespondenzen. Schlusslicht sind die HeA bzw. HaA, die zwar absolut immer noch am meisten Korrespondenzen finden, ihre relative Reproduzierbarkeit aber deutlich sinkt. Daher kann man davon ausgehen, dass die Genauigkeit der MSER und danach IBR am höchsten ist und bei den HeA bzw. HaA durch die aufwändige Konstruktion der Regionen am geringsten.

Untersucht man die Genauigkeit abhängig von der Regionengröße, lassen sich bei allen Verfahren mittlere bzw. große Regionen exakter rekonstruieren als kleine. Einzig die MSER schneiden bei allen Regionengrößen, d.h. auch bei kleinen ähnlich gut ab.

Eindeutigkeit In die Größen Reproduzierbarkeit und Genauigkeit geht die Eindeutigkeit, d.h. die Unterscheidbarkeit verschiedener Regionen nicht ein. Auf Eindeutigkeit lässt nur indirekt durch einen Vergleich von Reproduzierbarkeit (d.h. den potentiellen Korrespondenzkandidaten) und den tatsächlichen Korrepondenzen nach Auswertung des nächsten Nachbarn im Feature-Raum eines Deskriptors. Je eindeutiger bzw. aussagekräftiger die Regionen eines Detektors sind, desto getrennter liegen sie im Feature-Raum und desto weniger Regionen werden einander falsch zugeordnet. Auch hier sind nur Daten für die affinen Detektoren verfügbar. Es wird nur eine Zusammenfassung der Resultate aus (MTS$^+$05) präsentiert.

Allgemein muss man zwischen einer *distinguished* Region und einer *measurement* Region unterscheiden. Erstere ist das Ergebnis des Detektors, letzteres eine durch Skalierung $s \geq 1$ daraus gewonnene Region gleicher Ankerpunktlage und Form für die Verwendung im Deskriptor. Je größer die measurement Region, desto größer auch die potentielle Eindeutigkeit aber desto größer auch die Gefahr die lokalen geometrischen und photometrischen Modelle zu verletzen. s lässt sich daher nur anwendungspezifisch festlegen, je besser sich die Szeneinhalte allerdings durch eine Homographie beschreiben lassen oder je statischer eine Szene ist, desto größer kann s gewählt werden. Da eines von beiden auf alle Szenen in (MTS$^+$05) zutrifft, wird dort $s = 3$ gewählt.

Die Ergebnisse verhalten sich auf den Sequenzen ähnlich wie die Reproduzierbarkeit. Nur die

							kleine Transf.				große Transf.					
Algorithmus	Modell **geo**	Muster $\mathcal{M}$	Zusammenfassung	Geschwindigkeit	Eindeutigkeit	Genauigkeit	Blickwinkel	Skalierung	Schärfe	Beleuchtung	Blickwinkel	Skalierung	Schärfe	Beleuchtung	Größe	Anzahl
HaS	S	C	o+	+	o	o	o+	++	o	o+	−	o	o	o	k-m	++
HeS	S	B	+	+	o	+	+	++	+	+	--	+	+	+	k-m	++
SIFT	S	B	+	++	o	+	+	++	+	+	--	o+	+	+	k-m	++
SURF	S	B	+	++	o	+	o+	++	+	+	--	+	+	+	k-m	++
HaA	A	C	o+	o+	o	o	o+	+	o	o+	-	o	o	o	k-m	++
EBR	A	C*	−o	--	+	o	−o	o+	o	o	-	o	+	o	g	+
HeA	A	B	+	o	o	+	+	++	++	+	o	+	+	o+	k-m	++
MSER	A	F	++	+	+	++	++	+	+	++	+	o	−o	++	k-g	o
IBR	A	F	o	-	+	+	o+	o	−o	o	o	−o	−o	o+	m-g	+
SaA	A	E	-	--	+	-	-	-	-	-	-	--	o	-	m-g	-

Tabelle 3.3: Überblick und Zusammenfassung der Detektor-Bewertung durch eine Einschätzung des Autors. Von links nach rechts sind für jeden Detektor die Kovarianzklasse **geo**, das zugrundeliegende Muster $\mathcal{M}$, eine alles einschließende zusammenfassende Bewertung, die Laufzeit, die Eindeutigkeit, die Genauigkeit und aufgeschlüsselt nach kleinen und großen Transformationsunterschieden der Blickwinkel, die Skalierung, die Schärfe, die Beleuchtung sowie die Größe (k: klein, m: mittel, g:groß) und Anzahl der Regionen angegeben.

HaA bzw. HeA haben im Vergleich zu den anderen Detektoren ein geringeres Verhältnis von richtigen und falschen Zuordnungen. Ihre Regionen sind daher etwas weniger eindeutig. Absolut betrachtet liefern sie aber immer noch die größte Anzahl richtiger Korrespondenzen.

Auffällig ist ebenfalls eine starke Abhängigkeit der Ergebnisse von dem Szeneninhalt. Handelt es sich vorwiegend um texturierte Muster (z.B. eine Steinmauer) sind viele Strukturen einander ähnlich und das Verhältnis von richtigen zu falschen Korrespondenzen sinkt verglichen mit strukturierten Szenen ab. Details zu texturierten und strukturierten Szenen finden sich in (MTS$^+$05).

Betrachtet man das Verhältnis von richtigen zu falschen Korrespondenzen mit einer ansteigenden Anzahl von Regionen, so sinkt das Verhältnis allgemein ab. Detailliert betrachtet liefern MSER und IBR bei wenigen Regionen gute Ergebnisse, HeA und HaA dagegen erst bei ausreichend vielen. Auch dies deutet darauf hin, dass die auf Intensitätsextrema basierten Verfahren robuster und eindeutiger sind.

Zusammenfassung Betrachtet man Reproduzierbarkeit, Genauigkeit, Eindeutigkeit und Performance zusammen, fällt es wesentlich schwieriger eine Rangordnung der Detektoren anzugeben. Zu sehr hängt es von der Aufgabe, den Szeneninhalten und den vorkommenden Transformationen an. Tab. 3.3 gibt daher noch einmal einen Überblick über alle Detektoren, die auf der Meinung des Autors beruht. Bei geschwindigkeitsrelevanten Aufgaben sind sicherlich die ähnlichen Detektoren den affinen Detektoren vorzuziehen, ebenso bei geringen Blickwinkeländerungen. SIFT und SURF schneiden dabei vergleichbar ab, danach der etwas langsamere HeS und zuletzt der etwas weniger robuste HaS. Ist eine Änderung der Schärfe nicht zu erwarten, ist unter den affinen Detektoren MSER sowohl was die Laufzeit als auch die Robustheit angeht das Mittel der Wahl. Insbesondere bei Beleuchtungschwankungen und Blickwinkeländerungen sind diese Regionen allen anderen Verfahren

vorzuziehen. Die HeA sind zwar etwas langsamer, dafür aber robuster gegenüber Schärfe- und Skalierungsänderungen. Ähnlich aber in allen Bereichen außer der Laufzeit den HeA etwas unterlegen sind die HaA. Beide Verfahren eignen sich insbesondere dann, wenn es vorwiegend auf die absolute Anzahl an Korrespondenzen ankommt. Die IBR kann man als Ergänzung zu den MSER zählen, da sie vergleichbare homogene flächige Strukturen finden, sind ihnen aber in allen Belangen unterlegen. Die EBR sind aufgrund ihrer Laufzeit praktisch kaum interessant und auch in ihrer Qualität nur im Mittelfeld. Die SaA bzw. SaS sind sowohl was die Güte aber vor allem die Laufzeit anbelangt für die meisten Aufgaben ungeeignet.

Um regionbasierte Systeme auf unterschiedlichsten Szenen anwenden zu können, ist es sinnvoll, eine Kombination mehrerer Detektoren zu nutzen, die im Idealfall auf komplementäre Muster ansprechen. Daher sind in Tab. 3.3 für jeden Detektor auch die verschiedenen Musterklassen angegeben. Für affine Detektoren ist sicherlich eine Kombination von MSER, HeA und HaA am sinnvollsten, da sie einen großen Bereich an verschiedenen Strukturen abdecken. Benötigt man noch weitere Verfahren lassen sich die IBR als Ergänzung verwenden. Spielt die Laufzeit keine Rolle und verwendet man strukturierte Szenen mit vielen Kanten sind sicherlich die EBR ebenfalls eine interessante Erweiterung.

3.3 Deskriptoren

Ein Deskriptor $\mathbf{des}(\cdot)$ hat die Aufgabe, den *Support*, d. h. den beinhaltenden Bildinhalt einer Region auf geeignete Weise in einen Merkmalsvektor $\mathbf{r}$ zu beschreiben, um Regionen untereinander auf Ähnlichkeit vergleichen zu können. Der Merkmalsvektor verkörpert dabei im Idealfall die relevanten Informationen des zugehörigen Szenenausschnitts einer Region, losgelöst von unerwünschten Beleuchtungs- und Abbildungseinflüssen. Ein Deskriptor lässt sich u. a. formal als Funktion $\mathbf{des}(\cdot) : \bigcup_{\mathcal{M}} \rightarrow \mathcal{W}$ von der Menge aller möglichen Intensitätsmuster $\bigcup_{\mathcal{M}}$ in einen (allgemeinen) Merkmalsraum $\mathcal{W}$ verstehen. Ein Muster $\mathcal{M}(\mathrm{i}(\cdot), \mathcal{R})$ beschreibt dabei den Support der Region $\mathcal{R}$ in Bild $\mathrm{i}(\cdot)$, vereinfacht wird daher auch der Ausdruck $\mathbf{r} = \mathbf{des}(\mathcal{R})$ verwendet.

Jeder Deskriptor kann seinen eigenen Merkmalsraum definieren, meist wird allerdings für $\mathcal{W}$ der $\mathbb{R}^n_+$ bzw. für ordinale Merkmale der $\mathbb{N}^n_+$ verwendet. Zur Bestimmung von Ähnlichkeiten verschiedener Regionen wird die Distanz der zugehörigen Merkmalsvektoren im Merkmalsraum herangezogen. Dazu ist je nach Merkmalsraum und Deskriptor-Verfahren ein geeignetes Distanzmaß notwendig, um die Vektoren miteinander vergleichen zu können. Diese Maße werden in Abschnitt 3.4.1 zusammenfassend vorgestellt, in den meisten Fällen wird aber die bekannte euklidische Distanz auf dem $\mathbb{R}^n$ benutzt.

Um auf effiziente Weise robuste Korrespondenzen zwischen Regionen finden zu können, muss ein Deskriptor sehr ähnliche Eigenschaften wie ein Detektor aufweisen. Es gibt allerdings einen zentralen Unterschied, der Deskriptor soll den Merkmalsvektor *invariant* gegenüber dem geometrischen Modell $\mathbf{geo}(\cdot)$ berechnen, der Detektor die Regionen dagegen *kovariant* konstruieren. Ebenfalls ist die Invarianz gegenüber dem photometrischen Modell $\mathrm{photo}(\cdot)$ bzw. die Robustheit gegenüber Rauschen notwendig. Weiterhin soll $\mathbf{r}$ unempfindlich gegenüber kleinen Fehlern bei der Regionenkonstruktion des Detektors sein, der Deskriptor muss daher kleine Verschiebungen, Rotationen und Deformationen ausgleichen können. Ebenfalls ist eine geringe Dimension *n* des Merkmalsraum wichtig, um eine effiziente Auswertung der Distanzen während des Matchings zu ermöglichen. Der Deskriptor muss daher die Information des Grauwertmusters $\mathcal{M}$ genügend komprimieren ohne die Eindeutigkeit der Merkmalsvektoren zu verlieren. Zu Letzt soll die Berechnung des Merkmalsvektors eine geringe Komplexität aufweisen, um den Deskriptor schnell für viele Regionen berechnen zu können.

Durch die kovariante Eigenschaft der Detektoren, kann die Invarianz gegenüber $\mathbf{geo}(\cdot)$ durch eine Überführung des Supports in eine kanonische Form erreicht werden. Wie schon in Abschnitt 3.2

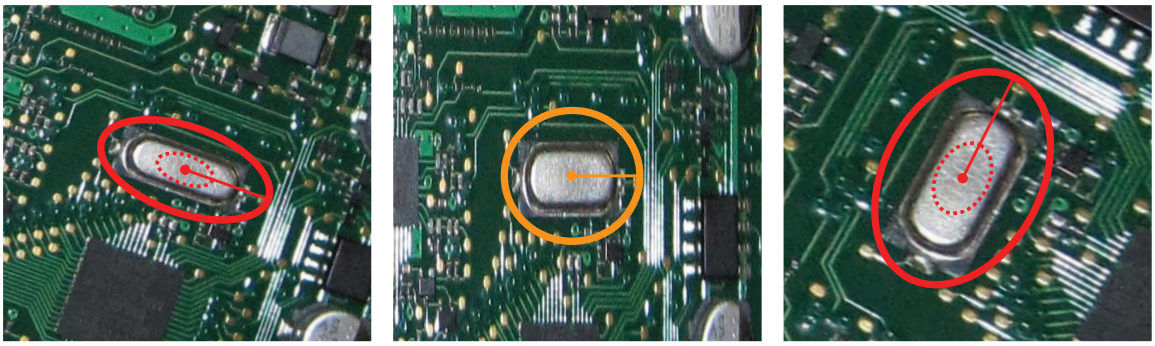

Abbildung 3.9: Beispiele für die geometrische Normalisierung von Bildpatches durch die Überführung der korrespondierenden Regionen (rot) in ihre kanonische Form (orange). Der Support einer Region ist der gesamte Bereich innerhalb der Ellipsen und beschreibt im Idealfall in allen Bildpatches den selben Szeneninhalt. Die Striche deuten die Orientierung der gerichteten Ellipse an. Beispielhaft ist ebenfalls der Unterschied zwischen der für den Deskriptor verwendeten Mess-Region (durchgezogenen rote Linie) und der vom Detektor ermittelten Konstruktions-Region (gestrichelte Linie) angegeben.

beschrieben, lässt sich eine Region $\mathcal{R}[\mathbf{u}_\perp, \mathbf{A}_\mathrm{N}]$ als gerichtete Ellipse mit den Parametern $\mathbf{u}_\perp$ und $\mathbf{A}_\mathrm{N}$ beschreiben. Die normalisierte Repräsentation N ist durch ${}^\mathrm{N}\mathcal{R} = \mathbf{A}_\mathrm{N}{}^{-1}(\mathcal{R} - \mathbf{u}_\perp)$ gegeben und überführt die Ellipse in einen Einheitskreis mit definierter Orientierung. Der Support wird auf dieselbe Weise durch ${}^\mathrm{N}\mathrm{i}(\mathbf{u}) = \mathrm{i}(\mathbf{A}_\mathrm{N}\mathbf{u} + \mathbf{u}_\perp)$ transformiert, so dass man ein geometrisch normalisiertes lokales Bildpatch ${}^\mathrm{N}\mathrm{i}(\cdot)$ erhält (vgl. Abb. 3.9), auf dem der Merkmalsvektor im weiteren Verlauf berechnet wird. Üblicherweise wird das lokale Bildpatch mit $k_\mathrm{N} = 41\,\mathrm{pix}$ je Richtung gesampelt (MS05). Dabei ist allerdings auf die Abtastung zu achten, so dass bei einem Verhältnis $\sigma = \frac{k}{k_\mathrm{N}} > 1$, wobei k die Anzahl der Pixel des ursprünglichen Supports in Richtung der größten Ausdehnung der Region bezeichnet, das Bild $\mathrm{i}(\cdot)$ erst mit einem Gaußkernel der Breite σ geglättet werden muss.

Die Detektoren konstruieren die Regionen anhand salienter Bildstrukturen wie z. B. Blobs oder homogener Flächen. Der Support dieser Regionen lässt aber meist keine ausreichend diskriminative Beschreibung zu, so dass in der Praxis die vom dem Detektor zurückgelieferte Konstruktions-Region $\mathcal{R}[\mathbf{u}_\perp, \mathbf{A}_\mathrm{N}]$ vor der Deskriptorberechnung mittels einer Skalierung s zu einer Mess-Region $\mathcal{R}' = s\mathcal{R} = \mathcal{R}[\mathbf{u}_\perp, s\mathbf{A}_\mathrm{N}]$ vergrößert wird (vgl. Abb. 3.9). Dabei muss die Skalierung groß genug sein, um eine eindeutige Beschreibung der Regionen zu ermöglichen, darf andererseits nicht zu groß sein, da sonst die Gefahr von lokalen Störungen wie Verdeckungen oder Glanzpunkten innerhalb des Supports stark ansteigt. In der Praxis wird oft $s = 3$(MS05) verwendet.

Meist werden während der Regionenkonstruktion Extrema bestimmter Funktionen ausgewertet. Die binäre Information ob die Region aus einem Minimum oder ein Maximum entstanden ist, wird oftmals *nach* der Deskriptorberechnung kodiert, in dem der Merkmalsvektor z. B. bei einem Minimum mit -1 multipliziert wird (LÁJA06). Dies eignet sich natürlich nur für Fälle, bei denen der Merkmalsraum des Deskriptors ursprünglich nur im positiven Bereich definiert ist.

Manche Detektoren verzichten bei der kovarianten Rekonstruktion auf die Ermittlung einer definierten Orientierung des normalisierten Patches. Daher sind Deskriptoren entstanden, die invariant gegenüber einer Rotation des Patches um sein Zentrum sind. Sie sind im Allgemeinen weniger Eindeutig, lassen sich aber ebenfalls bei den in dieser Arbeit vollständig konstruierten Regionen anwenden.

Die Invarianz bzgl. $\mathrm{photo}(\cdot)$ erfolgt wie in Abschnitt 3.1.1 beschrieben meist durch eine Nor-

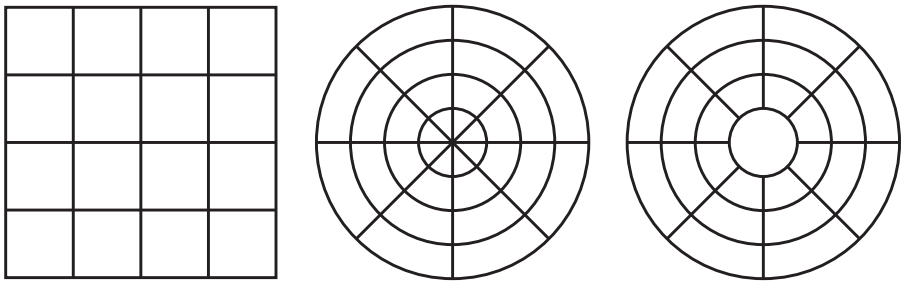

Abbildung 3.10: Räumliche Strukturen zur Tesselierungen eines Bildpatches. Links die Gitter-Struktur mit 3 horizontalen und 3 vertikalen Unterteilungen, in der Mitte die Polar-Gitter-Struktur mit 3 Unterteilungen des Radius und 7 Unterteilungen des Winkels und rechts die Zentrums-Polar-Gitter-Struktur mit den gleichen Unterteilungen wie die Polar-Gitter-Struktur. Bei den Polar-Strukturen kann optional eine logarithmische Unterteilung des Radius vorgenommen werden.

malisierung des mittleren Grauwerts und des Kontrastes des lokalen Bildpatches. Oftmals wird dies allerdings nicht explizit durchgeführt, sondern implizit durch die Verwendung von Gradienten und die Normalisierung des berechneten Merkmalsvektors.

Im einfachsten Fall kann ein Deskriptor die Intensitätswerte des normalisierten Patch in einen Merkmalsvektor übertragen und dieser mittels Kreuzkorrelation verglichen werden (z. B. in (MS05) mit einem 9×9 pix Bildpatch). Dies ist zwar sehr effizient, aber weder robust noch komprimierend. (KS04) erweitert den trivialen Ansatz auf den (unglücklich benannten) PCA-SIFT-Deskriptor, indem anstatt der Intensitätswerte die Gradientenbilder in u- und v-Richtung verwendet werden, und der entstehende Vektor mit PCA noch auf $n = 36$ Dimensionen komprimiert wird. Im Folgenden werden aber weit bessere Techniken auf der Basis von Verteilungen, räumlichen Frequenzen, Differentialen, Momenten und Ordnungen vorgestellt. Sie berücksichtigen alle nur Grauwertinformationen, für eine Verwertung von Farbinformationen wird auf (SGS09) verwiesen. Eine Bewertung aller vorgestellten Deskriptoren schließt diesen Abschnitt ab.

3.3.1 Verteilungsbasierte Techniken

Verteilungsbasierte Techniken nutzen Histogramme über bestimmte Merkmale, wie z. B. die Intensität oder den Gradienten um das normalisierte Bildpatch zu beschreiben. Dabei geht allerdings jegliche räumliche Information verloren. Es wird das Pacht daher in räumliche Zellen unterteilt, auf denen die Verteilung des betrachteten Merkmals getrennt ausgewertet wird. Die im weiteren Verlauf vorgestellten Deskriptoren *2D-Spin-Images*, *2D-Shape-Context*, *SIFT*, *SURF* und *GLOH* nutzen alle eine der in Abb. 3.10 dargestellten Tesselierungen des Bildpatches. Eine ausführliche Evaluierung weiterer räumlicher Aufteilungen findet sich in (WB07).

2D-Spin-Images Die 2D-Spin-Images (LSP03) nutzen die Polar-Gitter-Struktur mit $n_r - 1 = 4$ Unterteilungen des Radius und keiner Unterteilung des Winkels. Die Struktur besteht daher aus n_r konzentrische Ringe um das Zentrum des geometrisch und photometrisch normalisierten Patch und bewirkt einen rotationsinvarianten Merkmalsvektor. Für jeden der n_r Ringe wird ein Histogramm mit $n_i = 10$ Zellen über die Intensitäten ermittelt und alles in dem $n = n_r n_i = 50$ dimensionalen Merkmalsvektor $\mathbf{r}$ gespeichert. Dieser wird anschließend so normiert, dass der Mittelwert bzw. die Streuung über die einzelnen Einträge 0 bzw 1 ergibt.

SIFT Der SIFT-Deskriptor (Low04) ist eines der bekanntesten und robustesten Verfahren zur Beschreibung eines geometrisch normalisierten Bildpatches und weist nach (Low04) Ähnlichkeiten mit der menschlichen Wahrnehmung auf. Es nutzt für die räumliche Aufteilung die Gitterstruktur mit jeweils $n_l - 1 = 3$ Unterteilungen und insgesamt $n_l^2 = 16$ Zellen. Für jede dieser Zellen wird ein Gradienten-Orientierungs-Histogramm mit $n_\beta = 8$ Zellen nach der in Abschnitt 3.2.1.4 beschriebenen Methode aufgebaut. Um eine Robustheit gegenüber kleinen Positions- und Deformationsfehlern zu erreichen, wird jeder Pixel des Bildpatches mit einer zentrierten Gaußfunktion der Größe $\sigma = \frac{1}{2}k_N$ gewichtet. Außerdem werden die einzelnen Werte beim Eintragen in das 3D-Histogramm (räumlich in u- und v-Richtung, sowie entlang des Winkels der Gradientenrichtungen) binomial gewichtet, so dass kleine Änderungen der Werte nahe den Zellengrenzen nicht zu Sprüngen führen können sondern ebenfalls nur kleine Änderungen in dem Histogramm verursachen. Das 3D-Histogramm wird dann in den $n = n_l^2 n_\beta = 128$ dimensionalen Merkmalsvektor $\mathbf{r}$ eingetragen. Für eine photometrische Invarianz wird $\mathbf{r}$ anschließend auf die Länge 1 normiert. Danach werden alle Einträge von $\mathbf{r}$ auf maximal 0.2 begrenzt und $\mathbf{r}$ nochmals normiert. Letzteres soll verhindern, dass einzelne, besonders starke Gradienten den Vektor dominieren.

SURF Der SURF-Deskriptor (BTVG06) nutzt dasselbe Vorgehen wie der SIFT-Deskriptor. Allerdings wird anstatt von Gradientenorientierungen ein speziell auf den SURF-Detektor (siehe Abschnitt 3.2.3.4) angepasstes Histogramm über die Haar-Wavelet-Antworten δ_u bzw- δ_v in $u-$ und $v-$Richtung aufgebaut. *SURF-64* berechnet in jeder räumlichen Zelle die 4 aufsummierten Größen $(\sum \delta_u, \sum \delta_v, \sum |\delta_u|, \sum |\delta_v|)$ über alle Pixel und speichert die Werte in dem $n = 4n_l^2 = 64$ dimensionalen Vektor $\mathbf{r}$ ab. Alternativ kann bei der Summierung der Werte δ_u bzw. δ_v jeweils unterschieden werden ob δ_v bzw. δ_u positiv oder negativ ist und damit 8 Werte pro Zelle berechnet werden. Dies führt zu dem 128 dimensionalen Merkmalsvektor von *SURF-128*.

GLOH Der *Gradient-Location-Orientation-Histogramm*-Deskriptor (MS05) ist aufgebaut wie der SIFT-Deskriptor mit der einzigen Ausnahme, dass anstatt der Gitter-Struktur die Zentrums-Polar-Gitter-Struktur mit $n_r - 1 = 2$ Unterteilungen des Radius und $n_\alpha - 1 = 7$ Unterteilungen des Winkels benutzt wird und $n_\beta = 16$ Zellen bei dem Orientierungs-Histogramm. Dies führt zu einem $n = n_\beta(1 + (n_r - 1)n_\alpha) = 272$ dimensionalen Merkmalsvektor, der anschließend mittels PCA auf 128 Dimensionen komprimiert wird.

2D-Shape-Context Der 2D-Shape-Context-Deskriptor (BMP02) wurde ursprünglich für Positionen von Kanten des Canny-Edge-Detektors (Can86) entwickelt. In (MS05) wird aber wie bei dem SIFT-Deskriptors anstatt der Position der Kante deren mit der Kantenstärke gewichtete Orientierung in ein Histogramm eingetragen. Als räumliche Struktur wird die Zentrums-Polar-Gitter-Struktur mit einer logarithmischen Unterteilung des Radius bei 6, 11 und 15 pix verwendet und 4 Zellen bei dem Orientierungs-Histogramm. Dies führt auf einen 36 dimensionalen Merkmalsvektor, der abschließend auf $|\mathbf{r}| = 1$ normiert wird.

3.3.2 Frequenzbasierte Techniken

Im Bereich der Texturanalyse werden oftmals Merkmale aus dem Orts-Frequenz-Raum herangezogen. Dieser kann z. B. mit einer Fourier-Transformation berechnet werden. Allerdings handelt es sich dabei um eine globale Transformation ohne explizite Ortsrepräsentation und mit unendlichen Basisfunktionen, was eine Anwendung auf lokale Bildpatches schwierig macht. Die Wavelet- (VK95) oder die Gabortransformation (Gab46) überwinden diese Probleme. Sie sind allerdings ein Forschungsgebiet für sich allein und sollen hier nicht weiter beschrieben werden. Der schon erwähnte SURF-Deskriptor könnte auch diesem Abschnitt zugeordnet werden.

3.3.3 Differentielle Techniken

Differentielle Techniken nutzen Derivate des geometrisch und photometrisch normalisierten Bildpatches durch Faltung mit einer Gaußfunktion $g(\mathbf{u})$ der Größe $\sigma = 6.7\,\text{pix}$ (MS05) bei einer Patchbreite von 41 pix. Diese werden auch als *lokale Jets* bezeichnet. (MS05) definiert zwei Deskriptoren, die *differentiellen Invarianten* (FHRKV94) und die eng mit den lokalen Jets verknüpften *Steerable Filters* (KVD87; FA91). Diese Deskriptoren werden in der Bewertung berücksichtigt aber in dieser Arbeit nicht weiter erläutert. Ihre Details können ausführlich in den zitierten Artikeln nachgeschlagen werden. Im Weiteren werden abschließend die *komplexen Filter* kompakt beschrieben.

Komplexe Filter Die komplexen Filter (SZ02) sind ein rotationsinvarianter Deskriptor der mittels einer Bank linearer Filter berechnet wird. Die Filterfamilie wird definiert durch

$$k_{mn}(u,v) = (u+\mathrm{i}v)^m (x-\mathrm{i}v)^n \mathrm{g}(u,v) \tag{3.16}$$

wobei $\mathrm{g}(u,v)$ obig erwähnte Gaußfunktion darstellt. Es werden 15 Filter mit $0 < m+n \leq 6$, $m \geq n \geq 0$ im Bereich des Einheitskreises berechnet und dann mit der gleichen Auflösung wie das lokale, geometrisch und photometrisch normalisierte Bildpatch gesampelt. Von den 15 komplexen Filterantworten auf dem Bildpatch wird der rotationsinvariante Betrag in den $n = 15$ dimensionalen Merkmalsvektor eingetragen.

3.3.4 Momentenbasierte Techniken

Es gibt eine Vielzahl unterschiedlicher Momente und Deskriptoren auf Basis von Momenten, wobei ein kleiner Teil schon in Abschnitt 2.1.1.3 bei der Beschreibung ganzer Ansichten behandelt wurde. Dieser Abschnitt beschränkt sich auf einen Ansatz mit *generalisierten Grauwertmomenten*:

$$m^a_{pq} = \frac{1}{4l^2} \sum_{-l \leq u,v \leq l} u^q v^p \mathrm{i}(u,v)^a \qquad u,v \in \mathbb{N} \tag{3.17}$$

der Ordnung $p+q$ und Grad a. Diese können allgemein über einen beliebigen Bildbereich angegeben werden, werden hier aber auf dem geometrisch und photometrisch normalisierten Bildpatch der Kantenlänge $2l = 41\,\text{pix}$ berechnet. Sie können auf eine beliebige Anzahl von Bildkanälen erweitert werden und wurden ursprünglich für Farbbilder vorgestellt (MMVG98).

Die generalisierten Grauwertmomente codieren reine Formmomente m^0_{pq}, wie dem Schwerpunkt (m^0_{10}, m^0_{01}), reine Intensitätsmomente m^a_{00}, wie dem mittleren Grauwert m^1_{00} und kombinierte Größen. Sie können herangezogen werden, um invariante Größen bzgl. dem geometrischen und dem photometrischen Model abzuleiten (MMVG98). Im weiteren Verlauf wird das Verfahren von (MS05) beschrieben und als *Momenten-Deskriptor* bezeichnet.

Momenten-Deskriptor Aufgrund der Normalisierung des Patches müssen keine abgeleiteten invarianten Größen betrachtet werden, sondern es können direkt die Momente für den Deskriptor herangezogen werden. Momente der Ordnung oder des Grades 0 besitzen dabei aber triviale Werte und werden nicht berücksichtigt. Höhere Momente sind sensitiv gegenüber geometrischen oder photometrischen Störungen. Es werden daher hier nur die 10 unterschiedlichen Momente der Ordnung und des Grades 1 und 2 betrachtet. Die Berechnung erfolgt nicht direkt auf den Intensitäten sondern auf den Gradientenbildern in u- und v-Richtung, so dass insgesamt $n = 20$ Momente berechnet und in dem Merkmalsvektor eingetragen werden.

3.3.5 Ordnungsbasierte Techniken

Dieser Abschnitt beschäftigt sich mit einer Technik, die anstatt metrischer Merkmale nur deren Ordnung betrachtet und daraus ordinale Merkmale ableitet. Dieses Vorgehen hat den Vorteil, dass Ordnungen invariant gegenüber jeglicher *monotonen* Transformation der metrischen Merkmale sind und daher wesentlich flexibler, als z. B. eine affin-invariante Konstruktion durch Normalisierung von Mittelwert und Streuung. Aus Ordnungen lassen sich eigenständige Deskriptoren, wie z. B. der *Ordinal-Spatial-Intensity-Distribution*-Deskriptor (OSID) ableiten und eine Meta-Technik angeben, die als Nachverarbeitung mit jedem beliebigen Deskriptor anwendbar ist.

Meta-Technik (TW09) stellt ein Verfahren vor, das auf jeden beliebigen Merkmalsvektor $\mathbf{r} = (r_1, \ldots, r_n)^{\mathrm{T}}$ eines Deskriptors anwendbar ist, falls sich dessen einzelne Elemente ordnen lassen. Dafür wird $\mathbf{r}$ in seine Rangrepräsentation $\mathbf{o} = (o_1, \ldots, o_n)^{\mathrm{T}} \in \mathbb{N}_+^n$ mit $o_i = |\{r_j | r_j \leq r_i\}|$ überführt, welche in jedem Element o_i nur noch die Stelle enthält, an der das ursprüngliche Element r_i nach der Sortierung steht. $\mathbf{o}$ ist daher eine der $n!$ Permutationen des Vektors $(1, \ldots, n)^{\mathrm{T}}$. Die Rangrepräsentation ist daher nur diskriminativ, falls der ursprüngliche Deskriptor einen genügend hochdimensionalen Merkmalsvektor berechnet. Durch Experimente wird in (TW09) vermutet, dass $n = 36$ dafür noch nicht ausreicht, $n = 128$ dagegen auf jeden Fall. Zum Vergleich der Rangrepräsentation wird der in Abschnitt 3.4.1 beschriebene Spearman-Korrelationskoeffizient benutzt. Die Meta-Technik erhöht zwar einerseits die Komplexität des Deskriptors bei der Berechnung, andererseits wird aber auch bei genügend großem n die Robustheit des Deskriptors verbessert. (TW09) wendet sie auf die schon beschriebenen Deskriptoren SIFT, GLOH und PCA-SIFT an. Die kombinierten Deskriptoren werden als *R-SIFT*, *R-GLOH*, und *R-PCA-SIFT* bezeichnet und ebenfalls bei der Bewertung im folgenden Abschnitt mit den anderen Deskriptoren verglichen.

OSID Für den OSID-Deskriptor (TLCT09) werden die Intensitätswerte innerhalb des geometrisch normalisierten Bildpatches geordnet und jeder Pixel mit seinem Intensitäts-Rang gelabelt. Dies gewährleistet nicht nur eine Invarianz gegenüber dem linearen Modell photo$(\cdot)$, sondern gegenüber jeglicher monotonen Transformation der Bildintensitäten. Dann wird das Bildpatch wie bei den verteilungsbasierten Deskriptoren räumlich in 16 Zellen mit der Polar-Gitter-Struktur (0 Unterteilungen von r, 15 Unterteilungen von α) unterteilt. Für jede räumliche Zelle wird ein Histogramm mit 8 gleichmäßigen Zellen über den gelabelten Intensitäts-Rang aufgebaut und alles in einem $n = 128$ dimensionalen Merkmalsvektor $\mathbf{r}$ gespeichert.

3.3.6 Bewertung der Deskriptoren

In (MS05; BTVG06; TW09; TLCT09) werden die Deskriptoren anhand unterschiedlicher Sequenzen, Einflüsse und mit unterschiedlichen Parametereinstellungen evaluiert. Dieser Abschnitt stellt eine kurze qualitative Zusammenfassung dieser Untersuchungen in Tab. 3.4 dar. Dabei wurden die ausführlichen Ergebnisse auf zwei wesentlichen Bewertungskriterien verdichtet. Erstens die *Rechenzeit*, die ein Deskriptor-Algorithmus zur Berechnung eines Merkmalsvektors aus einem Bildpatch benötigt und zweitens die *Güte* eines Deskriptors bei der Korrespondenzzuordnung. Das zweite Kriterium leitet sich aus den in (MS05; BTVG06; TW09; TLCT09) verwendeten quantitativen Größen der *Reproduzierbarkeit*, dem Verhältnis zwischen der Anzahl der korrekten Zuordnungen und der theoretisch möglichen korrekten Zuordnungen und der *1-Präzision*, dem Verhältnis der Fehlkorrespondenzen zu allen gefundenen Korrespondenzen, ab. Aufgrund der qualitativen Betrachtung stellt Tab. 3.4 eine persönliche Einschätzung des Autors dar.

Bei der Beschreibung der Deskriptoren wurden jeweils die besten Parametereinstellungen angegeben, die in den jeweiligen Artikeln bei der Evaluierung der Deskriptoren ermittelt wurden. Dies ist

Deskriptor	Dimension	Rotationsinvariant	Distanzmaß	Rechenzeit	Güte
Grauwerte	81	Nein	Korrelation	++	−o
PCA-SIFT	36	Nein	Euklid	+	o
Verteilungsbasierte Techniken					
SIFT	128	Nein	Euklid	o+	+
SURF-128	128	Nein	Euklid	+	+
GLOH	128	Nein	Euklid	o	+
SURF-64	64	Nein	Euklid	+	+
2D-Spin-Images	50	Ja	Euklid	+	-
2D-Shape-Context	36	Nein	Euklid	o+	o+
Differentielle Techniken					
Steerable Filter	41	Nein	Mahalanobis	+	-
Komplexe Filter	15	Ja	Mahalanobis	+	--
Differentielle Invarianten	8	Ja	Mahalanobis	+	--
Momentenbasierte Techniken					
Momente	20	Nein	Mahalanobis	+	−o
Ordnungsbasierte Techniken					
OSID	128	Nein	Euklid	o	++
R-SIFT	128	Nein	Spearman	o	++
R-GLOH	128	Nein	Spearman	−o	+
R-PCA-SIFT	36	Nein	Spearman	o+	−o

Tabelle 3.4: Überblick und Zusammenfassung der Deskriptor-Bewertung durch eine Einschätzung des Autors anhand der Ergebnisse aus (MS05; BTVG06; TW09; TLCT09). Es sind ebenfalls die zur Evaluierung verwendeten Distanzmaße angegeben. Sie werden in Abschnitt 3.4.1 beschrieben.

insbesondere wichtig, falls diese Parameter einen Einfluss auf die Dimension n des Merkmalsvektors haben. Ist diese zu niedrig, repräsentiert der Merkmalsvektor keine eindeutige Beschreibung. Daher schneiden die Deskriptoren in Tab. 3.4 mit niedriger Dimension im Schnitt schlechter ab als die höher dimensionalen Verfahren. Wird die Dimension zu groß sinkt die Robustheit gegenüber Störeinflüssen und die Reproduzierbarkeit sinkt. Es finden sich daher in Tab. 3.4 keine Deskriptoren mit $n > 128$.

Unter den niedrig-dimensionalen Deskriptoren mit $n \leq 20$ sind die Momente den komplexen Filtern und den differentiellen Invarianten bei der Güte deutlich überlegen.

Bei den Deskriptoren mit mittlerer Dimensionalität $20 < n \leq 50$ schneidet der 2D-Shape-Kontext am besten ab, gefolgt von PCA-SIFT. Allerdings hat der 2D-Shape-Context Schwierigkeiten bei texturierten Szenen ohne ausgeprägte Kanten (MS05). Die Steerable Filter und die 2D-Spin-Images schneiden trotz ihrer höheren Dimension schlechter ab als die Momente.

Die hoch-dimensionalen Deskriptoren mit $n > 50$ schneiden bei der Güte mit Abstand am besten ab. SIFT, GLOH und SURF verhalten sich sehr ähnlich, wobei SIFT im Gegensatz zu SURF leichter anwendbar ist und im Gegensatz zu GLOH die bessere Balance zwischen Rechenzeit und Güte besitzt. Die ordnungsbasierten Deskriptoren OSID und R-SIFT sind zwar etwas aufwändiger zu berechnen besitzen aber nochmals eine bessere Güte als die restlichen Deskriptoren. R-GLOH und R-PCA-SIFT können von der Meta-Technik allerdings nicht profitieren, bei GLOH wird in (TW09) eine ungünstige Korrelation mit der PCA-Reduktion vermutet und bei R-PCA-SIFT eine zu niedrige Dimension.

Zusammenfassend lässt sich sagen, dass die verteilungsbasierten Deskriptoren den differentiellen und momentenbasierten Deskriptoren deutlich überlegen sind. Sie sind deutlich robuster und gehören bei einer Kombination mit den ordnungsbasierten Techniken zu den besten Verfahren die z. Z. im wissenschaftlichen Umfeld verfügbar sind. OSID besitzt von allen getesteten Deskriptoren die beste Gü-

te und sollte daher die erste Wahl ein, falls eine hohe Korrespondenzausbeute benötigt wird. R-SIFT dagegen hat den Vorteil, dass der Merkmalsvektor im Gegensatz zu OSID (und fast allen anderen Verfahren) nicht mit Fließkommazahlen dargestellt werden muss, sondern wesentlich effizienter mit Ganzzahlen repräsentiert werden kann. Dies führt zu einer kompakteren Speichergröße und schnelleren Verarbeitung bei der Korrespondenzzuordnung. Benötigt man einen niedrig-dimensionalen Deskriptor sind die Momente die beste Alternative, allerdings müssen dort im Vergleich zu OSID deutliche Abstriche bei der Güte gemacht werden.

3.4 Matcher

Der abschließende Schritt zur Ermittlung lokaler Korrespondenzen auf Basis interessanter Regionen ist der Matcher. Er sucht für eine gegebene Region $\mathcal{R}_a$ in einer Datenbank $\mathcal{DB}$ vieler gegebener Regionen $\mathcal{R}_i \in \mathcal{DB}$ korrespondierende Paare $(\mathcal{R}_a, \mathcal{R}_i)$ anhand einer *Matching Strategie* und einem *Distanzmaß* $d_{\mathbf{des}}(\cdot)$ auf dem Merkmalsraum der Deskriptoren. Im Folgenden werden zuerst verschiedene Distanzmaße und abschließend unterschiedliche Matching Strategien vorgestellt.

3.4.1 Distanzmaße

Zum Auffinden korrespondierender Regionen wird ein Maß $d_{\mathbf{des}}(\cdot) = d(\cdot)$ benötigt, um die Ähnlichkeit der zugehörigen Merkmalsvektoren bestimmen zu können. Dazu kann entweder ein *Distanzmaß* $d(\cdot) : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_0^+$ oder ein *Ähnlichkeitsmaß* $s(\cdot) : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ auf dem Merkmalsraum $\mathcal{W}$ des Deskriptors verwendet werden. In beiden Fällen müssen die Maße die Symmetrie $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ bzw. $s(\mathbf{x},\mathbf{y}) = s(\mathbf{y},\mathbf{x})$ für alle $\mathbf{x},\mathbf{y} \in \mathcal{W}$ erfüllen. Für das Distanzmaß muss weiterhin $d(\mathbf{x},\mathbf{x}) = 0$ und $d(\mathbf{x},\mathbf{y}) > 0$ für alle $\mathbf{x},\mathbf{y} \in \mathcal{W}$, $\mathbf{x} \neq \mathbf{y}$ gelten. Im Gegensatz dazu muss für das Ähnlichkeitsmaß zusätzlich $d(\mathbf{x},\mathbf{x}) > d(\mathbf{x},\mathbf{y})$ für alle $\mathbf{x},\mathbf{y} \in \mathcal{W}$, $\mathbf{x} \neq \mathbf{y}$ gelten. In dieser Arbeit werden die Matching Strategien mittels eines Distanzmaßes angegeben, so dass jedes Ähnlichkeitsmaß über eine Formel, wie z. B.

$$d(\mathbf{x},\mathbf{y}) = \sqrt{s(\mathbf{x},\mathbf{x}) + s(\mathbf{y},\mathbf{y}) - 2s(\mathbf{x},\mathbf{y})} \tag{3.18}$$

in ein Distanzmaß übergeführt werden muss. Oftmals wird nicht die tatsächliche Distanz, sondern nur eine Rangordnung der Abstände mehrerer Paare von Merkmalsvektoren benötigt. In diesem Fall können alle monotonen Funktionen auf oberster Ebene, wie z. B. die Wurzel in obigem Beispiel aus Effizienzgründen weggelassen werden. Im weiteren Verlauf werden die wichtigsten Maße im Bereich der Korrespondenzzuordnung vorgestellt und gegebenenfalls ihre Besonderheiten diskutiert. Die Variable n bezieht sich dabei auf die Dimension eines Merkmalsvektors $\mathbf{x} = (x_1,\ldots,x_n)^{\mathrm{T}} \in \mathcal{W}$.

Euklidische Distanz

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.19}$$

Sie ist das bekannteste und in der Literatur (Low04; MS05; BTVG06; TLCT09) am häufigsten eingesetzte Maß zum Vergleich zweier Merkmalsvektoren. Sie sollte nur verwendet werden, falls die Merkmale der einzelnen Elemente x_i demselben Skalenbereich zugeordnet werden können. Dies ist z. B. für alle Histogramme der Fall.

Mahalanobis-Distanz

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^{\mathrm{T}}\mathbf{P}^{-1}(\mathbf{x}-\mathbf{y})} \tag{3.20}$$

Die euklidische Distanz wird durch eine Berücksichtigung unterschiedlicher Skalenbereiche durch die Mahalanobis-Distanz erweitert und entspricht für $\mathbf{P} = \mathbf{I}_{n\times n}$ ihrem Ergebnis. Über die Kovarianzmatrix $\mathbf{P}$ kann die Varianz, d. h. der Skalenbereich der einzelnen Merkmale sowie der statistische Zusammenhang zwischen den einzelnen Merkmalen modelliert werden. Die Mahalanobis-Distanz eignet sich daher für unterschiedlich skalierte Merkmale wie z. B. den Momenten. Die Kovarianzmatrix $\mathbf{P}$ wird meist aus einer großen Zahl realer Merkmalsvektoren ermittelt.

Earth-Movers-Distanz Die in (PW08) präsentierte *Earth-Movers-Distanz* (EMD) ist eine spezielle Metrik für Histogramme, die die minimalen Kosten zur Überführung zweier Histogramme berechnet. Sie kann auf Ringstrukturen wie z. B. den Gradienten-Orientierungs-Histogrammen optimiert werden und zeigt dort eine bessere Eignung als die euklidische Distanz (PW08). Die EMD lässt sich zwar über Max-Flow-Min-Cut-Algorithmen in $O(n)$ berechnet, benötigt aber u. a. durch eine Überführung des Problems in eine Graphenstruktur bei der Berechnung um Größenordnungen länger als alle anderen hier vorgestellten Maße. Sie ist daher für zeitkritische Systeme nicht zu empfehlen.

χ^2-Distanz

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{3.21}$$

Dieses Maß beruht auf dem χ^2-Test zweier Häufigkeitsverteilungen auf Ähnlichkeit und kann verwendet werden, wenn sich die Merkmale sinnvoll als Häufigkeitsverteilung interpretieren lassen (BMP02). Dafür müssen die beiden Merkmalsvektoren zumindest auf $|\mathbf{x}| = |\mathbf{y}| = 1$ normiert sein.

Kreuzkorrelation

$$s(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} x_i y_i \tag{3.22}$$

Dieses Ähnlichkeitsmaß findet häufig Verwendung bei dem Vergleich zweier Grauwertbilder mittels einer Faltung und wird hier angepasst auf den Vergleich zweier Vektoren in der diskreten Variante ohne Verschiebungsparameter dargestellt.

Innenwinkel

$$s(\mathbf{x}^*,\mathbf{y}^*) = \arccos \sum_{i=1}^{n+1} x_i^* y_i^* \tag{3.23}$$

Für die Berechnung des Innenwinkels müssen die beiden Merkmalsvektoren $\mathbf{x}$ und $\mathbf{y}$ mittels der in Gl. 2.6 angegeben Funktion $\gamma(\cdot)$ erst aus dem n dimensionalen kartesischen Merkmalsraum auf die $n+1$ dimensionale Einheitskugel projiziert werden (RCSM01). Damit lässt sich dann sehr effizient entweder der Innenwinkel $\alpha = s(\mathbf{x}^*,\mathbf{y}^*)$ bzw. der Korrelationskoeffizient $\tau = \cos\alpha$ aus $\mathbf{x}^* = \gamma(\mathbf{x})$ und $\mathbf{y}^* = \gamma(\mathbf{y})$ berechnet. (RCSM01) sieht in diesem Vorgehen einen Vorteil im Vergleich zur euklidischen Distanz.

Spearman-Rangkorrelationskoeffizient

$$s(\mathbf{x},\mathbf{y}) = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)} \tag{3.24}$$

Dieses Maß ist auf ordinale Merkmalsvektoren definiert, bei denen die einzelnen Einträge $x_i \in \mathbb{N}_0^+$ einen Rang repräsentieren und der gesamte Merkmalsvektor aus einer Permutation von $(1,\ldots,n)^{\mathrm{T}}$ besteht (TW09). Alternativ gibt es auch noch das wesentlich aufwändigere *Kendalls Tau*, welches aber nach (TW09) für die Korrespondenzzuordnung keinen Vorteil bringt.

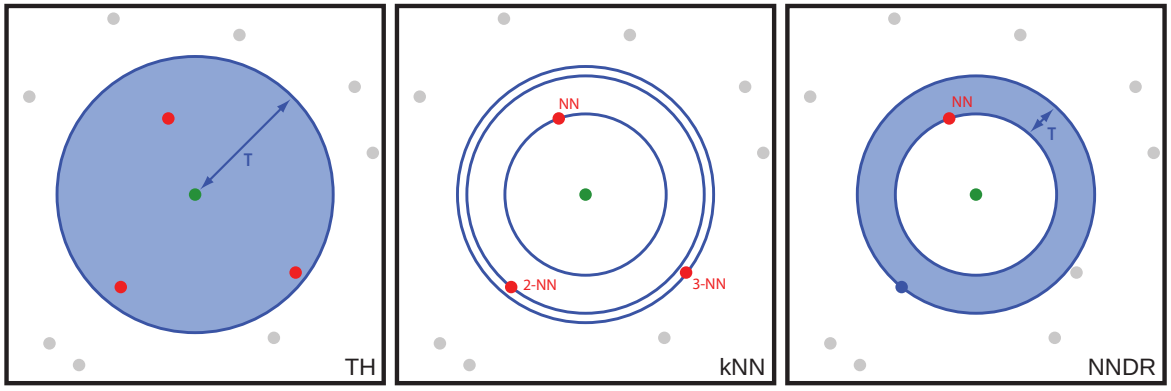

Abbildung 3.11: Unterschiedliche Matching Strategien bei der Findung lokaler Korrespondenzen. Dargestellt ist der Merkmalsraum mit Merkmalsvektor $\mathbf{r}_a$ (grün) der Anfrage-Region, den Vektoren der Datenbank (grau) und den darin gefundenen Vektoren (rot) der Korrespondenzmenge $\mathcal{K}$. Die Bereichsanfrage (TH) ermittelt alle Vektoren in einem festgelegtem Bereich um $\mathbf{r}_a$, die Nächste-Nachbar-Anfrage (kNN) nutzt die k nächsten Vektoren, und die Nächste-Nachbarn-Distanz-Verhältnis-Anfrage (NNDR) gibt den NN zurück, falls dieser eindeutig mit großer Toleranz ermittelt werden kann.

3.4.2 Matching Strategien

Die Aufgabe des Matchers ist es, für eine gegebene Anfrage-Region $\mathcal{R}_a$ eine Korrespondenzmenge $\mathcal{K} \subseteq \mathcal{DB}$ aus der Datenbank zu bestimmen. Aus dieser Menge werden alle Korrespondenzen $(\mathcal{R}_a, \mathcal{R}_i)$ mit $\mathcal{R}_i \in \mathcal{K}$ gebildet. $\mathcal{K}$ kann je nach Matching Strategie und Anfrage-Region leer, genau eine Region oder mehrere Regionen enthalten.

Die Bestimmung der Korrespondenzmenge erfolgt über das schon diskutierte Distanzmaß $d_{\mathbf{des}}(\cdot)$ im Merkmalsraum des Deskriptors. Im weiteren Verlauf werden die Merkmalsvektoren $\mathbf{r}_i = \mathbf{des}(\mathcal{R}_i)$ mit den Regionen gleichgesetzt, so dass $\mathcal{K}$ bzw. $\mathcal{DB}$ alternativ ebenfalls als eine Menge von Merkmalsvektoren dargestellt werden kann. $\mathbf{r}_a$ bezeichnet in diesem Fall immer den Merkmalsvektor der Anfrage-Region. Abb. 3.11 stellt die im Folgenden diskutierten Matching Strategien (vgl. dazu auch (Low04; MS05)) visuell dar.

Bereichsanfrage (TH) Im einfachsten Fall werden alle Regionen als korrespondierend angenommen, die eine gewisse Ähnlichkeit mit der Anfrage-Region aufweisen bzw. deren Merkmalsvektoren in einem gewissen Bereich um den Anfragevektor liegen. Der Bereich wird je nach Distanzmaß durch eine n-dimensionale „Kugel“ im Merkmalsraum mit Schwellwert (TH) bzw. Radius τ definiert. Formal lässt sich $\mathcal{K}$ wie folgt definieren:

$$\mathcal{K} = \{\mathbf{r} \mid d_{\mathbf{des}}(\mathbf{r}_a, \mathbf{r}) < \tau, \mathbf{r} \in \mathcal{DB}\} \tag{3.25}$$

Je nach Lage des Anfragevektors und der Verteilung der Vektoren in der Datenbank kann $\mathcal{K}$ keine oder sehr viele korrespondierende Regionen enthalten. Diese Matching Strategie wird daher häufig angewendet, falls viele Korrespondenzen verarbeitet werden können, wie z. B. bei der Bildersuche.

Nächste-Nachbar-Anfrage (kNN) Während bei der Bereichsanfrage der Bereich festgelegt und die Anzahl der Korrespondenzen variabel belassen wird, wird bei der Nächsten-Nachbarn-Anfrage

genau umgekehrt vorgegangen. Unabhängig von dem tatsächlichen Abstand im Merkmalsraum, werden die k am nächsten zum Anfragevektor liegenden Merkmalsvektoren bestimmt und zurückgegeben. Die Korrespondenzmenge hat daher genau die Größe $|\mathcal{K}| = k$ und enthält alle i-Nächsten-Nachbarn $\mathbf{r}_{\mathrm{NN}}[i]$ mit $0 < i \leq k$:

$$r_{\mathrm{NN}}[i] := \underset{\mathbf{r} \in \mathcal{DB} \setminus \{r_{\mathrm{NN}}[1],\ldots,r_{\mathrm{NN}}[i-1]\}}{\operatorname{argmin}} d_{\mathbf{des}}(\mathbf{r}_a, \mathbf{r}) \tag{3.26}$$

In der Praxis wird im Anschluss meistens noch überprüft ob die gefundenen Regionen eine gewisse Mindestähnlichkeit $d_{\mathbf{des}}(\mathbf{r}_a, \mathbf{r}) < \tau$ erfüllen. Die kNN-Anfrage eignet sich besser als die Bereichs-Anfrage, falls aufgrund unbekannter photometrischer und geometrischer Transformationen die Abstände zwischen den Merkmalsvektoren stark variieren. Insbesondere der Nächste-Nachbar $\mathbf{r}_{\mathrm{NN}} = \mathbf{r}_{\mathrm{NN}}[1]$ wird häufig genutzt wenn genau eine Korrespondenz erwartet wird, wie es z. B. oftmals bei Objekterkennungssystemen der Fall ist (Low04).

Nächste-Nachbarn-Distanz-Verhältnis-Anfrage (NNDR) Diese Matching Strategie liefert in $\mathcal{K}$ den Nächsten-Nachbarn $\mathbf{r}_{\mathrm{NN}}$ zurück, falls dieser die Überprüfung des Verhältnisses

$$\frac{d_{\mathbf{des}}(\mathbf{r}_a, \mathbf{r}_{\mathrm{NN}})}{d_{\mathbf{des}}(\mathbf{r}_a, \mathbf{r}_{\mathrm{NN}}[2])} < \tau \tag{3.27}$$

der Distanzen zwischen dem NN und 2-NN besteht (Low04). Die NNDR-Strategie erhöht daher die Präzision bei der Korrespondenzfindung auf Kosten der Reproduzierbarkeit, da leicht verwechselbare Regionen ignoriert werden. Sie sollte eingesetzt werden, wenn man wenige, dafür aber zuverlässige Korrespondenzen benötigt.

Implementierungsdetails Je nach Größe $|\mathcal{DB}|$ der Datenbank und der Dimensionalität n des Merkmalsraum können die Anfragen sehr schnell abgearbeitet werden oder benötigen einen enormen Zeitbedarf. Es sollen daher hier kurz unterschiedliche Implementierungsstrategien diskutiert werden. Eine Brute-Force-Implementierung vergleicht alle Merkmalsvektoren der Datenbank mit dem Anfragevektor und ist daher von der theoretischen Seite betrachtet nicht sehr effizient. Allerdings kann man die Implementierung aufgrund ihrer Einfachheit sehr stark optimieren und damit für kleine Datenbankgrößen schnelle Auswertezeiten erreichen.

Eine Alternative sind intelligente, meist hierarchische Datenstrukturen (für eine Auflistung siehe (Tra09)) die eine geschickte Partitionierung der Daten oder des Merkmalsraums vornehmen, um eine effizientere, selektive Auswertung vornehmen zu können. Sie sind allerdings deutlich komplexer, schwieriger zu optimieren und besitzen einen langsamen Initialisierungsschritt zum Partitionieren der Datenbank. Außerdem sind sie vom Fluch der Dimensionen betroffen und werden mit steigendem n ineffektiv.

Die letzte Möglichkeit sind approximative Techniken, die zwar meistens aber nicht immer das korrekte Ergebnis liefern. Dazu gehören z. B. die Komprimierungstechniken wie PCA oder die *Random-Projection* (RCSM01) zur Verringerung der Dimension des Merkmalsraums oder eine schnelle Auswertung der kNN-Anfrage mittels *Best-Bin-First* Technik (BL99). Sie erkaufen sich allerdings die Beschleunigung auf Kosten der Präzision bei der Korrespondenzfindung. Abschnitt 6.1.1 enthält konkrete Vergleiche der Strategien.

EINE GUTE THEORIE
IST DAS PRAKTISCHSTE,
WAS ES GIBT.

u. a. Gustav R. Kirchhoff

Kapitel 4

Konzept

Dieses Kapitel beschäftigt sich mit dem grundlegenden Konzept zur Lösung der in Abschnitt 1 vorgestellten Aufgabenstellung. Dazu wird in Abschnitt 4.1 das Problem exakt definiert, die zu beachtenden Randbedingungen vorgestellt und die in Kapitel 2 vorgestellten Verfahren daran gespiegelt. Abschnitt 4.2 analysiert Stärken und Schwächen der favorisierten Verfahren und stellt Verbesserungs- bzw. Erweiterungsmöglichkeiten vor. Diese werden in dem neu entwickelten Ansatz, der *6 DoF Lageerkennung mit kovarianten Regionen* in Abschnitt 4.3 umgesetzt. Anschließend wird in Abschnitt 4.4 der notwendige, aber autonome Trainingsschritt behandelt. Abschnitt 4.5 stellt abschließend optionale Erweiterungen vor. Die Einschätzung dieses neu entwickelten Verfahrens ist in Tab. 2.1 mit aufgeführt.

4.1 Problemdefinition und Randbedingungen

Das in dieser Arbeit zu entwickelnde Verfahren soll die Ermittlung der Lage von beliebigen, aber genügend texturierten bzw. strukturierten Starrkörper-Objekten in allen 6 Freiheitsgraden (3 rotatorische und 3 translatorische) ermöglichen. Dazu steht die in Abschnitt 1.3 vorgestellte Sensorik, eine Stereokamera mit homogener Beleuchtungseinheit zur Verfügung.

Das Verfahren soll aus projektrelevanten Gründen *single-shot* fähig sein, d. h. eine robuste Lage schon mittels eines einzigen Stereobildes ermitteln können. Explorative oder regelnde Ansätze können daher nicht angewendet werden und sind allenfalls als Erweiterung denkbar. Ebenfalls können Tiefenbilder nur aus einem einzigen Stereobild gewonnen werden, welches die zu erwartende Güte im Gegensatz zu einer geeignet aufgenommenen Sequenz deutlich reduziert. Verfahren aus Tab. 2.1, die eine hohe Güte (D^+) der Tiefeninformationen voraussetzen, sind daher für diese Arbeit nicht geeignet.

Weiterhin soll das Verfahren komplexe Szenen, d. h. mit unstrukturiertem Hintergrund und vielen, auch gleichen, Objekten beherrschen. Es ist daher eine nicht triviale Segmentierung notwendig, für die keines der in Kapitel 2 vorgestellten *globalen* Verfahren eine Lösung anbietet. Gegen die meisten globalen Verfahren spricht ebenfalls die Problematik bei Verdeckungen und lokalen Störungen bei Glanzlichtern. Gerade letztere sind allerdings durch die integrierte Beleuchtung aus Abschnitt 1.3 kaum zu vermeiden. Zusätzlich tritt ein starker Helligkeitsabfall an den Bildrändern auf, so dass lineare Beleuchtungsmodelle (vgl. Abschnitt 3.1.1) über das gesamte Bild, wie sie oftmals den globalen Verfahren unterliegen, nur unzureichende Ergebnisse erwarten lassen.

Das Verfahren soll in der Lage sein, Objekte in einem großen 6 DoF Lagebereich zu detektieren und zu lokalisieren. Es ist daher eine Groblagebestimmung notwendig, die alle 6 Freiheitsgrade abdeckt, sei es durch eine affine Transformation im Bild oder direkt im Raum der Lagen. Ansätze aus Tab. 2.1, die bei den ermittelten Lage DoF's nur eine eingeschränkte Anzahl ermitteln können, sind daher ungeeignet.

Ebenfalls muss die gefundene Lage akkurat genug sein, damit nachfolgende Schritte, wie z. B. ein Greifprozess erfolgreich durchgeführt werden können. Es ist daher eine Feinlageschätzung notwendig, die mittels der gefundenen Groblage eine akkurate 6 DoF Lage schätzt. Weiterhin muss dieser Schritt ein Qualitätsmaß bereitstellen, welches die Güte des Ergebnisses bewertet. Spätestens die Feinlageschätzung benötigt ein 3D-Modell mit absoluten Tiefen- oder Längenwerten in irgendeiner Form um die Lage im 3D-Raum der Kamera berechnen zu können.

Alle benötigten Informationen für den Segmentierungsschritt, die Groblagebestimmung und die Feinlagebestimmung - zusammen gefasst im Modell eines Objektes - müssen in einem *autonomen* Trainingsschritt ohne Hilfe eines Menschen oder zusätzlichen Daten (wie ein CAD-Modell) selbständig erworben werden. Im Vordergrund dürfen daher nicht vom Menschen bevorzugte und ersichtliche Objektstrukturen stehen, sondern von dem System selbständig ausgewählte. Dies führt aber in vielen Fällen dazu, dass ein allgemeiner Ansatz für eine ausreichende Robustheit viele Strukturen und eine statistische Auswertung verwendet werden muss. Ansätze wie das geometrische Hashing, die 2D-3D- oder 3D-3D-Korrespondenzverfahren auf ausgewählten Punkten, das Tracking sichtbarer Kanten mittels 3D-Kantenmodell oder das perzeptuelle Gruppieren werden daher aufgrund der Schwierigkeit bei der autonomen Auswahl der verwendeten Objektstrukturen (alleinstehend) als wenig geeignet eingestuft.

Zu guter Letzt spielt auch die Verarbeitungsgeschwindigkeit des Verfahrens eine wichtige Rolle, da dieses je nach Aufgabe in eine industrielle Prozesskette eingebunden werden soll und den Prozesszyklus nicht aufhalten darf. Je nach Komplexität der Szene soll eine Lageerkennung in unter 1 s möglich sein. Langsame Verfahren aus Tab. 2.1 mit einer negativen Geschwindigkeitsbewertung (-/--) sind daher nicht geeignet. Moderne Hardwarestrukturen der CPU und GPU haben ebenfalls einen großen Einfluss auf die Geschwindigkeit, da deren Tendenz zu immer stärkerer echter Parallelität führt. Das zu entwickelnde Verfahren sollte daher fein granular parallelisierbar sein, um das volle Potential heutiger und zukünftiger Hardware ausnutzen zu können. Lokale Verfahren haben dabei einen natürlichen Vorteil, da deren Daten zum Großteil unabhängig voneinander bearbeitet werden können. Ebenso ist z. B. eine RANSAC (FB81) basierte Ausreißer-Elimination einem iterativen Verfahren vorzuziehen, da es sich auf natürliche Weise parallelisieren lässt.

Betrachtet man die Problemstellung sowie die Randbedingungen gemeinsam und wendet das Ausschlussprinzip auf alle Verfahren aus Tab. 2.1 an, so kristallisieren sich die lokalen, regionenbasierten Verfahren in Kombination mit einer geeigneten Feinlageschätzung heraus. Im Gegensatz zu den anderen Verfahren haben sie das Potential sowohl auf robuste Weise mit Verdeckungen, komplexen Szenen und Beleuchtungsstörungen umzugehen, als auch ein autonomes Training des Modells und eine effiziente Lokalisation zu ermöglichen. Der nächste Abschnitt 4.2 stellt daher die Schwächen und Stärken dieser Ansätze sowie Verbesserungsmöglichkeiten vor. Der Fokus liegt dabei auf den ansichtsbasierten Verfahren, da die geometriebasierten 3D-Regionen genauere Tiefeninformationen voraussetzen als sie die Sensorik aus Abschnitt 1.3 zurzeit liefen kann. Allerdings werden in Abschnitt 4.3 Möglichkeiten aufgezeigt, wie diese auf einfache Weise in das neue System integriert werden können.

4.2 Lageerkennung mit lokalen Regionen

Verfahren auf der Basis von lokalen, salienten bzw. interessanten Bildregionen haben meist die gleiche Struktur. Sie wird für intensitätsbasierte 2D-Regionen ausführlich in Kapitel 3 und für Geometriebasierte 3D-Regionen in Abschnitt 2.2.2.3 behandelt. Die Struktur gliedert sich in Detektor-, Deskriptor- und Matching-Phase und hat zum Ziel, lokale regionenhafte Korrespondenzen zwischen Bildern zu finden. Meist handelt es sich dabei um ein oder mehrere aktuelle Kamerabilder (im Weiteren als *Suchansichten* bezeichnet) und einer bestehenden Bilddatenbank (im weiteren *Modellansichten* ge-

nannt) ein oder mehrerer Objekte.

Die in Abschnitt 2.1.2 beschriebenen ansichts- und regionenbasierten Verfahren unterscheiden sich vor allem in der weiteren Verarbeitung lokaler Korrespondenzen, sobald diese mit obiger Struktur etabliert worden sind. Essentiell ist dabei die Verwertung des Form-, Lage- und Orientierungsunterschieds korrespondierender Regionen, der durch die kovariante Regionenkonstruktion des Detektors ermittelt wird. Je nach Detektor bzw. zu Grunde gelegtem geometrischen Modell (vgl. Abschnitt 3.1.2) werden diese Unterschiede durch eine 6 DoF affine Transformation (AT) oder 4 DoF Ähnlichkeitstransformation (ST[1]) beschrieben.

Durch Ausnutzung dieser Informationen lassen sich nun sowohl Suchansichten segmentieren und zu den Modellansichten registrieren als auch Fehlkorrespondenzen eliminieren. Gerade letzteres ist für eine erfolgreiche Registrierung von herausragender Bedeutung, da der Korrespondenzfindungsprozess aufgrund seiner lokalen Auswertung nicht sehr robust ist. Ein Verhältnis von korrekten zu Fehlkorrespondenzen von 0.1 und darunter (Low01) ist keine Seltenheit und muss explizit bei dem Algorithmendesign berücksichtigt werden.

Eine kombinierte Segmentierung und Ausreißer-Eliminierung erfolgt meist durch Gruppierung von Korrespondenzen, deren Regionenunterschiede ähnlich sind. Am häufigsten (Low99; Low04; GL06; AAD07) werden dabei die einzelnen Transformationsparameter in einen Houghakkumulator eingetragen. Maxima ergeben dabei mit hoher Wahrscheinlichkeit einerseits eine Groblage, andererseits über die zugehörigen Korrespondenzen eine durch Ausreißer reduzierte Segmentierung. Die Vorgehensweise birgt neben diesen Vorteilen aber auch einige Schwächen:

- Der Houghakkumulator besteht meist aus den Transformationsparametern einer ST in der Bildebene, kann aber theoretisch bei entsprechender Datengröße auch für eine AT eingesetzt werden. Allerdings wird dabei vorausgesetzt, dass sich der komplette überdeckende Inhalt zwischen Such- und Modellansicht durch solch eine lineare Transformation beschreiben lässt. Wie in Abschnitt 3.1.2 gezeigt wird, ist dies aber nur für bestimmte Annahmen der Fall. Diese mögen zwar lokal Gültigkeit besitzen, global können sie aber nicht garantiert werden. Dies führt mit zunehmendem Transformationsunterschied zu immer größeren Modellierungsfehlern, die nur teilweise durch größere Akkumulatorzellen ausgeglichen werden können.

- Da die Transformation zwischen zwei Ansichten innerhalb der Bildebene betrachtet wird, muss für jedes Such- und Modellansichts-Paar eine eigene Houghauswertung erfolgen und anschließend die größten Maxima weiter verwertet werden. Dies führt dazu, dass der Algorithmus sich bei jedem Segment einer Suchansicht für eine bestimmte Modellansicht entscheiden muss und die Ergebnisse der restlichen Ansichten verworfen werden. Gleiches gilt auch für zwei Suchansichten, die einen gemeinsamen Szeneninhalt beschreiben. Dies führt zu einer geringen Ausbeute von Korrespondenzen und kann insbesondere bei einer schlechten Verhältnis von korrekten zu Fehlkorrespondenzen zu einem weniger robusten Ergebnis führen.

Gruppierte Korrespondenzen werden dann verwendet, um eine robuste globale Lage zu schätzen. Meist werden dabei nur die Mittelpunkte der Regionen herangezogen, da diese am stabilsten ermittelt werden können. Im einfachsten Fall werden sie in einem iterativen Verfahren genutzt, um eine globale AT (Low99) oder ST (Low01; Low04) zwischen Such- und bester Modellansicht zu schätzen. Dabei treten aber dieselben Probleme wie bei der Houghauswertung auf und somit gelten obige Kritikpunkte auch für diese Art des Feinfits.

(GL06; CBSF09; AAD07) ermitteln dagegen die 6 DoF Starrkörpertransformation (RT[2]) zwischen Such- und Modellansicht. Allerdings sind die Verfahren entweder auf ebene Objekte (AAD07)

[1] engl.: similarity transformation
[2] engl.: rigid transformation

oder ein einzelnes Suchbild angewiesen. Weiterhin wird in (AAD07) gezeigt, dass die Herangehensweise in (GL06; CBSF09) mit einem PnP-Algorithmus allgemein nicht sehr robust ist.

Allen Verfahren ist gemeinsam, dass sie zwar mit unterschiedlichen Regionentypen einer Klasse (2D-affin, 2D-ähnlich, 3D) arbeiten können, die Integration verschiedener Klassen, insbesondere zwischen 2D- und 3D-Regionen, selten oder gar nicht berücksichtigt wird. Ein allgemeines und erweiterbares System sollte aber alle möglichen Informationsquellen nützen können. Unterschiedliche Regionentypen und -klassen sind dafür ein erster Schritt, da sie verschiedene charakteristische Merkmale von Objekten kodieren (vgl. Abschnitt 3.2.6).

Die Lösung obiger Herausforderungen liegt in einem frühzeitigen Wechsel von den approximativen Transformationen in der Bildebene zu den korrekten zugrunde liegenden Starrkörpertransformationen im 3D-Raum der Szene. Sobald man den Übergang zwischen 2D-Bildinhalt und 3D-Szeneninhalt modellieren kann, lassen sich die Inhalte nicht nur approximativ innerhalb des Bildes registrieren sondern auch auf korrekte Weise durch eine RT der zugrunde liegenden Kamera-Koordinatensysteme. Dabei müssen dann (auf globaler Ebene) keinerlei Annahmen getroffen werden, die Modellierung ist exakt. Dies gilt sowohl bei der Modellierung des Transformationsunterschieds korrespondierender Regionen als auch bei der Beschreibung der Blickwinkeländerungen zwischen mehreren Such- und Modellansichten.

Ziel des nächsten Abschnittes ist es daher einen Ansatz auf Basis von RT's im 3D-Raum anstatt AT's oder ST's im 2D-Bildraum zu entwickeln. Dafür wird für jede Regionenklasse die Überführung des Transformationsunterschieds in den 3D-Raum modelliert, der Gruppierungsschritt auf RT's angepasst und die globale 6 DoF Feinlageschätzung überarbeitet. Ziel ist dabei, beliebig viele Such- als auch Modellansichten integrativ zu behandeln, eine globale Lage ohne weitere Annahmen zu schätzen und beliebige Regionentypen bzw. -klassen zu fusionieren. Weiterhin sollte das Verfahren aufgrund des ungünstigen Verhältnisses von korrekten zu Fehlkorrespondenzen mit möglichst wenig korrekten Korrespondenzen auskommen und Ausreißer frühzeitig eliminieren. Im besten Fall erhält man die benötigten RT's aus jeweils einer *einzigen* Regionenkorrespondenz.

4.3 6 DoF Lageerkennung mit kovarianten Regionen

Die Kovarianzeigenschaft der Detektoren erlaubt einem nicht nur die Mittelpunkte korrespondierender Regionen als Informationsquelle zu nutzen, sondern ebenfalls deren Formunterschied. In Abschnitt 3.1.2 wurde für 2D-Bildregionen die Abbildung von RT's in der Szene in damit assoziierten AT's bzw. mit Einschränkungen auch ST's in der Bildebene modelliert. Der folgende Abschnitt 4.3.1 kehrt diesen Zusammenhang um und führt beobachtete AT's bzw. ST's korrespondierender 2D-Regionen auf die zugrunde liegenden RT's zurück (vgl. Abb. 1.1). Abschnitt 4.3.2 behandelt dieselbe Problemstellung ergänzend für korrespondierende 3D-Regionen.

Damit lassen sich alle kovarianten ansichts- und geometriebasierten Regionentypen bzw. -klassen in eine gemeinsame Repräsentation, den RT's im 3D-Szenenraum überführen. Dabei reicht eine *einzelne* Korrespondenz zur Rekonstruktion einer (teilweise eingeschränkten) RT aus, die wiederum als lokale 6 DoF Lagehypothese verwendet werden kann. Sind die Such- bzw. Modellansichten untereinander durch RT's registriert, können die Lagehypothesen weiterhin in eine *universelle, ansichtsunabhängige* Form gebracht werden.

Die weiteren Verarbeitungsschritte, die Gruppierung bzw. Segmentierung und Groblageschätzung in Abschnitt 4.3.3 sowie die globale Feinlageschätzung in Abschnitt 4.3.4 verwenden unabhängig von den eingesetzten Regionendetektoren ausschließlich die universelle Repräsentation. Sie sind damit allgemein gültig, ermöglichen eine integrierte Behandlung aller verwendeten Ansichten und erlauben die Fusion aller geeigneten Regionentypen und -klassen.

Der letzte Abschnitt 4.3.5 gibt dann abschließenden einen Überblick über die gesamte Verar-

beitungskette, von der Bildakquise über die Korrespondenzfindung bis zu der robusten Ausgabe der globalen 6 DoF Lage.

4.3.1 6 DoF Lagerekonstruktion mittels einzelner 2D-Regionenkorrespondenz

Dieser Abschnitt beschäftigt sich mit der Fragestellung, welche Informationen man aus zwei lokalen 2D-Bildregionen ${}^{\mathrm{A}}\mathcal{R}[\mathbf{u}]$ und ${}^{\mathrm{B}}\mathcal{R}[\mathbf{u}]$ gewinnen kann, die sich auf dieselbe Szenenteilmenge $\mathcal{S}$ beziehen. Dabei handelt es sich um die inverse Problemstellung des geometrischen Transformationsmodells aus Abschnitt 3.1.2. Konkret wird beantwortet, ob und wie man aus der affinen Verzerrung ${}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}}$ der Grundmuster ${}^{\mathrm{B}}\mathcal{R} = {}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{R}$ und den Bezugspunkten ${}^{\mathrm{A}}\mathbf{u}_{\mathbf{p}}$ bzw. ${}^{\mathrm{B}}\mathbf{u}_{\mathbf{p}}$ die zugrunde liegende RT ${}^{\mathrm{B}}\mathcal{S} = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{S} + {}^{\mathrm{B}}\mathbf{t}$ der Szenenteilmenge rekonstruieren kann. ${}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}} = {}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}}\,{}^{\mathrm{A}}\mathbf{A}_{\mathrm{N}}^{-1}$ ergibt sich dabei aus den affinen Matrizen zur Abbildung der ellipsenförmigen Regionen ${}^{\mathrm{A}}\mathcal{R}[\mathbf{u}_{\mathbf{p}}, \mathbf{A}_{\mathrm{N}}]$ und ${}^{\mathrm{B}}\mathcal{R}[\mathbf{u}_{\mathbf{p}}, \mathbf{A}_{\mathrm{N}}]$ in die kanonische Repräsentationsform für den Deskriptor (vgl. Kap. 3). Das Verfahren kann analog auch für kreisförmige Regionen $\mathcal{R}[\mathbf{u}_{\mathbf{p}}, \mathbf{S}_{\mathrm{N}}]$ angewendet werden, allerdings können dann nur 4 der 6 DoF's der RT bestimmt werden.

Eine ähnliche Problemstellung wird in (KK08) behandelt. Allerdings müssen dort für die Szenenteilmengen sogenannte Orthobilder mit einem senkrechten Blickwinkel auf die angenommenen Ebenen zur Verfügung stehen, wohingegen bei der folgenden Ausführung beliebige Blickwinkel genügen. Ebenfalls wird im Gegensatz zu diesem Abschnitt keine vollständige analytische Lösung beschrieben.

Zuerst wird im folgenden Abschnitt 4.3.1.1 die Rekonstruktion von ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}$ behandelt, damit im anschließenden Abschnitt 4.3.1.2 der Translationsvektor ${}^{\mathrm{B}}\mathbf{t}$ bestimmt und im abschließenden Abschnitt 4.3.1.3 die Eigenschaften der daraus gewonnenen lokalen Lage- bzw. RT-Hypothese diskutiert.

4.3.1.1 Bestimmung der Rotationsmatrix

Die affine Verzerrung ${}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}}$ besteht nach Gl. 3.8 aus ${}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}} = s\,{}^{\mathrm{B}}\mathbf{K}\,{}^{\mathrm{B}}\mathbf{D}_{\mathbf{p}}\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{D}_{\mathbf{p}}^{-1}\,{}^{\mathrm{A}}\mathbf{K}^{-1}$, d.h. einer unbekannten translationsabhängigen Skalierung s, einer unbekannten rotationsabhängigen Verzerrung ${}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}$, zweier bekannter Kameramatrizen ${}^{\mathrm{A}}\mathbf{K}$ und ${}^{\mathrm{B}}\mathbf{K}$, sowie zweier bekannter Bildverzerrungsmatrizen ${}^{\mathrm{A}}\mathbf{D}_{\mathbf{p}}$ bzw. ${}^{\mathrm{B}}\mathbf{D}_{\mathbf{p}}$. Es folgt daher aus Gl. 3.7

$$s\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}} = \begin{pmatrix} sr_{11} - sm_x r_{13} & sr_{12} - sm_y r_{13} \\ sr_{21} - sm_x r_{23} & sr_{22} - sm_y r_{23} \end{pmatrix} = {}^{\mathrm{B}}\mathbf{D}_{\mathbf{p}}^{-1}\,{}^{\mathrm{B}}\mathbf{K}^{-1}\,{}^{\mathrm{B}}\mathbf{A}'_{\mathrm{A}}\,{}^{\mathrm{A}}\mathbf{K}\,{}^{\mathrm{A}}\mathbf{D}_{\mathbf{p}} \tag{4.1}$$

Störend sind dabei die Ebenenparameter m_x und m_y zum Zeitpunkt A. Sind diese allerdings bekannt, lässt sich das Problem vereinfachen. Sei ${}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}$ eine Rotationsmatrix, die ${}^{\mathrm{A}}\mathcal{S}$ so rotiert, dass die Normale der (approximativen) Ebene von ${}^{\mathrm{N}}\mathcal{S} = {}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\mathcal{S}$ parallel zur optischen Achse der Kamera zum Zeitpunkt A steht. Wie sich durch Lemma B.2.1 leicht zeigen lässt, ist eine mögliche Form für ${}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}$:

$$\begin{pmatrix} -r^{-1} & 0 & m_x r^{-1} \\ -m_x m_y r^{-1} d^{-1} & r d^{-1} & -m_y r^{-1} d^{-1} \\ -m_x d^{-1} & -m_y d^{-1} & -d^{-1} \end{pmatrix} \tag{4.2}$$

mit $r = \sqrt{m_x^2 + 1}$ und $d = \sqrt{m_y^2 + r^2}$. Für die zu bestimmende Rotationsmatrix gilt dann ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}} = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{N}}\,{}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}$. Nach Lemma B.2.2 verhält sich die affine Verzerrung ${}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}} = {}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}}\,{}^{\mathrm{N}}\mathbf{A}_{\mathrm{A}}$ dazu analog, wobei ${}^{\mathrm{N}}\mathbf{A}_{\mathrm{A}}$ durch ${}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}$ bekannt ist. Da in der Normalen-Darstellung N die Ebenenparameter ${}^{\mathrm{N}}m_x = {}^{\mathrm{N}}m_y = 0$ sind, ergibt sich die einfachere Form

$$s\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}} = \begin{pmatrix} sr_{11} & sr_{12} \\ sr_{21} & sr_{22} \end{pmatrix} = s\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{A}}\,{}^{\mathrm{N}}\mathbf{A}_{\mathrm{A}}^{-1} \tag{4.3}$$

Ohne Beschränkung der Allgemeinheit lässt sich ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{N}} = \mathbf{R}_z[\gamma]\mathbf{R}_x[\alpha]\mathbf{R}_y[\beta] = \mathbf{R}_z[\gamma]\mathbf{R}_{xy}[\alpha,\beta]$ in elementare Rotationsmatrizen um die Koordinatenachsen aufspalten und nach Lemma B.2.2 korrespondierend dazu ${}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}} = \mathbf{A}_z\mathbf{A}_{xy}$. α, β und γ werden auch als Euler-Winkel bezeichnet. Nach Lemma B.2.3 folgt durch $\mathbf{R}_z$, dass $\mathbf{A}_z \in \mathcal{SO}_2$ ist, d.h. $\mathbf{A}_z$ ist eine Rotationsmatrix. Aus der Form

$$\mathbf{R}_{xy}(\alpha,\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ \sin\alpha\sin\beta & \cos\alpha & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \sin\alpha & \cos\alpha\cos\beta \end{pmatrix} \tag{4.4}$$

lässt sich ablesen, das $r_{12} = 0$ und daher $\mathbf{A}_{xy}$ eine untere Dreiecksmatrix sein muss. Die Zerlegung ${}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}} = \mathbf{A}_z\mathbf{A}_{xy}$ lässt sich daher eindeutig und numerisch stabil mittels einer QL[3]-Zerlegung realisieren:

$$s\,{}^{\mathrm{B}}\mathbf{A}_{\mathrm{N}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \mathbf{A}_z s\mathbf{A}_{xy} = \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}\begin{pmatrix} e & 0 \\ f & g \end{pmatrix} \tag{4.5}$$

mit

$$\begin{aligned} g &= \sqrt{b^2+d^2} \\ e &= (ad-bc)g^{-1} \\ f &= (ab+cd)g^{-1} \\ \cos\gamma &= dg^{-1} \\ \sin\gamma &= -bg^{-1} \end{aligned}$$

$\mathbf{R}_z[\cos\gamma,\sin\gamma]$ ist damit eindeutig bestimmt. Unter Ausnutzung der Struktur $e = s\cos\beta$, $f = s\sin\alpha\sin\beta$ und $g = s\cos\alpha$ lässt sich durch Quadrieren, Substitution und Verwendung von $\cos^2+\sin^2 = 1$ der Skalierungsparameter s bestimmen:

$$s_\pm = \sqrt{\frac{e^2+g^2+f^2 \pm \sqrt{(e^2+g^2+f^2)^2-4e^2g^2}}{2}} \tag{4.6}$$

Mehrdeutigkeiten durch das mehrfache Quadrieren lassen sich leicht auflösen, da nur eine positive Skalierung $s > 0$ physikalisch Sinn macht und in Lemma B.2.4 gezeigt wird, dass nur die Lösung $s_+ = s$ zu einer mathematisch korrekten Rotationsmatrix $\mathbf{R}_{xy}[\alpha,\beta]$ führt. Es folgt $\cos\alpha = fs^{-1}$, $\cos\beta = fs^{-1}$, $\sin^\pm_\alpha = \pm\sqrt{1-\cos\alpha}$ und $\sin^\pm_\beta = \pm\sqrt{1-\cos\beta}$. $\mathbf{R}_{yx}$ lässt sich allerdings nicht eindeutig bestimmen, da aus f nur das Produkt der Sinus-Vorzeichen ermittelt werden kann. Abhängig von dem Vorzeichen von f ergeben sich mit $\alpha^\pm = \mathrm{arctan2}(\sin^\pm_\alpha,\cos\alpha)$ und $\beta^\pm = \mathrm{arctan2}(\sin^\pm_\beta,\cos\beta)$ daher zwei Lösungen $\mathbf{R}^+_{yx}$ und $\mathbf{R}^-_{yx}$:

$$\begin{cases} \mathbf{R}^+_{yx}[\alpha^+,\beta^+],\ \mathbf{R}^-_{yx}[\alpha^-,\beta^-] & f \geq 0; \\ \mathbf{R}^+_{yx}[\alpha^+,\beta^-],\ \mathbf{R}^-_{yx}[\alpha^-,\beta^+] & f < 0. \end{cases} \tag{4.7}$$

Aus diesem Grund lässt sich die Rotationsmatrix ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}} = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{N}}\,{}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}} = \mathbf{R}_z\mathbf{R}_{xy}\,{}^{\mathrm{N}}\mathbf{R}_{\mathrm{A}}$ nicht eindeutig angeben, sondern es existieren in Abhängigkeit von $\mathbf{R}^+_{yx}$ und $\mathbf{R}^-_{yx}$ zwei Lösungen ${}^{\mathrm{B}}\mathbf{R}^+_{\mathrm{A}}$ und ${}^{\mathrm{B}}\mathbf{R}^-_{\mathrm{A}}$. Diese unterscheiden sich durch eine Spiegelung der Verkippung an der Ebene von ${}^{\mathrm{A}}\mathcal{S}$ und haben in B dasselbe affin verzerrte Abbild.

[3] Bekannter ist die QR-Zerlegung, bei der eine Matrix $\mathbf{A} = \mathbf{QR}$ in eine orthogonale Matrix $\mathbf{Q}$ und eine obere Dreiecksmatrix $\mathbf{R}$ zerlegt wird. Mittels Givens-Rotationen lässt sich sowohl die QR- als auch die QL- Zerlegung analytisch und numerisch stabil angeben. Hat $\mathbf{A}$ vollen Rang, existieren die Zerlegungen immer, fordert man zusätzlich $|\mathbf{Q}| = 1$, sind beide Zerlegungen eindeutig. Vgl. hierzu auch (Bro99).

4.3.1.2 Bestimmung des Translationsvektors

Der gesuchte Translationsvektor ${}^{\mathrm{B}}\mathbf{t}$ ist die Verschiebung der beiden Bezugspunkte ${}^{\mathrm{A}}\mathbf{p} = \mathbf{h}^{-1}({}^{\mathrm{A}}\mathbf{u_p}, {}^{\mathrm{A}}p_z)$ und ${}^{\mathrm{B}}\mathbf{p} = \mathbf{h}^{-1}({}^{\mathrm{B}}\mathbf{u_p}, {}^{\mathrm{B}}p_z)$ im Koordinatensystem B. Gegeben sind allerdings nur die Bildpunkte ${}^{\mathrm{A}}\mathbf{u_p}$ und ${}^{\mathrm{B}}\mathbf{u_p}$, die mittels des inversen Kameramodells $\mathbf{h}^{-1}(\cdot)$ in die jeweiligen Koordinatensysteme A bzw. B zurück projiziert werden müssen. Dazu ist zumindest einer der Tiefenwerte ${}^{\mathrm{A}}p_z$ bzw. ${}^{\mathrm{B}}p_z$ notwendig, der andere lässt sich dann über den Skalierungsparameter der Rotationenrekonstruktion $s = {}^{\mathrm{A}}p_z/{}^{\mathrm{B}}p_z$ (vgl. Formel 3.8) aus dem gegebenen Wert ermitteln. ${}^{\mathrm{A}}\mathbf{p}$ muss nun noch unter Zuhilfenahme der schon ermittelten Rotationsmatrix ${}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}$ in dem Koordinatensystem B dargestellt und von ${}^{\mathrm{B}}\mathbf{p}$ abgezogen werden:

$$\begin{aligned} {}^{\mathrm{B}}\mathbf{t}^{+} &= {}^{\mathrm{B}}\mathbf{p} - {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}^{+}\,{}^{\mathrm{A}}\mathbf{p} \\ {}^{\mathrm{B}}\mathbf{t}^{-} &= {}^{\mathrm{B}}\mathbf{p} - {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}^{-}\,{}^{\mathrm{A}}\mathbf{p} \end{aligned} \tag{4.8}$$

4.3.1.3 Eigenschaften der Lagerekonstruktion

Während mittels des geometrischen Modells $\mathbf{geo}(\cdot)$ unter gewissen Annahmen eine *eindeutige* Abbildung $\mathbf{A} = \mathbf{G}(\mathbf{R}, m_x, m_y)$ (genaue Definition siehe Anhang B.2) angegeben werden kann, die einer Rotation $\mathbf{R}$ in der Szene ihre induzierte affine Transformation $\mathbf{A}$ im Bild zuordnet, kann die Umkehrung nur durch eine zweideutige Relation beschrieben werden.

Dies begründet sich durch die approximative Modellierung der perspektivischen Abbildung als skalierte orthographische Projektion (SOP), die zwar einerseits erst eine lineare Beschreibung der Bildtransformationen ermöglicht andererseits aber keine eindeutige Rekonstruktion der Verkippung $\mathbf{R}_{xy}$ in Gl. 4.4 aus der Kameraebene hinaus zulässt. Handelt es sich bei der Rotation allerdings nur um eine Drehung $\mathbf{R_z}$ in der Kameraebene, gilt $\mathbf{A} = \mathbf{S} \in \mathcal{T}_{S,n}^{+}$ sowie $\mathbf{R}_{xy} = \mathbf{I}_{3\times 3}$ und eine eindeutige Rekonstruktion ist weiterhin möglich.

Zur Rekonstruktion werden einerseits die Formparameter $\mathbf{A}_{\mathrm{N}}$ und $\mathbf{u_p}$ der Region in den beiden Bildern A und B benötigt, als auch der Tiefenwert p_z sowie die Ebenenparameter m_x und m_y in Bild A. Die Ebenenparameter sollten sinnvollerweise genau in dem Bereich $\mathcal{S}$ der Szene geschätzt werden, auf den sich die verwendete Region bezieht. Es gilt, dass die Ebenenparameter nur zur Rekonstruktion der Verkippung $\mathbf{R}_{xy}$ benötigt werden, bei einer ausschließlichen Betrachtung von $\mathbf{R_z}$ als Rotation sind sie überflüssig.

Von der Genauigkeit der rekonstruierten RT darf man sich nicht zu viel erwarten, da mehrere ungünstige Faktoren aufeinander treffen. Einerseits werden in 3.1.2 mehrere Approximationen getroffen, die in der Realität nur lokal eine eingeschränkte Gültigkeit besitzen und deren Fehler mit zunehmender Regionengröße und Transformationsunterschied an Bedeutung gewinnen. Andererseits leitet sich die rekonstruierte Tiefe ${}^{B}p_z = {}^{A}p_z s^{-1}$ indirekt proportional aus der tatsächlich ermittelten Skalierung s der korrespondierenden Regionen ab. Selbst kleine Fehler in s wirken sich daher stark auf ${}^{B}p_z$ aus. Ebenfalls wird der Verkippungswinkel vereinfacht betrachtet aus dem tatsächlich beobachteten Cosinus gewonnen, so dass aufgrund des flachen Verlaufs von $\cos(\cdot)$ um 0° schon kleinste Fehler eine große Auswirkung auf den Winkel haben. Die Skalierung sowie der Cosinus sollten daher möglichst exakt bestimmt werden, wobei gerade der Einfluss von additiven Fehlern bei kleinen, lokalen Regionen an Bedeutung gewinnt.

Zum Abschluss sei gesagt, dass die korrekte 6 DoF Rekonstruktion der RT über eine AT in der Bildebene im Gegensatz zu einer eingeschränkten 4 DoF Rekonstruktion der RT ohne Verkippung mittels einer ST deutlich aufwändiger ist. Weiterhin ist zu erwarten, dass die vollständige Rotation weniger robust ermittelt werden kann, da sie einerseits auf die zusätzlich bestimmbaren Ebenenparameter m_x und m_y angewiesen ist und andererseits die Cosinus-Problematik nicht umgangen werden kann.

4.3.2 6 DoF Lagerekonstruktion mittels einzelner 3D-Regionenkorrespondenz

Obwohl die Klasse der geometriebasierten 3D-Regionen aus Abschnitt 2.2.2.3 in dieser Arbeit aufgrund der eingeschränkten Genauigkeit der Tiefeninformationen mittels Triangulation einer Stereokamera (vgl. Abschnitt 1.3) nicht weiter verfolgt werden, sollen sie doch als Ergänzung in das Konzept mit aufgenommen werden.

Geometriebasierte 3D-Regionen $\mathcal{R}_{3D}$ werden im 3D-Raum auf Tiefenbildern, Punktwolken oder Gitternetzen konstruiert. Ihre kovariante Konstruktion bezieht sich schon auf eine RT im 3D-Szenenraum und muss nicht mehr dorthin überführt werden. Daher beschreiben 3D-Regionen direkt die auf sie bezogene Szenenteilmenge $\mathcal{S} = \mathcal{R}_{3D}$ und eine Unterscheidung beider Mengen muss nicht wie im ansichtsbasierten Fall durchgeführt werden. 3D-Regionen $\mathcal{R}_{3D}[\mathbf{p}, \mathbf{R}_\mathrm{N}]$ sind durch einen Mittel- bzw. Ankerpunkt $\mathbf{p} \in \mathbb{R}^3$ und eine Rotationsmatrix $\mathbf{R}_\mathrm{N} \in \mathcal{SO}_3$ vollständig definiert. Beide Parameter dienen wie im ansichtsbasierten Fall der 2D-Regionen dazu, die 3D-Regionen in eine kanonische Form N

$$^{\mathrm{N}}\mathcal{R}_{3D} = \mathbf{R}_\mathrm{N}{}^{-1}(\mathcal{R}_{3D}[\mathbf{p}, \mathbf{R}_\mathrm{N}] - \mathbf{p}) \tag{4.9}$$

zu überführen, in der eine geometrische invariante Deskription möglich ist. Wie in Abschnitt 2.2.2.3 beschrieben, wird N meist durch eine Ausrichtung an einer Oberflächennormalen $\mathbf{n}$ am Punkt $\mathbf{p}$ sowie einem definierten Winkel α um $\mathbf{n}$ normiert.

Beobachtet man zwei korrespondierende 3D-Regionen $^{\mathrm{A}}\mathcal{R}_{3D}[\mathbf{p}, \mathbf{R}_\mathrm{N}]$ und $^{\mathrm{B}}\mathcal{R}_{3D}[\mathbf{p}, \mathbf{R}_\mathrm{N}]$ zu verschiedenen Zeitpunkten bzw. in verschiedenen Bezugssystemen A und B, so lässt sich die gesuchte RT $^{\mathrm{B}}\mathcal{R}_{3D} = {}^{\mathrm{B}}\mathbf{R}_\mathrm{A}\,{}^{\mathrm{A}}\mathcal{R}_{3D} + {}^{\mathrm{B}}\mathbf{t}$ zur Überführung leicht ermitteln:

$$^{\mathrm{B}}\mathbf{R}_\mathrm{A} = {}^{\mathrm{B}}\mathbf{R}_\mathrm{N}\,{}^{\mathrm{A}}\mathbf{R}_\mathrm{N}^{-1} \tag{4.10}$$

$$^{\mathrm{B}}\mathbf{t} = {}^{\mathrm{B}}\mathbf{p} - {}^{\mathrm{B}}\mathbf{R}_\mathrm{A}\,{}^{\mathrm{A}}\mathbf{p} \tag{4.11}$$

Im Gegensatz zur ansichtsbasierten Lösung ergeben sich keine Mehrdeutigkeiten in der rekonstruierten RT, sofern die kovariante Konstruktion der 3D-Regionen eindeutig ist. Ebenfalls sind keinerlei nichtlineare Transformationen nötig, die Genauigkeit bzw. Güte hängt daher direkt von der Qualität der kovarianten Parameter $\mathbf{R}_\mathrm{N}$ und $\mathbf{p}$ ab. Den größten Einfluss dürfte wohl die Normierung von α und die Normalenrekonstruktion $\mathbf{n}$ besitzen. Allerdings sollte die Qualität der lokal ermittelten RT bei entsprechenden Ausgangsdaten durch die direktere Konstruktion um mind. eine Größenordnung besser sein als im ansichtsbasierten Fall.

4.3.3 Gruppierung lokaler Hypothesen

Im weiteren Verlauf wird für eine Lage bzw. RT $({}^{\mathrm{A}}\mathbf{R}_\mathrm{B}, {}^{\mathrm{A}}\mathbf{t})$ die Repräsentation mittels einer homogenen Matrix $^{\mathrm{A}}\check{\mathbf{H}}_\mathrm{B}$ verwendet, um eine kompakte und übersichtliche schriftliche Darstellung zu ermöglichen (siehe dafür auch B.1).

Die Modelldatenbank eines einzelnen Objekts besteht aus einer Menge $\mathcal{M}$ von m Modellansichten $\mathbf{m}_j(\cdot),\ j = 1, \ldots, m$. Jede Modellansicht $\mathbf{m}_j$ korrespondiert mit einem Koordinatensystem $\mathrm{M_j}$, welches die Lage der Kamera zum Zeitpunkt der Aufnahme repräsentiert. Alle Modellansichten sind bzgl. eines gemeinsamen Objekt-Koordinatensystems O registriert, d. h. alle $^{\mathrm{Mj}}\check{\mathbf{H}}_\mathrm{O}$ sind bekannt.

Die Menge der Suchansichten $\mathcal{C}$ besteht aus c aktuellen Kameraansichten $\mathbf{c}_i(\cdot),\ i = 1, \ldots, c$ der Szene. Jede Suchansicht $\mathbf{c}_i$ korrespondiert mit einer aktuellen Kameralage $\mathrm{C_i}$, die alle bzgl. eines gemeinsamen Welt-Koordinatensystems W über $^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}$ miteinander kalibriert sind.

Eine *universelle, ansichtsunabhängige* Lage lässt sich nun zwischen Welt- und Objekt-Koordinatensystem definieren. Sei $^{\mathrm{C_i}}\check{\mathbf{H}}_{\mathrm{M_j}}$ die lokale 6 DoF Lage einer Regionenkorrespondenz zwischen der Suchansicht $\mathbf{c}_i$ und der Modellansicht $\mathbf{m}_j$. Die universelle lokale Lagerepräsentation erhält man durch:

$$^{\mathrm{W}}\check{\mathbf{H}}_\mathrm{O} = {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}\,{}^{\mathrm{C_i}}\check{\mathbf{H}}_{\mathrm{M_j}}\,{}^{\mathrm{M_j}}\check{\mathbf{H}}_\mathrm{O} \tag{4.12}$$

Alle Korrespondenzen zwischen $\mathcal{C}$ und $\mathcal{M}$ aller Regionentypen werden in diese Repräsentation überführt und stehen im weiteren Verlauf ansichtsunabhängig zur Verfügung.

Jede lokale Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ ist eine vollständige 6 DoF Lageschätzung eines Objekts im Welt-Koordinatensystem W. Sie wird allerdings nur aus einem kleinen, von der Region umschlossenen Objekt- bzw. Szenenausschnitt geschätzt und ist daher von limitierter Genauigkeit (vgl. Abschnitt 4.3.1.3). Weiterhin sind in der Praxis viele Korrespondenzen fehlerhaft und es lässt sich daher nicht ohne weiteres entscheiden, welche der lokalen Lage zur Lokalisierung herangezogen werden soll.

Während allerdings lokale Lagen korrekter Korrespondenzen nur in einem kleinen Bereich um die wahre Lage streuen, verteilen sich dagegen Lagen falscher Korrespondenzen einigermaßen gleichförmig über den gesamten Raum der 6 DoF Lagen. Gruppen bzw. Cluster lokaler Lagen sind daher ein starkes Indiz für ein Objekt an der entsprechenden Position und der entsprechenden Orientierung.

Viele Verfahren nutzen zur Gruppierung die in 2.1.2.2 beschriebene Hough-basierte Auswertung, allerdings nur in der Bildebene mit 4 DoF's. Dazu partitioniert der Houghakkumulator den 4 DoF Raum in einzelne Houghzellen und trägt dort die lokalen Lagen ein. Zellen mit vielen Einträgen repräsentieren die gesuchten Cluster und werden weiter verarbeitet. Diese Herangehensweise lässt sich allerdings nicht ohne weiteres auf 6 DoF erweitern, da die Anzahl der Zellen exponential mit der Anzahl der DoF's steigt. Der Houghakkumulator lässt sich daher mit der heutigen Hardware weder ausreichend fein partitioniert im Hauptspeicher darstellen noch effizient auswerten. Weiterhin kann innerhalb der Bildebene der zu partitionierende Raum bzw. die zu suchende Lage eingegrenzt werden, was in der universellen Darstellung weder erwünscht, noch ohne Kenntnis der einzelnen Lagen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}$ der Suchansichten möglich ist.

In dieser Arbeit werden daher in Abschnitt 4.3.3.2 die einzelnen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ ohne explizite Raumpartitionierung oder Lageeinschränkungen direkt im Raum der 6 DoF Lagen gruppiert bzw. geclustert. Dazu ist allerdings ein im nächsten Abschnitt definiertes Distanzmaß nötig, um die Differenz lokaler Lagen messen zu können.

4.3.3.1 Distanzmaß im 6D-Raum der Lagen

Gegeben sind zwei Lagen A und B in einem gemeinsamen Koordinatensystem X, die durch ${}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{A}}$ und ${}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{B}}$ gegeben sind. Der Unterschied beider Lagen lässt sich durch die RT ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}} = {}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{A}}^{-1}\,{}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{B}}$ ausdrücken. Ziel dieses Abschnittes ist es, eine Funktion

$$d : \mathcal{X} \rightarrow \mathbb{R}_{0+}$$

zu finden, die eine Lage bzw. Lagedifferenz ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}$ aus der (allgemeinen) Menge der 6 DoF Lagen bzw. RT's $\mathcal{X}$ auf eine reelle Größe $d({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}})$ abbildet, genau dann 0 ist, wenn ${}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{A}} = {}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{B}}$ gilt und ansonsten bei zunehmenden Unterschied beider Lagen größer wird. Weiterhin muss das Maß symmetrisch sein, d. h. $d({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}) = d({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}^{-1}) = d({}^{\mathrm{B}}\check{\mathbf{H}}_{\mathrm{A}})$.

Es ist einfacher und physikalisch sinnvoll die Rotations- und Translationsdifferenz von ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}$ zuerst getrennt zu behandeln. Zur besseren Darstellung werden die Lagen A bzw. B in der Rotationsmatrizen-Schreibweise $({}^{\mathrm{X}}\mathbf{R}_{\mathrm{A}}, {}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}})$ bzw. $({}^{\mathrm{X}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}})$ und der Quaternionen-Schreibweise $({}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{A}}, {}^{\mathrm{X}}\dot{\mathbf{t}}_{\mathrm{A}})$ bzw. $({}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{X}}\dot{\mathbf{t}}_{\mathrm{B}})$ benötigt (siehe Anhang B.1). Der Lageunterschied ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}$ ergibt sich damit zu $({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{t}) = ({}^{\mathrm{X}}\mathbf{R}_{\mathrm{A}}^{\mathrm{T}}\,{}^{\mathrm{X}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{X}}\mathbf{R}_{\mathrm{A}}^{\mathrm{T}}({}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}} - {}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}}))$, der Orientierungsunterschied als Quaternion zu ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}} = {}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{A}}^{*}\,{}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{B}}$.

Die Distanz $d_t : \mathbb{R}^3 \rightarrow \mathbb{R}_{0+}$ der Translation kann über die euklidische Distanz bzw. die Länge des Verschiebungsvektors ${}^{\mathrm{A}}\mathbf{t} = {}^{\mathrm{X}}\mathbf{R}_{\mathrm{A}}^{\mathrm{T}}({}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}} - {}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}})$ zwischen beiden Lagen angegeben werden:

$$d_t({}^{\mathrm{A}}\mathbf{t}) = |{}^{\mathrm{X}}\mathbf{R}_{\mathrm{A}}^{\mathrm{T}}({}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}} - {}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}})| = |{}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}} - {}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}}| = |{}^{\mathrm{X}}\mathbf{t}_{\mathrm{A}} - {}^{\mathrm{X}}\mathbf{t}_{\mathrm{B}}| \tag{4.13}$$

Offensichtlich erfüllt $d_t(\cdot)$ sowohl die Symmetrieanforderung als auch $d_t(\cdot) = 0$, genau dann, wenn beide Lagen identisch sind. Weiterhin wird $d_t(\cdot)$ mit zunehmender Verschiebung größer.

Zur Messung des Orientierungsunterschieds der Lagen A und B wird das Quaternion ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}} = {}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{A}}^{*}\,{}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{B}}$ verwendet. Jede Rotation im $\mathbb{R}^3$ lässt sich mittels einer Achse und einem Winkel α repräsentieren. Der Unterschied ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}$ der Orientierungen lässt sich daher auf natürliche Weise durch $\alpha({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) \in [0,\ldots,\pi]$ beschreiben. Der Realteil $\mathrm{re}({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}})$ beinhaltet dabei den $\cos\frac{\alpha}{2}$ des gesuchten Winkels. Es ist allerdings zu beachten, dass $-{}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}$ dieselbe Rotation repräsentiert. Da immer der kleinstmögliche Winkel im Bereich $[0,\ldots,\pi]$ gesucht wird, ergibt sich $\alpha({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) = 2\arccos|\mathrm{re}({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}|)$. Der Realteil von ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}$ lässt sich leicht berechnen, da er dem Skalarprodukt $w_a w_b + x_a x_b + y_a y_b + z_a z_b = \mathrm{re}({}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{A}}^{*}\,{}^{\mathrm{X}}\dot{\mathbf{q}}_{\mathrm{B}}) = \mathrm{re}({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}})$ der beiden Ausgangsquaternionen ${}^{\mathrm{X}}\dot{\mathbf{a}}_{\mathrm{A}} = (w_a, x_a, y_a, z_a)^{\mathrm{T}}$ und ${}^{\mathrm{X}}\dot{\mathbf{a}}_{\mathrm{B}} = (w_b, x_b, y_b, z_b)^{\mathrm{T}}$ in Vektorschreibweise entspricht. Daher gilt auch die Symmetrie $\alpha({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) = \alpha({}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{A}})$.

Das Distanzmaß $d_\alpha : \mathbb{H}_1 \rightarrow \mathbb{R}_{0+}$ der Orientierung lässt sich über den Winkel wie folgt definieren:

$$d_\alpha({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) = \alpha({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) = 2\arccos|\mathrm{re}({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}})| = 2\arccos|w_a w_b + x_a x_b + y_a y_b + z_a z_b| \tag{4.14}$$

Es ist wiederum symmetrisch und erfüllt ebenfalls $d_\alpha(\cdot) = 0$ falls beide Orientierungen identisch sind. Allerdings bildet es den Orientierungsunterschied nur auf den beschränkten Bereich $[0,\pi]$ ab und kann daher nicht wie $d_t(\cdot)$ über alle Grenzen steigen. Dieser Effekt ist aber durchaus gewollt, da maximal unterschiedliche Orientierungen mit $d_\alpha(\cdot) = \pi$ im Gegensatz zu einer unendlichen Verschiebung in der Praxis durchaus beobachtet werden können.

Auf Basis der zwei separaten Distanzmaße kann nun das Gesamtmaß $d(\cdot)$ definiert werden. Im einfachsten Fall können diese über $d(\cdot) = a d_t(\cdot) + b d_\alpha(\cdot)$ linear miteinander verknüpft werden. Die Frage ist allerdings, wie man die beiden Gewichtungsfaktoren a und b festlegt. Oder anders ausgedrückt, welche Bedeutung hat ein Translationsunterschied von z. B. 1 mm bzgl. eines Orientierungsunterschieds von 1°. Dies lässt sich ohne weiteres Wissen über die Verwendung des Distanzmaßes nicht vernünftig beantworten.

Eine weitere Möglichkeit ist die Verwendung der Mahalanobisdistanz. Sei $\mathbf{d}(\cdot) = (d_t(\cdot), d_\alpha(\cdot))^{\mathrm{T}}$ ein Vektor, der beide separaten Maße enthält. Die Mahalanobisdistanz zur Bestimmung des Lageunterschieds ist dann definiert durch:

$$d_{\mathbf{P}}({}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{A}}, {}^{\mathrm{X}}\check{\mathbf{H}}_{\mathrm{B}}) = d({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}})_{\mathbf{P}} = \sqrt{\mathbf{d}({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}})^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{d}({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}})} \tag{4.15}$$

Die Gewichtungsmatrix $\mathbf{P}$ modelliert dabei nicht nur eine lineare Gewichtung sondern auch die Korrelation zwischen den beiden Maßen $d_t(\cdot)$ und $d_\alpha(\cdot)$. Die Matrix ist kontextabhängig und muss für jede Problemstellung bestimmt werden. Dazu wird normalerweise eine Statistik über sehr viele beobachtete Lageunterschiede herangezogen. Die Kovarianzmatrix der Daten entspricht dann der Gewichtungsmatrix.

Die Kovarianzmatrix sorgt dafür, dass beide separaten Distanzen in Abhängigkeit ihres durchschnittlichen Fehleraufkommens gleichberechtigt in der endgültigen Distanz repräsentiert sind. Diese Idee könnte auch weiter verfolgt werden, da, wie in Abschnitt 4.3.1.3 diskutiert, z. B. der durchschnittliche Translationsfehler entlang der optischen Achse größer ist als parallel zur Bildebene. Dazu müsste sich die Distanzberechnung aber immer auf das gleiche Kamera-Koordinatensystem beziehen, was in der universellen Darstellung nicht möglich ist. Alternativ müsste man zusätzlich zu den Lagen Unsicherheiten mit propagieren und auf eine kompliziertere Distanzberechnung ausweichen. Beides wird in dieser Arbeit nicht weiter verfolgt.

4.3.3.2 Clustern im 6D-Raum der Lagen

Geben ist eine Menge $\mathcal{H} \subset \mathcal{X}$ lokaler 6 DoF Lagen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i} \in \mathcal{H}$ in der universellen Darstellung. Ziel dieses Abschnitts ist es, mögliche Cluster innerhalb von $\mathcal{H}$ zu finden und alle zugehörigen Lagen auszugeben. Ein Cluster $\mathcal{G}$ beinhaltet dabei Lagen die zueinander ähnlicher sind als zu Lagen außerhalb des Clusters. Als Ähnlichkeitsmaß wird die in Formel 4.15 definierte Distanzfunktion $d_{\mathbf{P}}$ verwendet.

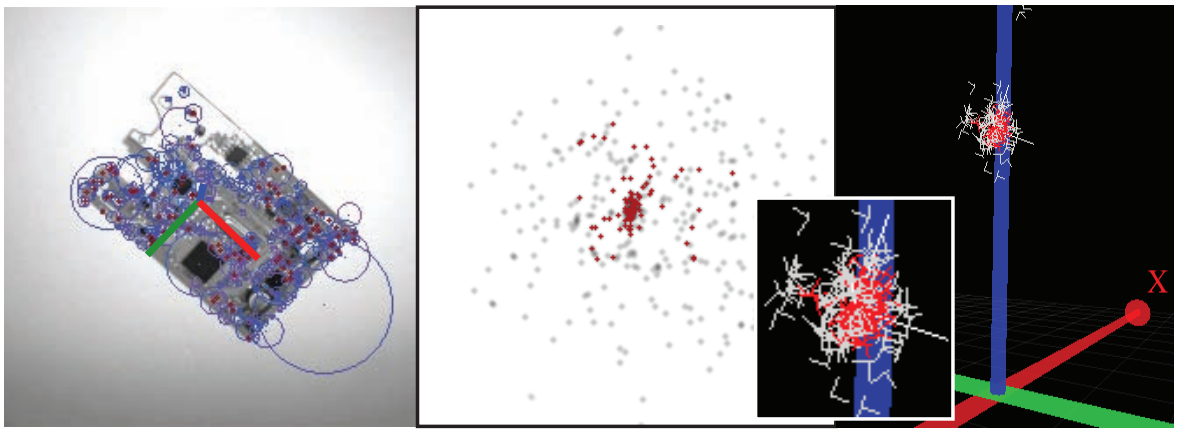

Abbildung 4.1: Links sind die Regionen einer Suchansicht dargestellt, in der Mitte, die nach der Korrespondenzfindung abgeleiteten lokalen Lagen, deren Ursprünge zurück in die Suchansicht projiziert wurden. Im Idealfall sollten die lokalen Lagen dem in der Suchansicht dargestellten Objekt-Koordinatensystem entsprechen. Rechts sind dieselben lokalen Lagen als (kleine) Koordinatensysteme relativ zu dem Kamera-Koordinatensystem der Suchansicht in einer 3D-Visualisierung abgebildet. Die große blaue Achse entspricht dabei der optischen Achse der Kamera. Das gefundene Cluster wurde mit rot markiert, ebenfalls die Zentren der dazugehörigen Suchregionen. Die 3D-Darstellung beinhaltet zwar alle 6 DoF, allerdings lässt die Informationsdichte keine sinnvolle Interpretation zu. In dieser Arbeit wird daher die übersichtlichere mittlere Darstellung ohne dargestellte Orientierungsinformationen zur Visualisierung verwendet. Dort sind aber auch nur 2 der 6 zum Clustern verwendeten DoF's ersichtlich, daher die in dieser Darstellung nicht immer ganz offensichtliche Gruppierung.

Jedes Cluster repräsentiert eine potentielle Objektinstanz mit entsprechender Lage. Allerdings ist die Anzahl beobachteter Objekte und damit auch die Anzahl k potentieller Cluster unbekannt. Partitionierungsverfahren wie *Mean Shift* (Che95) oder *k-Means* (KMN+02) benötigen allerdings k und sind daher zum Gruppieren lokaler Lagen unbrauchbar. Ebenfalls sind sie iterativ und daher schwer zu parallelisieren. Behelfen kann man sich zwar, indem die Algorithmen mit allen möglichen Werten für k angewendet werden und dann das beste Ergebnis zurückgegeben wird. Dies wird aber aus Effizienzgründen in dieser Arbeit abgelehnt. Wahrscheinlichkeitsbasierte Ansätze, wie der *EM-Algorithmus*[4] (DHS00) unterliegen derselben Argumentation und werden daher ebenfalls nicht weiter verfolgt.

Eine andere Möglichkeit besteht durch hierarchische Clusterverfahren. Sie benötigen kein Wissen über k sondern fassen in einem bottom-up Ansatz ähnliche Lagen zusammen (agglomeratives Clustern) bzw. zerteilt $\mathcal{H}$ in einem top-down Ansatz (diversives Clustern). In beiden Fällen benötigt man eine Bewertung für Cluster, um die Zusammenfassung bzw. Zerteilung rechtzeitig zu stoppen. Sie sind im Grunde für die Aufgabenstellung geeignet, allerdings durch ihre iterative Struktur schwierig zu parallelisieren und wenig effizient.

Man kann das Clusteringproblem allerdings vereinfachen, in dem man die Ausprägung der Menge $\mathcal{H}$ (vgl. Abb. 4.1) berücksichtigt. $\mathcal{H}$ besteht aus einer Menge von zufällig verteilten Lagen fehlerhafter Korrespondenzen, sowie Lagen korrekter Korrespondenzen, die sich in einem durch Rauschen bestimmten Radius um die wahren Lagen von Objektinstanzen gruppieren. Es ist daher weder notwendig noch sinnvoll alle Lagen zu clustern, sondern geschickter, nach einer großen Anzahl von Lagen innerhalb einer gewissen Distanz τ zu suchen. Man kann weiterhin schon aufgrund der An-

[4] engl.: expectation maximization algorithm

zahl mit einer hohen Wahrscheinlichkeit annehmen, dass sich eine der Lagen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i} \in \mathcal{H}$ nahe am Clusterzentrum, d. h. der wahren Objektlage befindet. Es reicht daher aus, alle ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i} \in \mathcal{H}$ als potentielles Clusterzentrum zu überprüfen. Dazu wird eine Bewertungsfunktion $\chi_\tau : \mathcal{X} \times \mathcal{X} \rightarrow [0,\ldots,1]$ eingeführt, die angibt ob eine Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j} \in \mathcal{H}$ dem Cluster $\mathcal{G}_i$ mit Zentrum ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i} \in \mathcal{H}$ angehört:

$$\chi_\tau({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i}, {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j}) = \begin{cases} 0, & d_{\mathbf{P}}({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i}, {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j}) > \tau; \\ \left(1 + d_{\mathbf{P}}({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i}, {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j})\tau^{-1}\right)^{-1}, & \text{sonst.} \end{cases} \tag{4.16}$$

Die Funktion ist 1, falls die Lage mit dem Clusterzentrum identisch ist, 0 falls sie einen Unterschied $d_{\mathbf{P}}(\cdot)$ zum Zentrum größer τ aufweist und ansonsten im Intervall $[\frac{1}{2}, 1]$. Damit kann nun die Größe bzw. Qualität q_i jedes Clusters $G_i = \{{}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j} \in \mathcal{H} \mid \chi_\tau({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i}, {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j}) \neq 0\}$ bewertet werden:

$$q_i = \sum_{{}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j} \in \mathcal{H}} w_j \chi_\tau({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},i}, {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j}) \tag{4.17}$$

Dabei ist w_j ein optionales Gewicht, welches die Güte der lokalen Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},j}$ wiederspiegelt. Es ist im einfachsten Fall 1, kann aber bei diversen Erweiterungen (siehe Abschnitt 4.5) einen anderen Wert annehmen. Das Cluster $\mathcal{G}_{\mathrm{argmax}_i q_i}$ mit der besten Bewertung wird dann zurückgegeben. Alle weiteren Cluster werden daraufhin auf der reduzierten Menge $\mathcal{H} \setminus \mathcal{G}_{\mathrm{argmax}_i q_i}$ bestimmt. Diese Vorgehensweise hat einige Vorteile:

- Die Anzahl k der Cluster muss nicht bekannt sein. Der Algorithmus liefert immer jeweils den am besten bewerteten Cluster der übergebenen Menge $\mathcal{H}$.
- Der Algorithmus kann iterativ *n* mal durchlaufen werden oder adaptiv abgebrochen werden, wenn die beste Bewertung unter eine Schwelle fällt.
- Lagen fehlerhafter Korrespondenzen werden selten einem gut bewerteten Cluster zugeordnet. Daher arbeitet der Algorithmus nicht nur als Datensegmentierung sondern auch als effektive Ausreißer-Eliminierung.
- Die beiden einzigen Parameter $\mathbf{P}$ und τ lassen sich gut interpretieren und daher sinnvoll im Experimentalteil in Abschnitt 6.1.2.1 bestimmen.
- Der Algorithmus lässt sich bis zur Anzahl der Lagen in $\mathcal{H}$ parallelisieren und sich daher effizient implementieren.

4.3.4 Robuste globale 6 DoF Lagebestimmung

Ein Cluster $\mathcal{G}$ besteht aus einer Menge von lokalen Posen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l} \in \mathcal{G}$. Jede dieser Posen ist für sich genommen eine 6 DoF Lagehypothese eines Objektes. Allerdings ist sie aufgrund ihrer lokalen Informationsauswertung von begrenzter Genauigkeit. Es sollen daher alle geclusterten Lagen von $\mathcal{G}$ herangezogen werden, um eine robuste und genaue 6 DoF Lagehypothese ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ des Objekts ermitteln zu können. Da in diese Hypothese über das gesamte Objekt verstreute Regionen einfließen, wird sie auch als *globale* Lage bezeichnet. Weiterhin soll neben der Lage ein absolutes Qualitätsmaß $q_a \in \mathbb{R}_{0+}$ sowie ein relatives Qualitätsmaß $q_r \in [0,1]$ ermittelt werden, welches die Güte bzw. die Vertrauenswürdigkeit der gefundenen globalen Lagehypothese wiederspiegelt. Es werden in dieser Arbeit mehrere Ansätze zur globalen 6 DoF Lagebestimmung aus einem Cluster untersucht, die im Folgenden beschrieben werden. Ihre Ansätze werden in Abb. 4.2 schematisch dargestellt.

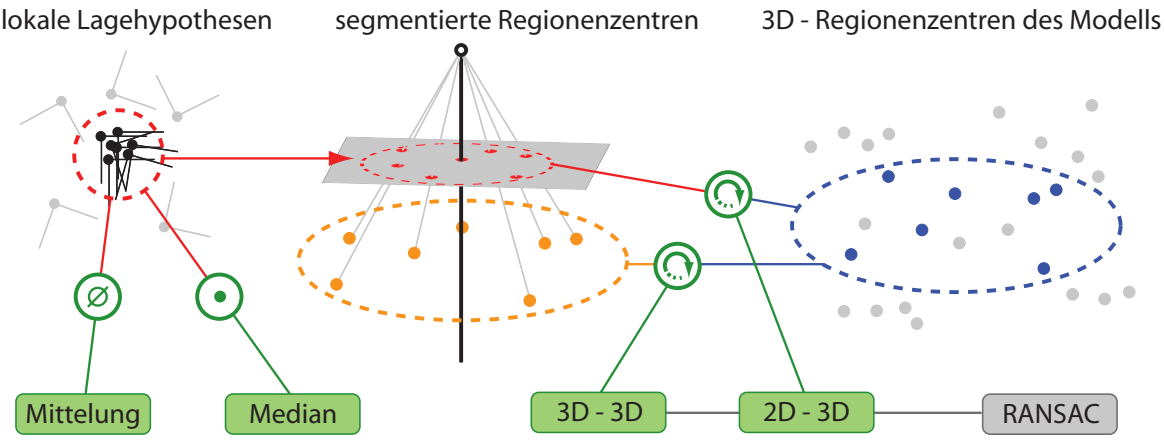

Abbildung 4.2: Schematische Darstellung der vier unterschiedlichen Ansätze der globalen Lageauswertung (grün). Ausgangspunkt ist jeweils ein gefundenes Cluster (schwarz) im Raum der lokalen Lagen. Die *Mittelung* und der *Median* leiten eine globale Lagehypothese direkt anhand der geclusterten lokalen Lagen durch eine Durchschnittsbildung bzw. durch das Clusterzentrum ab. Die 3D-3D bzw. 2D-3D Auswertung nutzt die, durch das Clustering bekannte Segmentierung der Regionenzentren in den Suchansichten und ermittelt eine globale Transformation zwischen deren Punktwolke und der Punktwolke der korrespondierenden 3D-Regionenzentren (blau) in den trainierten Modellansichten. Dazu können entweder die 2D-Zentren (rot) in der Bildebene oder die rückprojizierten 3D-Zentren (orange) der Regionen in den Suchansichten verwendet werden. Die 3D-3D- bzw. 2D-3D-Punktkorrespondenz-Verfahren nutzen zusätzlich RANSAC um Fehlkorrespondenzen innerhalb des Clusters kompensieren zu können.

4.3.4.1 Lagebestimmung mittels Durchschnittsbildung

Nimmt man an, dass das Rauschen bzw. die Fehler der lokalen Posen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l} \in \mathcal{G}$ symmetrisch verteilt ist, kann man über alle Posen mitteln um eine genauere globale Lagehypothese zu erhalten. Dabei werden Orientierung und Position getrennt behandelt. Sei ${}^{\mathrm{W}}\mathbf{t}_i$ der Verschiebungsvektor bzw. Ursprung von ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l} \in \mathcal{G}$, dann ergibt sich der gemittelte Ursprung zu:

$$ {}^{\mathrm{W}}\bar{\mathbf{t}} = \Big(\sum_{{}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l}\in\mathcal{G}} w_l\Big)^{-1} \sum_{{}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l}\in\mathcal{G}} w_l\, {}^{\mathrm{W}}\mathbf{t}_l \tag{4.18} $$

Die Gewichte w_l sind dabei dieselben wie in Abschnitt 4.3.3.2 während des Clusterns. Etwas komplizierter ist die Berechnung der mittleren Orientierung. Dazu wird auf die Quaternionen-Darstellung ${}^{\mathrm{W}}\dot{\mathbf{q}}_{\mathrm{O},l}$ der Orientierung von ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l}$ zurückgegriffen und mittels der Formel B.1 iterativ berechnet:

$$ {}^{\mathrm{W}}\bar{\mathbf{q}}_{\mathrm{O}}^{[1]} = {}^{\mathrm{W}}\dot{\mathbf{q}}_{\mathrm{O},1} \tag{4.19} $$

$$ {}^{\mathrm{W}}\bar{\mathbf{q}}_{\mathrm{O}}^{[l]} = \dot{\mathbf{q}}_{t^{[l]}}\big({}^{\mathrm{W}}\bar{\mathbf{q}}_{\mathrm{O}}^{[l-1]}, {}^{\mathrm{W}}\dot{\mathbf{q}}_{\mathrm{O},l}\big) \tag{4.20} $$

mit dem Interpolationsparameter $t^{[l]} = w_l(\sum_{i=1}^{l} w_i)^{-1}$. Die Iterationsschleife wird dabei genau $l_{\max} = |\mathcal{G}|$ mit allen Elementen von $\mathcal{G}$ durchlaufen. Eine einfache Addition wie bei der Positionsmittelung ist bei Quaternionen nicht erlaubt, da die Mittlung auf einer sphärischen Mannigfaltigkeit erfolgt. Die gemittelte Position ${}^{\mathrm{W}}\bar{\mathbf{t}}$ und Orientierung ${}^{\mathrm{W}}\bar{\mathbf{q}}_{\mathrm{O}}$ wird dann zur globalen Hypothese ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ zusammengefasst.

Um robust gegenüber fälschlicherweise geclusterten Lagen zu werden, werden nach der Mittelung die Residuen $e_l = d_{\mathbf{P}}({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l}, {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}})$ zwischen den lokalen und der globalen Lage berechnet. Die

$p = 10\%$ der Lagen mit dem größten Fehler werden aus dem Cluster entfernt und die globale Hypothese erneut berechnet. Dieser Vorgang wird $k = 10$ mal wiederholt, so dass am Schluss 35% der Lagen aus dem ursprünglichen Cluster für die Ermittelung von ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ verwendet werden. Das Verfahren verhält sich recht gutmütig bzgl. der Parameter p und k, so dass auf deren Optimierung verzichtet wird. Die Ausreißer-Eliminierung kann natürlich über Gewichte weiter verbessert werden, es zeigt sich allerdings in Kapitel 6, dass die Durchschnittsbildung den anderen Methoden unterlegen ist. Sie wird daher nicht weiter ausgearbeitet.

Das absolute Qualitätsmaß $q_a = |\mathcal{G}|$ wird über die Anzahl der lokalen Lagen festgelegt, die in die globale Hypothese mit eingeflossen sind. Für das relative Qualitätsmaß $q_r = q_a n_a^{-1}$ wird das absolute Qualitätsmaß normiert. $n_a = cn_r$ ist dabei die maximal beobachtete Anzahl von Regionen n_r in einem Trainingsbild mit nur einem Objekt, multipliziert mit der Anzahl c verwendeter Suchansichten bzw. Kameras.

4.3.4.2 Lagebestimmung mittels Medianbildung

Im Hinblick auf Fehlkorrespondenzen bzw. Ausreißer innerhalb eines Clusters ist es eine interessante Alternative, den Median anstatt den arithmetischen Durchschnitt zu bilden. Allerdings kann der Median in einem 6D-Raum nicht mehr vernünftig angegeben werden, da eine Sortierung der Daten in mehreren gleichberechtigten Dimensionen nicht mehr möglich ist. Der Median ist allerdings ein Maß für dasjenige Element, das die Datenmenge in zwei gleichgroße Untermengen teilt. Er kann daher als die „Mitte“ der Datenmenge angesehen werden und liegt bei gaußverteilten Daten im Häufungszentrum.

Ein Element mit diesen Eigenschaften lässt sich aber auch für $\mathcal{G}$ angeben, es ist das Clusterzentrum ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ aus Abschnitt 4.3.3.2. Im Gegensatz zur arithmetischen Mittelung ist es unabhängig von einzelnen Fehlkorrespondenzen und wird über die gesamte Verteilung der lokalen Lagen bestimmt. Wird das Clusterzentrum als globale Hypothese verwendet, so entspricht die globale Lagebestimmung einem weiteren, sehr restriktiven Eleminierungsschritt. Im Idealfall entspricht ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ der besten Regionenkorrespondenz, d. h. der besten lokalen Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l} \in \mathcal{G}$.

Das Clusterzentrum kann aber nur verwendet werden, falls die komplette 6 DoF Lage aus den Regionenkorrespondenzen extrahiert wird. Im Falle von 4 DoF ähnlich-kovarianten Regionen kann im Gegensatz zu den anderen Verfahren auch global keine Verkippung des Objekts aus der Bildebene geschätzt werden.

Die Qualitätsmaße werden wie in Abschnitt 4.3.4.1 für die Durchschnittsbildung definiert.

4.3.4.3 Lagebestimmung mit 3D-3D-Korrespondenzen

Die Lagebestimmung über die arithmetische Mittelung oder die Medianbildung greifen beide auf die rekonstruierten lokalen 6 DoF Lagen zurück. Wie in den Abschnitten 4.3.1 und 4.3.2 bereits diskutiert, ist gerade die Orientierung aufgrund ihrer indirekten Konstruktionsweise stärker von Fehlern behaftet als die Zentren ${}^{\mathrm{C_i}}\mathbf{p}$ und ${}^{\mathrm{M_j}}\mathbf{p}$ der korrespondierenden Regionen in Suchansicht $\mathbf{c}_i$ und Modellansicht $\mathbf{m}_j$. Es ist daher sinnvoll, bewusst auf den unsicheren Informationsgehalt der lokalen Orientierungen zu verzichten und statt dessen die zugrunde liegenden robusteren 3D-3D-Punktkorrespondenzen $({}^{\mathrm{C_i}}\mathbf{p}, {}^{\mathrm{M_j}}\mathbf{p})$ der lokalen Lagen für die globale Hypothese zu verwenden. Die Orientierungsinformationen dienen dann nur während des Clusters einer eindeutigeren Segmentierung. Dort kann aber aufgrund des Distanzmaßes besser auf die Genauigkeitsunterschiede eingegangen werden.

Verwendet man bildbasierte 2D-Regionen, wird die Tiefe des 3D-Zentrums ${}^{\mathrm{C_i}}\mathbf{p}$ über den rekonstruierten Skalierungsunterschied s_l der korrespondierenden Regionen aus dem Modell geschätzt. Auch diese Berechnung hat ein ungünstiges Fehlerverhalten, so dass alternativ auch ein triangulationsbasierter Ansatz zwischen zwei Suchansichten herangezogen werden kann oder muss, um die

Tiefe robust zu bestimmen. Beide Methoden werden in Kapitel 6 untersucht.

Unabhängig davon wie man die 3D-3D-Korrespondenzen $({}^{\mathrm{C_i}}\mathbf{p}, {}^{\mathrm{M_j}}\mathbf{p})$ berechnet, werden sie über die im Modell gespeicherten Starrkörper-Transformationen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}$ und ${}^{\mathrm{M_j}}\check{\mathbf{H}}_{\mathrm{O}}$ in die universelle Darstellung $({}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}\,{}^{\mathrm{C_i}}\check{\mathbf{p}}, {}^{\mathrm{M_j}}\check{\mathbf{H}}_{\mathrm{O}}^{-1}\,{}^{\mathrm{M_j}}\check{\mathbf{p}}) = ({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})$ übertragen. Ein Cluster $\mathcal{G}$ enthält dann genau $n = |\mathcal{G}|$ universelle 3D-3D-Punktkorrespondenzen $({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})_l$, $l = 1,\ldots,n$ von Welt-Koordinatensystem W und Objekt-Koordinatensystem O. Gesucht wird dann die globale Lage ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$, welche alle Punkte ${}^{\mathrm{W}}\check{\mathbf{p}}_l = {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}\,{}^{\mathrm{O}}\check{\mathbf{p}}_l$ ineinander überführt.

Für diese Problemstellung gibt es mehrere in Abschnitt 2.2.2.1 beschriebene Methoden. Alle Methoden lassen sich analytisch geschlossen berechnen und sind im Sinne der kleinsten Fehlerquadrate optimal. Sie schätzen optional ebenfalls einen globalen Skalierungsparameter s_g zwischen den zwei gesamten übergebenen Punktwolken. Da aber nur eine RT ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ ermittelt werden soll, wird $s_g = 1$ festgelegt. Hier wird die Methode mittels Einheitsquaternionen (Hor87) verwendet.

Das absolute Qualitätsmaß q_a wird über den quadratischen Fehler aus (Hor87) definiert. Sei χ_ε eine Bewertungsfunktion

$$\chi_\varepsilon[{}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}]({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}}) = \begin{cases} 0, & |{}^{\mathrm{W}}\check{\mathbf{p}} - {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}\,{}^{\mathrm{O}}\check{\mathbf{p}}| > \varepsilon; \\ \left(1 + |{}^{\mathrm{W}}\check{\mathbf{p}} - {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}\,{}^{\mathrm{O}}\check{\mathbf{p}}|^2\varepsilon^{-2}\right)^{-1}, & \text{sonst.} \end{cases} \tag{4.21}$$

die 0 ist, falls die zwei korrespondierenden Punkte $({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})$ nach Überführung mittels der geschätzten, globalen Hypothese ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ in das gemeinsame Koordinatensystem W weiter entfernt sind als ε. Ansonsten liegt $\chi_\varepsilon(\cdot)$ im Intervall $[\frac{1}{2}, 1]$, wobei die Funktion nur dann 1 annimmt, falls die beiden Punkte nach der Transformation perfekt übereinstimmen. Das absolute Qualitätsmaß wird dann über alle Korrespondenzen $({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})_i$, $i = 1,\ldots,n$ des Clusters $\mathcal{G}$ wie folgt berechnet:

$$q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}) = -3 + \sum_{l=1}^{n} w_l \chi_\varepsilon[{}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}]({}^{\mathrm{W}}\check{\mathbf{p}}_l, {}^{\mathrm{O}}\check{\mathbf{p}}_l) \tag{4.22}$$

Die Gewichte w_l sind dabei dieselben wie in Abschnitt 4.3.3.2 während des Clusterns. Da zu der Berechnung der globalen Hypothese ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ mindestens 3 Punktkorrespondenzen notwendig sind, wird q_a ein Penalty von 3 abgezogen. Dies führt dazu, dass ein Cluster mit $|\mathcal{G}| = 3$ ein Güte von $q_a = 0$ besitzt, da keinerlei Aussage über die Richtigkeit der gefundenen Hypothese gemacht werden kann. Durch das Penalty kann es vorkommen, das bei ungünstigen Konstellationen q_a negativ wird. In diesem Fall wird $q_a = 0$ gesetzt.

Das relative Qualitätsmaß $q_r = q_a(\sum_{l=1}^{n} w_l)^{-1}$ wird durch eine Normierung von q_a über die Clustergröße $n = |\mathcal{G}|$ unter Berücksichtigung der Gewichte erreicht. Es kann selbst bei $n \geq 3$ perfekten Korrespondenzen maximal $q_r = (n-3)n^{-1}$ werden. Über das Penalty wird daher gewährleistet, dass kleine, wenig vertrauenswürdige Cluster auch durch q_r nicht zu gut bewertet werden.

Bei der Berechnung der globalen Hypothese sind bis jetzt allerdings noch keine Fehlkorrespondenzen innerhalb des Clusters berücksichtigt worden. Eine Möglichkeit wäre ein iteratives Verfahren ähnlich wie bei der Durchschnittsbildung. Allerdings benötigt man nur 3 korrekte 3D-3D- Punktkorrespondenzen zur Berechnung von ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ und hat mit $q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}})$ eine plausible Bewertungsmöglichkeit, so dass sich ein RANSAC-basiertes Verfahren anbietet. Dieses hat im Vergleich zu einem iterativen Ansatz weiterhin den Vorteil, dass sich die globale Hypothesenbestimmung dann fein granular bis zur Anzahl der RANSAC-Zyklen parallelisieren lässt.

Für die ausreißertolerante RANSAC-basierte Lageermittlung werden zuerst m_{RAN} mal $n_{\mathrm{RAN}} \geq 3$ paarweise unterschiedliche Punktkorrespondenzen $({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})_{kl}$, $k = 1,\ldots,m_{\mathrm{RAN}}$, $l = 1,\ldots,n_{\mathrm{RAN}}$ aus dem Cluster $\mathcal{G}$ gleichwahrscheinlich zufällig bestimmt. In jedem RANSAC-Zyklus k wird dann die globale Lage ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},k}$ aus den m_{RAN} 3D-3D- Punktkorrespondenzen $({}^{\mathrm{W}}\check{\mathbf{p}}, {}^{\mathrm{O}}\check{\mathbf{p}})_{kl}$, über die Methode von (Hor87) bestimmt und mittels $q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},k})$ über *alle* Korrespondenzen von $\mathcal{G}$ bewertet. Die endgültige

Lage erhält man durch ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}} = {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},k_{\max}}$ mit $k_{\max} = \operatorname{argmax}_k q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},k})$, d. h. durch die am besten bewertete Lage der einzelnen RANSAC-Zyklen.

Dieses Verfahren ist äußerst robust gegenüber Fehlkorrespondenzen in dem Cluster $\mathcal{G}$. Die globale Lage wird zwar nur noch aus n_{RAN} Punkten berechnet, allerdings wird durch die Bewertung über alle Korrespondenzen innerhalb des Clusters gewährleistet, dass bei entsprechend vielen RANSAC-Zyklen m_{RAN} eine sehr gute Kombination von 3D-3D-Korrespondenzen zur Lagebestimmung herangezogen wird. Dabei muss man bei der Wahl von n_{RAN} einen Kompromiss eingehen. Wählt man n_{RAN} sehr klein, kann bei der Lagebestimmung das Rauschen innerhalb der Korrespondenzen dominieren, da man in (Hor87) nur noch wenig mitteln kann. Wählt man n_{RAN} dagegen größer, steigt die Wahrscheinlichkeit, dass sich innerhalb eines RANSAC-Zykluses mindestens ein Ausreißer befindet und damit auch die Wahrscheinlichkeit, dass in keinem der Zyklen eine korrekte globale Lage berechnet werden kann. Dieser Effekt kann durch eine entsprechende Erhöhung von m_{RAN} auf Kosten der Laufzeit bis zu einem gewissen Grad ausgeglichen werden.

4.3.4.4 Lagebestimmung mit 2D-3D-Korrespondenzen

Die Verwendung der 3D-3D-Punktkorrespondenzen bei bildbasierten 2D-Regionen hat den Nachteil, dass man zur Bestimmung des 3D-Zentrums ${}^{\mathrm{C_i}}\mathbf{p}$ entweder über die lokale Skalierung s_l mit ungünstiger Fehlerfortpflanzung gehen muss oder eine triangulationsbasierte Tiefenauswertung zweier Stereosuchansichten mit entsprechender Laufzeit benötigt. Für eine 2D-Region gibt es aber die Alternative anstatt dem 3D-Zentrum ${}^{\mathrm{C_i}}\mathbf{p}$ der Region nur das 2D-Zentrum ${}^{\mathrm{C_i}}\mathbf{u}$ innerhalb der Suchansicht $\mathbf{c}_i(\cdot)$ zu verwendet. Anstatt $n = |\mathcal{G}|$ 3D-3D-Korrespondenzen erhält man dann n 2D-3D-Korrespondenzen $({}^{\mathrm{C_i}}\mathbf{u}, {}^{\mathrm{O}}\check{\mathbf{p}}) = ({}^{\mathrm{C_i}}\mathbf{u}, {}^{\mathrm{O}}\check{\mathbf{H}}_{\mathrm{M_j}}\, {}^{\mathrm{M_j}}\check{\mathbf{p}})$ für ein Cluster $\mathcal{G}$. Allerdings verliert man dadurch die Möglichkeit die Korrespondenzen vollständig in die universelle Darstellung zu übertragen. Es muss daher die Berechnung der globalen Lage ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ über Korrespondenzen einer einzelnen Suchansicht erfolgen.

Sei $\mathcal{G}_{\mathbf{c}_i} \subseteq \mathcal{G}$ die Menge der Korrespondenzen zwischen Suchansicht $\mathbf{c}_i(\cdot)$ und allen Modellansichten. Gesucht ist dann die Transformation ${}^{\mathrm{C_i}}\bar{\mathbf{H}}_{\mathrm{O}}$, so dass die Objektpunkte ${}^{\mathrm{O}}\mathbf{p}_l$ möglichst genau über das Lochkameramodell $\mathbf{h}(\cdot)$ auf die Bildpunkte ${}^{\mathrm{C_i}}\mathbf{u}_l$ projiziert werden, d. h. $|{}^{\mathrm{C_i}}\mathbf{u}_l - \mathbf{h}({}^{\mathrm{C_i}}\bar{\mathbf{H}}_{\mathrm{O}})\,{}^{\mathrm{O}}\mathbf{p}_l|$ für alle Punkte $l = 1,\ldots,n_i$ mit $n_i = |\mathcal{G}_{\mathbf{c}_i}|$ minimal wird. Für diese Problemstellung werden in Abschnitt 2.2.2.2 verschiedene Algorithmen vorgestellt, wovon in dieser Arbeit POSIT (DD95) samt komplanarer Erweiterung (ODD96) und der OI-Algorithmus (LHM00) in Kapitel 6 untersucht werden. Im Falle von komplanaren Objektpunken liefert POSIT aus dem gleichen Grund wie die Rotationenrekonstruktion aus Abschnitt 4.3.1.1 zwei Lösungen ${}^{\mathrm{C_i}}\bar{\mathbf{H}}^{\pm}_{\mathrm{O}}$. Die globale Lagehypothesen erhält man durch ${}^{\mathrm{W}}\bar{\mathbf{H}}^{\pm}_{\mathrm{O}} = {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}\, {}^{\mathrm{C_i}}\bar{\mathbf{H}}^{\pm}_{\mathrm{O}}$.

Die Bewertung erfolgt analog wie in Abschnitt 4.3.4.3 über den quadratischen Fehler:

$$\chi_\kappa[{}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}]({}^{\mathrm{C_i}}\mathbf{u}, {}^{\mathrm{O}}\check{\mathbf{p}}) = \begin{cases} 0, & |{}^{\mathrm{C_i}}\mathbf{u} - \mathbf{h}({}^{\mathrm{W}}\check{\mathbf{H}}^{-1}_{\mathrm{C_i}}\, {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}\, {}^{\mathrm{O}}\check{\mathbf{p}})| > \kappa; \\ \left(1 + \frac{|{}^{\mathrm{C_i}}\mathbf{u} - \mathbf{h}({}^{\mathrm{W}}\check{\mathbf{H}}^{-1}_{\mathrm{C_i}}\, {}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}\, {}^{\mathrm{O}}\check{\mathbf{p}})|^2}{\kappa^2}\right)^{-1}, & \text{sonst.} \end{cases} \tag{4.23}$$

Allerdings bezieht sich der Schwellwert κ nun nicht mehr wie der Parameter ε auf Welt-Koordinaten sondern auf Bildkoordinaten. Das absolute Qualitätsmaß ergibt sich dann analog über *alle* n 2D-3D-Korrespondenzen innerhalb des Clusters zu

$$q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}) = -3 + \sum_{l=1}^{n} w_l \chi_\kappa[{}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}]({}^{\mathrm{C_i}}\mathbf{u}_l, {}^{\mathrm{O}}\check{\mathbf{p}}_l) \tag{4.24}$$

ebenso wie das relative Maß zu $q_r = q_a(\sum_{l=1}^{n} w_l)^{-1}$. Die Gewichte w_l sind dabei dieselben wie in Abschnitt 4.3.3.2 während des Clusterns. Da wie bei Berechnung mittels 3D-3D-Korrespondenzen

mindestens 3 unterschiedliche 2D-3D-Korrespondenzen benötigt werden, wird der Penalty ebenfalls auf 3 festgelegt.

Ebenso wie in Abschnitt 4.3.4.3 wird die globale Lageschätzung mit 2D-3D-Korrespondenzen ausreißertolerant mittels RANSAC realisiert. Es gibt allerdings zwei kleine Unterschiede im Vergleich zu dem 3D-3D-Ansatz. Beim zufälligen Ziehen der n_{RAN} Korrespondenzen in jedem RANSAC-Zyklus bestimmt die erste gezogene Korrespondenz $({}^{\mathrm{C_i}}\mathbf{u}, {}^{\mathrm{O}}\mathbf{p})$ die Suchansicht $\mathbf{c}_i$. Alle weiteren $n_{\mathrm{RAN}} - 1$ Korrespondenzen dieses Zyklusses werden dann nur aus der Menge $\mathcal{G}_{\mathbf{c}_i}$ gewählt. Dieser Mechanismus gewährleistet, dass alle Suchansichten in Abhängigkeit ihrer Anzahl von Regionenkorrespondenzen innerhalb des Clusters gleichberechtigt zur Berechnung von ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ herangezogen werden. Sollten in einem Zyklus durch komplanare Objektpunkte zwei Lösungen gefunden werden, wird nur diejenige mit der größeren Bewertung q_a behalten. Die restliche Auswertung erfolgt dann ebenso wie die Diskussion der Parameter analog zu Abschnitt 4.3.4.3.

Zwei Anmerkungen sind zur globalen Lageschätzung mittels 2D-3D-Korrespondenzen zu machen. Erstens ähnelt dieser Ansatz der globalen Auswertung von (AAD07; Aza08) falls nur eine einzige Suchansicht verwendet wird. Allerdings erfolgt dort das Clustering über Hough in der Bildansicht und nicht im 6D-Raum der RT. Es wird zwar bei dem hier beschriebenen Ansatz die Lage ebenfalls nur mittels einer Suchansicht bestimmt, allerdings hilft die Bewertung über alle Suchansichten die bestmögliche globale Lage der einzelnen RANSAC-Zyklen zu finden. Als zweite Anmerkung sei gesagt, dass es ebenfalls Algorithmen (CC04) über alle Suchansichten gibt. Allerdings standen diese zum Zeitpunkt der Arbeit nicht zur Verfügung und konnten daher aus zeitlichen Gründen nicht untersucht werden.

4.3.5 Ablauf der gesamten Verarbeitungskette

Abschließend soll noch einmal ein Überblick über die gesamte online (d. h. in einen Verarbeitungsprozess eingebundene) Verarbeitungskette der Objektlokalisation gegeben werden. Dabei wird auch zusammengefasst, welche Informationen im Vorfeld benötigt werden und in einem offline (d. h. nicht in einen Verarbeitungsprozess eingebundenen) Trainingsschritt gesammelt werden müssen. Es wird dabei zwischen dem Modell, d. h. objektspezifischen Informationen und den Kalibrierdaten, d. h. sensorspezifischen Informationen unterschieden. Erstere müssen für jedes Objekt neu trainiert werden, letztere nur bei einer Änderung des Hardwareaufbaus. Abbildung 4.3 stellt die Verarbeitungskette graphisch dar.

Ausgangspunkt ist eine Menge $\mathcal{C}$ von c Suchansichten $\mathbf{c}_i(\cdot)$, $i = 1,\ldots,c$. Auf jede dieser Ansichten werden ein oder mehrere Detektoren aus Abschnitt 3.2 angewandt, um n_l kovarianten Such-Regionen ${}^{\mathrm{C_{i(l)}}}\mathcal{R}_l$, $l = 1,\ldots,n_l$ zu bestimmen. Die Funktion $i(l)$ gibt dabei für jede Region $\mathcal{R}_l$ den Index i der zugrunde liegenden Suchansicht zurück. Mittels eines Deskriptors aus Abschnitt 3.3 werden für alle Regionen die n_l Merkmalsvektoren $\mathbf{o}_l = \mathbf{des}(\mathcal{R}_l)$ berechnet. Prinzipiell können für unterschiedliche Detektoren auch unterschiedliche Deskriptoren verwendet werden, im Zuge einer Vereinheitlichung ist aber eine gemeinsame Beschreibung für alle Regionentypen sinnvoller.

Das Modell eines Objektes besteht aus einer Menge $\mathcal{M}$ von m Modellansichten $\mathbf{m}_j(\cdot)$, $j = 1,\ldots,m$, die während des Trainings aufgenommen werden müssen. Auf diese Ansichten werden dieselben Detektoren zur Extraktion von n_k kovarianten Modell-Regionen ${}^{\mathrm{M_{j(k)}}}\mathcal{R}_k$, $k = 1,\ldots,n_k$ angewandt und der gleiche Deskriptor zu Bestimmung der Merkmalsvektoren $\mathbf{o}_k = \mathbf{des}(\mathcal{R}_k)$ verwendet. Die Merkmalsvektoren werden in einer Datenbank DB abgelegt und von einem Matcher aus Abschnitt 3.4 genutzt, um n_l korrespondierende Regionen $({}^{\mathrm{C_{i(l)}}}\mathcal{R}_l, {}^{\mathrm{M_{j(k(l))}}}\mathcal{R}_{k(l)})$ zu finden. Die Zuordnung der Such-Regionen $\mathcal{R}_l$ zu den abgelegten Modell-Regionen $\mathcal{R}_k$ beschreibt die Funktion $k(l)$.

Die n_l lokalen, kovarianten Regionenkorrespondenzen dienen als Schnittstelle zwischen den ersten drei Schritten zur Bestimmung der Korrespondenzen und der nachfolgenden 6 DoF Objektlokalisation. In Abb. 4.3 wird dies durch einen blauen Balken symbolisiert, der die Algorithmen zur

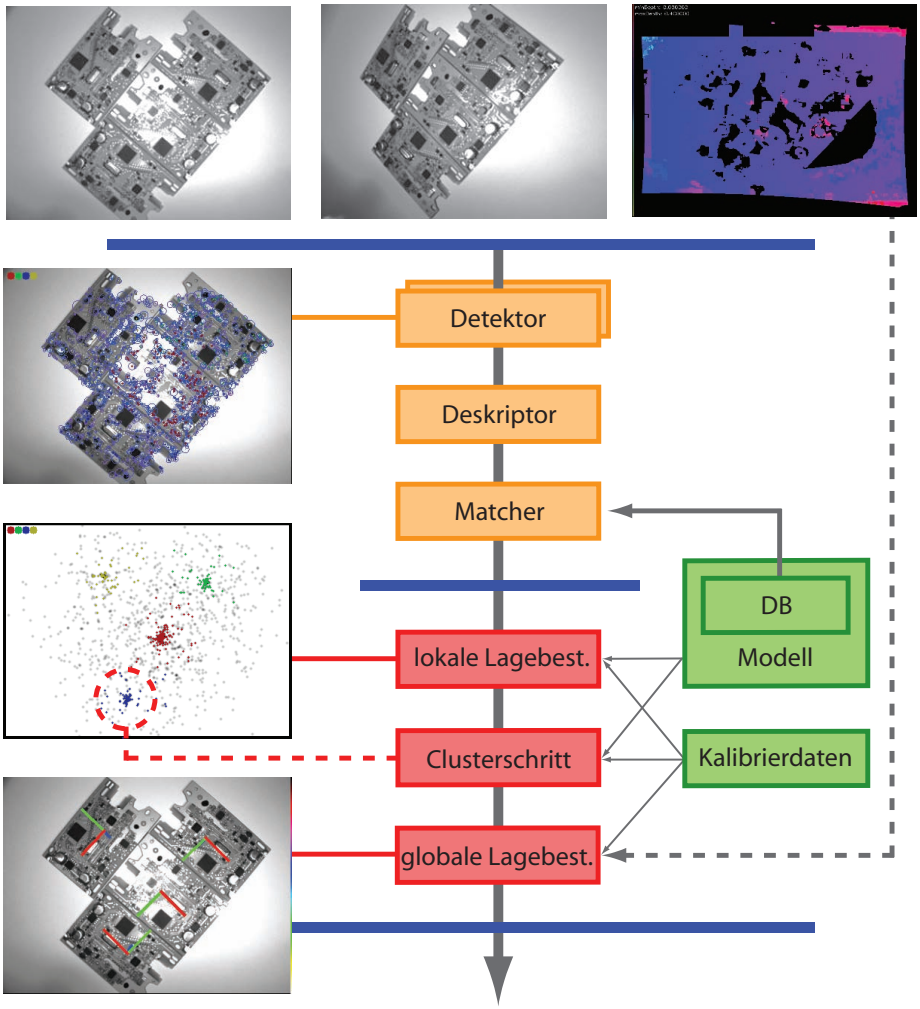

Abbildung 4.3: Überblick über die gesamte Verarbeitungskette der Objektlokalisation. Orange sind die Komponenten zur Findung lokaler, kovarianter Regionenkorrespondenzen dargestellt, rot die nachfolgenden Komponenten der in dieser Arbeit beschriebenen Objektlokalisation und grün die für die jeweiligen Komponenten notwendigen Daten. Ausgangspunkt sind die ganz oben dargestellten Suchansichten sowie optional für die globale Lageschätzung mittels 3D-3D-Korrespondenzen ein Tiefenbild.

Korrespondenzfindung (orange) von den unabhängigen Algorithmen der Lokalisation (rot) trennt. Es werden in der Lokalisation aus Laufzeitgründen nur maximal $n_{\max}$ Korrespondenzen verarbeitet. Sollte $n_l > n_{\max}$ sein, werden die Korrespondenzen $({}^{\mathrm{C_{i(l)}}}\mathcal{R}_l, {}^{\mathrm{M_{j(k(l))}}}\mathcal{R}_{k(l)})$ anhand ihrer Merkmalsabweichung $d_{\mathbf{des}}(\mathbf{o}_l, \mathbf{o}_{k(l)})$ sortiert und nur die ähnlichsten $n_{\max}$ Korrespondenzen weiter verwendet.

Der erste Schritt der Lokalisation bestimmt für jede einzelne Korrespondenz die lokale Lage ${}^{\mathrm{C_{i(l)}}}\check{\mathbf{H}}_{\mathrm{M_{j(k(l))}}}$ mittels dem Verfahren aus Abschnitt 4.3.1. Dafür werden die Tiefe ${}^{\mathrm{M_{j(k(l))}}}p_z$ des Mittelpunkts der Modell-Region als auch die Ebenenparameter m_x und m_y benötigt. Diese Größen müssen während des Trainings für alle n_k Modell-Regionen bestimmt werden. Weiterhin ist dieser Schritt abhängig von den Kalibrierdaten, da er die Kameramodelle $\mathbf{h}_i(\cdot)$ bzw. $\mathbf{h}_j(\cdot)$ der einzelnen Such- bzw. Modellansichten benötigt.

Der nachfolgende Clusterschritt überführt alle lokalen Hypothesen in die universelle Darstellung ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},l} = {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_{i(l)}}}\,{}^{\mathrm{C_{i(l)}}}\check{\mathbf{H}}_{\mathrm{M_{j(k(l))}}}\,{}^{\mathrm{M_{j(k(l))}}}\check{\mathbf{H}}_{\mathrm{O}}$. Dafür benötigt er die im Modell abgespeicherten Matrizen

${}^{\mathrm{M_j}}\check{\mathbf{H}}_{\mathrm{O}}$, die während des Trainings alle Modellansichten bzgl. eines Objekt-Koordinatensystems O registrieren. Weiterhin werden aus den Kalibrierdaten die extrinsischen Parameter ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C_i}}$ benötigt, die alle Kameras bzgl. eines gemeinsamen Welt-Koordinatensystems W ausrichten. In der universellen Darstellung werden mittels dem Verfahren aus Abschnitt 4.3.3 die n_g besten Cluster $\mathcal{G}_g$, $g = 1\ldots,n_g$ bestimmt. Dabei spielt für den Parameter n_g die tatsächliche Anzahl von beobachtbaren Objekten keine Rolle. Sollten mehr Cluster gefunden werden, als reale Objekte in der Szene zu sehen sind, dann werden die falschen Cluster in der nachfolgenden globalen Hypothesengenerierung ausgesondert. Werden dagegen weniger Cluster gesucht als durch reale Objekte vorgegeben sind, werden die am besten bewerteten Cluster zurückgegeben. Diese entsprechen normalerweise den sichtbarsten, d. h. am robustesten zu findenden Objekten.

Für jeden der n_g Cluster $\mathcal{G}_g$ wird anschließend eine globale Lagehypothese ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},g}$ mit einem der Verfahren aus Abschnitt 4.3.4 bestimmt und mittels dem absoluten Qualitätsmaß $q_a({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},g})$ und dem relativen Qualitätsmaß $q_r({}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O},g})$ bewertet. Sollten diese Maße unter den Schwellen τ_a bzw. τ_r liegen, wird die Hypothese und der zugehörige Cluster verworfen. Je nach Verfahren benötigt dieser Schritt die Kameramodelle aus den Kalibrierdaten oder Tiefenwerte aus dem Distanzbild der Suchansichten.

Die verbleibenden n_f globalen Lagehypothesen werden zusammen mit ihrem relativen Qualitätsmaß zurückgegeben und stehen dem weiterführenden Verarbeitungssystem dann zur Verfügung.

Bis auf die Verfahren für die Tiefenbildberechnung und die kovariante Regionenkonstruktion - deren Entwicklung nicht im Fokus dieser Arbeit stand - sind alle Algorithmen der Verarbeitungskette aus Abb. 4.3 bis zu dem Clusterschritt fein granular bis zur Anzahl n_l der gefundenen Regionen parallelisierbar. Die globale Hypothesengenerierung ist grob granular parallelisierbar bis zur Anzahl h_g der gefunden Cluster, je nach Verfahren aber auch fein granular bis zur Anzahl der RANSAC-Zyklen m_{RAN}.

4.4 Autonomes Training

Das Modell eines Objektes umfasst alle objektspezifischen Informationen, die für die 6 DoF Lokalisation benötigt werden. Es muss für jedes Objekt in einem separaten Trainingsschritt aufgebaut werden. Dieser Schritt geschieht offline, d. h. nicht innerhalb eines Prozesszyklus. Daher spielt im Gegensatz zur online Lokalisation die Verarbeitungsgeschwindigkeit keine Rolle, solange die gesamte Trainingszeit die Größenordnung der Projektvorgabe von ca. 10 min nicht überschreitet.

Ein wesentlicher Aspekt des Trainings ist seine Autonomie bzgl. menschlicher Eingriffe und zusätzlichem Objektwissen. Das Training verarbeitet daher weder bestehende CAD-Modelle, Texturinformtionen, Größenangaben oder ähnliches, noch ist es auf Expertenwissen oder Benutzereingaben angewiesen. Die einzige notwendige Benutzerinteraktion besteht in der Präsentation des einzulernenden Objektes vor den Kameras des Systems. Weiterhin benötigt das Training keine zusätzliche Sensorik, sondern beschränkt sich auf die während der Lokalisation verwendete Hardware.

Wie schon in der Lokalisation erwähnt, besteht das Modell eines Objektes aus einer Menge $\mathcal{M}$ von m Modellansichten $\mathrm{m}_j(\cdot)$, $j = 1,\ldots,m$ die mittels den RT's ${}^{\mathrm{M_j}}\check{\mathbf{H}}_{\mathrm{O}}$ bzgl. eines gemeinsamen Objekt-Koordinatensystems O registriert sind. Abschnitt 4.4.1 behandelt die Verarbeitung einer Modellansicht $\mathrm{m}_j(\cdot)$, d. h. die Regionenextraktion inklusive deren Tiefen- und Ebenenparameterschätzung. Für die erste Modellansicht $\mathrm{m}_1(\cdot)$ wird in Abschnitt 4.4.2 das Objekt-Koordinatensystem O festgelegt und in Abschnitt 4.4.3 genutzt, um weitere Ansichten zu registrieren.

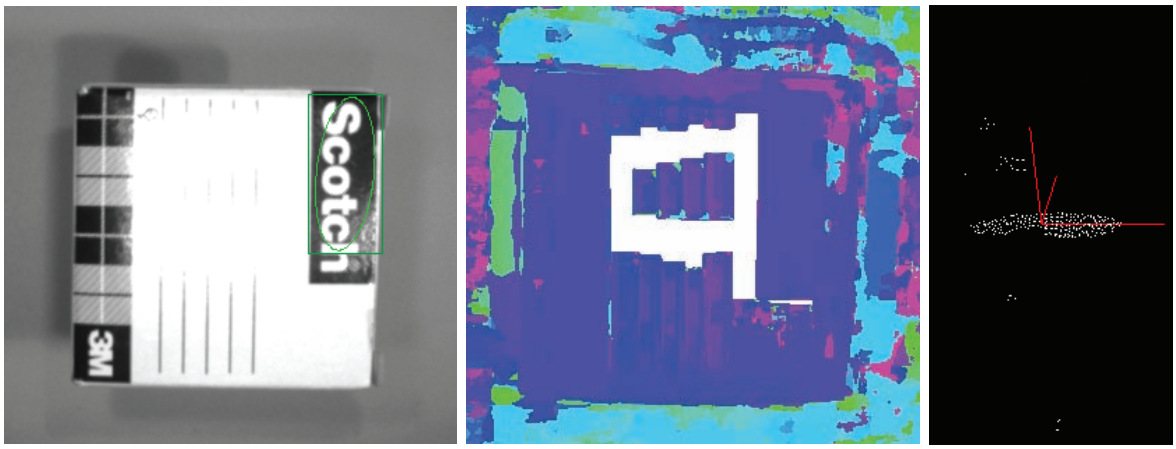

Abbildung 4.4: Schätzung der Tiefe und Ebenenparameter der im linken Ausschnitt einer Modellansicht $\mathrm{m}(\cdot)$ eingezeichneten Region $\mathcal{R}$ (grün). In der Mitte ist derselbe Ausschnitt für das Tiefenbild $\mathrm{m_d}(\cdot)$ in Falschfarbendarstellung von rot (nahe) über blau nach grün (entfernt) abgebildet. Weiß bedeutet, dass keine Informationen $\mathrm{m_d}(\cdot) = \varnothing$ vorliegen. Rechts sind für $\mathcal{R}$ die zugehörigen, rekonstruierten Szenenpunkte $\mathcal{S}[\mathcal{R}]$ (weiß) in einer 3D-Visualisierung samt geschätztem Ebenenfit (rot) dargestellt. Gut zu erkennen sind die Ausreißer aufgrund einer falschen Tiefenrekonstruktion am rechten, mittleren Rand von $\mathcal{R}$.

4.4.1 Verarbeitung einer Modellansicht

Eine Modellansicht $\mathrm{m}(\cdot)$ ist ein Abbild des einzulernenden Objektes auf einem geeigneten Hintergrund. Meist handelt es sich bei dem Hintergrund um eine, durch eine Fokussierung auf das Objekt unscharfe Region, eine homogene Fläche oder eine bekannte Szene. Wichtig ist dabei nur, dass auf dem Hintergrund entweder keine Strukturen enthalten sind, auf die der verwendete Regionen-Detektor reagiert oder eine Segmentierung zur Verfügung steht, um Objektstrukturen von Hintergrundstrukturen zu trennen.

Ist dies gewährleistet, werden aus $\mathrm{m}(\cdot)$ mit Hilfe einer oder mehrerer Detektoren die lokalen, kovarianten Regionen extrahiert, deren Deskriptoren berechnet und in der Datenbank zusammen mit einem Verweis auf die zugrunde liegende Modellansicht abgelegt. Dabei werden Regionen auf Hintergrundstrukturen nicht berücksichtigt. Für jede Region $\mathcal{R}[\mathbf{u}]$ mit Zentrum $\mathbf{u}$ muss nun der Tiefenwert p_z an der Stelle $\mathrm{m}(\mathbf{u})$ extrahiert werden, sowie die Ebenenparameter m_x und m_y innerhalb des Szenenbereichs $\mathcal{S}$ bestimmt werden, den $\mathcal{R}[\mathbf{u}]$ beschreibt.

Dazu ist ein Tiefenbild $\mathrm{m_d} : \mathcal{D} \rightarrow \mathcal{W}_\mathrm{d} \subseteq \mathbb{R}^+$ notwendig, welches an jeder Stelle $\mathbf{u} \in \mathcal{D}$ der Ansicht $\mathrm{m}(\cdot)$ einen Tiefenwert $z \in \mathcal{W}_\mathrm{d}$ zurückliefert. Je nach Aufnahmesystem kann das Tiefenbild allerdings Lücken $\mathrm{m_d}(\mathbf{u}) = \varnothing$ aufweisen, an denen keine Information zur Verfügung steht. In dieser Arbeit wird ein triangulationsbasierter Ansatz auf Basis eines Stereobildes verwendet (vgl. Anhang C.2), dessen Ergebnisse in Abb. 4.4 zu sehen sind.

Darüber lässt sich nun der Szeneninhalt bzw. die Szenenpunkte $\mathcal{S}[\mathcal{R}]$ der Region $\mathcal{R}$ bis auf die Lücken von $\mathrm{m_d}(\cdot)$ bestimmen (vgl. Abb. 4.4) :

$$\mathcal{S}[\mathcal{R}] = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{p} = \mathbf{h}^{-1}(\mathbf{v}, \mathrm{m_d}(\mathbf{v})),\ \mathrm{m_d}(\mathbf{v}) \neq \varnothing,\ \mathbf{v} \in \mathcal{R} \right\} \tag{4.25}$$

Die Ebenenparameter m_x und m_y parametrisieren die Ebene $0 = m_x x + m_y y + z = \mathbf{n}^\mathrm{T}\mathbf{p}$ mit Normalenvektor $\mathbf{n} = (m_x, m_y, 1)^\mathrm{T}$, in der nach Annahme 3 aus Abschnitt 3.1.2 alle Punkte $\mathbf{p} = (x, y, z)^\mathrm{T} \in \mathcal{S}[\mathcal{R}]$ liegen. Zur Ermittlung von $\mathbf{n}$ wird eine Regression mittels Eigenwertzerlegung aus (BV92) verwen-

det. Sei

$$\bar{\mathbf{p}} = |\mathcal{S}|^{-1} \sum_{\mathbf{p}\in\mathcal{S}} \mathbf{p} \tag{4.26}$$

$$\mathbf{P} = |\mathcal{S}|^{-1} \sum_{\mathbf{p}\in\mathcal{S}} (\mathbf{p}-\bar{\mathbf{p}})(\mathbf{p}-\bar{\mathbf{p}})^{\mathrm{T}} \tag{4.27}$$

der Schwerpunkt und die Streumatrix von $\mathcal{S}$. Der Schwerpunkt $\bar{\mathbf{p}}$ entspricht dem Aufhängepunkt der Ebene. Der Eigenvektor $\nu_{\min}(\mathbf{P}) = (x_\nu, y_\nu, z_\nu)^{\mathrm{T}}$ zum kleinsten Eigenwert $\lambda_{\min}(\mathbf{P})$ der Streumatrix $\mathbf{P}$ entspricht dem Normalenvektor $\mathbf{n} = z_\nu^{-1}\nu_{\min}$. Dabei ist $\lambda_{\min}$ gleichzeitig ein Gütemaß für den Ebenenfit, der im optimalen Fall 0 ist.

Dieser Ansatz ist aufgrund seines mittelnden Charakters zwar robust gegenüber verrauschten Daten, aber aufgrund der linearen Berechnung sehr ausreißeranfällig. Bedingt durch den Stereoansatz kann man von dem Tiefenbild $\mathrm{m_d}(\cdot)$ allerdings keine fehlerlosen Daten erwarten, im Gegenteil zeigt es sogar große, zusammenhängende Fehlerbereiche wie in Abb. 4.4 leicht ersichtlich. Es wird daher RANSAC genutzt, um einen robusten Ebenenfit zu erhalten.

Allerdings wird zur Bewertung der Ebenenparameter in jedem RANSAC-Zyklus nicht $\lambda_{\min}$ herangezogen, sondern eine Funktion q_ε, welche alle Punkte in einem ε-Schlauch um die Ebene zählt. Die Distanz eines Punktes $\mathbf{p} \in \mathcal{S}$ zur Ebene erhält man über $d(\mathbf{p}) = |\nu_{\min}^{\mathrm{T}}\mathbf{p}|$. Sei $\xi_\varepsilon(\mathbf{p}) = 1$ falls $d(\mathbf{p}) < \varepsilon$ und 0 sonst, dann ergibt sich $q_\varepsilon = \sum_{\mathbf{p}\in\mathcal{S}} \xi_\varepsilon(\mathbf{p})$. Dieses Maß hat sich in Bezug auf die charakteristische Fehlerausprägung des Stereotiefenbilds in Experimenten als am besten geeignet erwiesen.

Aufgrund des starken Rauschens ist es nicht sinnvoll die Tiefe p_z des Zentrums $\mathbf{u}$ der Region $\mathcal{R}[\mathbf{u}]$ über den Eintrag $\mathrm{m_d}(\mathbf{u})$ zu bestimmen. Einen besseren Wert erhält man, indem man den Sichtstrahl $\mathbf{o} + \mu\mathbf{x} \in \mathbb{R}^3$ des Pixels $\mathbf{u}$ mit der gefitteten Ebene schneidet und den Tiefenwert des Schnittpunktes in p_z einträgt. $\mathbf{o}$ ist dabei der Ursprung des Ansicht-Koordinatensystems M und entspricht dem Projektionszentrum der Kamera, die $\mathrm{m}(\cdot)$ aufgenommen hat. Den Geradenvektor $\mathbf{x} = \mathbf{h}^{-1}(\mathbf{u}, 1) - \mathbf{o}$ erhält man als Differenz des Projektionszentrums und des rückprojizierten Bildpunktes $\mathbf{u}$ mit einer beliebigen Tiefe, in diesem Fall 1. Falls $c = \nu_{\min}^{\mathrm{T}}\mathbf{x}$ gleich 0 ist, sind Gerade und Ebene parallel zueinander und ein Schnitt ist nicht möglich. Dieser Fall tritt allerdings nicht auf, da ansonsten die Szene $\mathcal{S}[\mathcal{R}]$ (fast) auf eine Linie abgebildet und $\mathcal{R}$ dann nicht detektiert worden wäre. Setzt man Ebene und Gerade gleich, erhält man

$$\mu_s = \nu_{\min}^{\mathrm{T}}(\bar{\mathbf{p}} - \mathbf{p})c^{-1} \tag{4.28}$$

und damit den Schnittpunkt $\mathbf{p}_s = \mathbf{o} + \mu_s\mathbf{x}$, dessen Tiefenwert dem gesuchten p_z entspricht.

4.4.2 Festlegung des Objekt-Koordinatensystems

Zur Registrierung vieler Modellansichten ist ein gemeinsames Bezugssystem notwendig, das objektfixe Koordinatensystem O. Es spielt für die Lokalisation keine entscheidende Rolle wo oder wie O festgelegt wird, allerdings ist es für den Menschen intuitiver, O in der Mitte oder einer markanten Stelle des Objekts festzumachen. Aus diesem Grund wird in dieser Arbeit ein heuristischer Ansatz verwendet, um eine ansprechende Visualisierung des Bezugsystems in Ansichten zu ermöglichen.

Ausgangspunkt ist eine beliebige Modellansicht, die das Objekt von seiner repräsentativsten bzw. charakteristischsten Seite, bezeichnet als die Oberseite zeigt. Dies ist in dieser Arbeit immer die erste Ansicht $\mathrm{m}_1(\cdot)$, die dem System während des Trainings gezeigt wird und die vom Benutzer so gewählt werden sollte, dass die Oberseite zur Kamera gerichtet ist. In dieser Ansicht wird eine Bildregion $\mathcal{R}_O$ gewählt, die approximativ das gesamte projizierte Objekt enthält. Dies kann entweder über eine Segmentierung erfolgen oder über die konvexe Hülle aller Mittelpunkte der detektierten Modell-Regionen. Mittels des Verfahrens aus Abschnitt 4.4.1 wird nun für $\mathcal{R}_O$ die Ebenenparameter und die Tiefe p_z des Regionenschwerpunkts $\bar{\mathbf{u}}$ bestimmt.

Das Objekt-Koordinatensystem O lässt sich nun bzgl. des Ansicht-Systems M_1 festlegen. Als Mittelpunkt ${}^{\mathrm{M_1}}\mathbf{o} = \mathbf{h}^{-1}(\bar{\mathbf{u}}, p_z)$ wird der rückprojizierte Regionenschwerpunkt $\bar{\mathbf{u}}$ verwendet, als Orientierung ${}^{\mathrm{M_1}}\mathbf{R}_{\mathrm{O}}$ die inverse Matrix aus Gl. 4.2, definiert durch die Ebenenparameter von $\mathcal{R}_O$. Dies führt dazu, dass die XY-Ebene des Objekt-Koordinatensystem O an der dominantesten Ebene der Objektansicht ausgerichtet ist. Die Z-Achse zeigt dabei von der Kamera weg. Sollte dies nicht der Fall sein, wird die Orientierung um 180° um die X-Achse gekippt.

4.4.3 Registrierung mehrerer Modellansichten

Eine Modellansicht beschreibt ein Objekt von einem bestimmten Blickwinkel und erlaubt in Abhängigkeit von dem kovarianten Detektor eine Lokalisation unter einer bestimmten Blickwinkeländerung. Abschnitt 3.2.6 zeigt, dass sich die kritischen Änderungen auf eine Verkippung des Objekts aus der Ansichtsebene heraus beziehen. Diese werden von den affin kovarianten Detektoren zwar teilweise modelliert, aber nur mit Hilfe von Approximationen, die auf reale Objekte und Abbildungen nur eingeschränkt zutreffen. Alle weiteren Änderungen, beschreibbar durch eine ST in der Ansichtsebene werden theoretisch von den Detektoren abgedeckt, allerdings kann es aufgrund der verwendeten Beleuchtung bei Objekten mit schwacher Textur bzw. Struktur selbst bei einer ST zu nicht modellierten Beleuchtungsvariationen und damit zu Schwierigkeiten bei der Lokalisation kommen.

In Abhängigkeit des gewünschten Abdeckungsbereichs der Lokalisation und des Objekts ist es daher notwendig, mehrere Ansichten in dem Modell abzulegen und diese auch miteinander zu registrieren. Das Objekt-Koordinatensystem O wird wie in Abschnitt 4.4.2 beschrieben anhand der ersten Ansicht $\mathrm{m}_1(\cdot)$ festgelegt. Zur Aufnahme und Registrierung aller weiteren Modellansichten wird in dieser Arbeit der in Abschnitt 1.3 vorgestellte 6 DoF Industrieroboter zu Hilfe genommen, es wird allerdings auch eine Möglichkeit skizziert diese Aufgabe mit Hilfe von Benutzerinteraktionen zu lösen.

Training mit Hilfe des Roboters Die Verwendung eines Roboters erlaubt ein nahezu autonomes Training des Objektmodells. Ein Benutzer muss das Objekt nur noch auf einem geeigneten Hintergrund (siehe Abb. 4.5) platzieren und den gewünschten Abdeckungsbereich der Blickwinkeländerungen angeben. Der Roboter fährt dann auf einer Sphäre um das Objekt herum und nimmt Ansichten unter verschiedenen Verkippungswinkeln auf. Optional können auch noch verschiedene Verschiebungen, Rotationen und Skalierungen in den verkippten Ansichtsebenen aufgenommen werden, allerdings besteht dabei sehr schnell die Gefahr, dass die Anzahl der Regionen in der Modelldatenbank und damit die Laufzeit bei der Korrespondenzfindung extrem ansteigt (vgl. dazu auch Abschnitt 6.1.2.3).

Die Registrierung erfolgt über die Kinematik des Roboters, die die Lage ${}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP}}$ des aktuellen TCP[5] in Roboterbasis-Koordinaten BASE liefert. Die Sensorik ist fest an dem TCP montiert und daher sind über die Kalibrierdaten die Transformationen ${}^{\mathrm{TCP}}\check{\mathbf{H}}_{\mathrm{C_i}}$ der einzelnen Kameras bekannt. Zum Aufnahmezeitpunkt 1 der ersten Modellansicht steht der TCP an der Lage ${}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP},1}$, womit sich das Objekt in Roboterbasis-Koordinaten

$$ {}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{O}} = {}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP},1}\,{}^{\mathrm{TCP}}\check{\mathbf{H}}_{\mathrm{C_i}}\,{}^{\mathrm{M_1}}\check{\mathbf{H}}_{\mathrm{O}} \tag{4.29} $$

angeben lässt. Das Koordinatensystem M_1 von $\mathrm{m}_1(\cdot)$ entspricht dabei dem Kamera-Koordinatensystem C_i der dafür verwendeten Kamera. Sei j der Aufnahmezeitpunkt und $i(j)$ die verwendete Kamera der Modellansicht $\mathrm{m}_j(\cdot)$, dann erhält man die gesuchte Registrierung mit Hilfe der Roboterkinematik über:

$$ {}^{\mathrm{M_j}}\check{\mathbf{H}}_{\mathrm{O}} = ({}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP},j}\,{}^{\mathrm{TCP}}\check{\mathbf{H}}_{\mathrm{C_{i(j)}}})^{-1}\,{}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{O}} \tag{4.30} $$

[5] engl.: tool center point

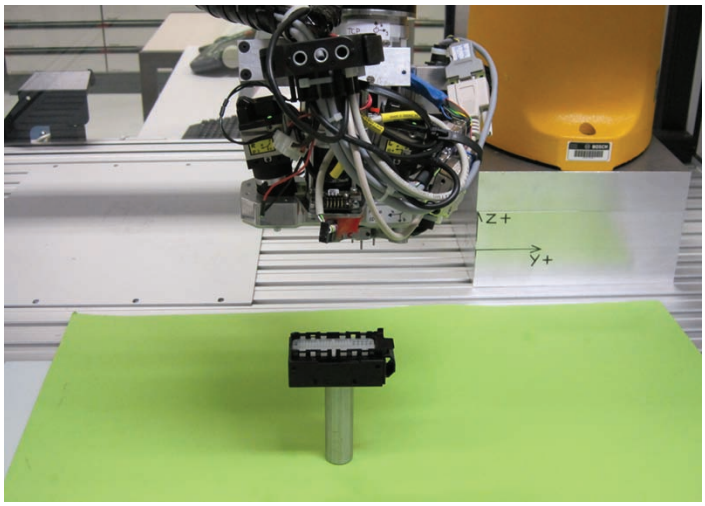

Abbildung 4.5: Training des Steckers auf einem homogenen Hintergrund mit Hilfe des Roboters. Der Metallständer dient nur der Sicherheit des Sensorkopfs, falls der Roboter seinen Ellenbogen umkonfiguriert.

Die Genauigkeit der Registrierung hängt dabei von der absoluten Verfahrgenauigkeit des Roboters ab. Sie ist bei dem eingesetzten Roboter aus Abschnitt 1.3 im Bereich von ca. $100 - 300\,\mu$m und daher so gut wie vernachlässigbar.

Training durch Benutzerinteraktion Steht kein Roboter zu Verfügung, muss ein Benutzer durch entsprechende Interaktion für Blickwinkeländerungen sorgen. Dies kann dadurch geschehen, dass er entweder die Kamera oder das Objekt bewegt. Dabei ist die Bewegung der Kamera vorzuziehen, da dann das Objekt weiterhin auf einem günstigen Hintergrund platziert werden kann. Die Bewegung der Kamera kann festgestellt werden, indem entweder ein marker-basiertes System auf dem Hintergrund verwendet wird oder mit dem bereits bestehendem Objektmodell der schon eingelernten Ansichten und der in dieser Arbeit entwickelten Lokalisation gearbeitet wird.

Im zweiten Fall kann über das absolute Qualitätsmaß geschätzt werden, ob die Registrierung, d. h. die Lage der aktuellen Ansicht bzgl. des bestehenden Modells gut geschätzt wurde oder nicht. Ist das Maß über einen gewissen Schwellwert, so wird die Ansicht dem Modell hinzugefügt, ansonsten nicht. Allerdings kann so weder das Objekt systematisch eintrainiert werden noch eine Ansicht außerhalb des Einzugsbereichs der bestehenden Ansichten aufgenommen werden. Die Registrierung erfolgt daher von Ansicht zu Ansicht inkl. einer ungünstigen Fehlerfortpflanzung. Aus diesem Grunde wird nur die Registrierung über die Roboterkinematik in dem Projekt und daher auch in dieser Arbeit verfolgt.

4.5 Erweiterungen

Das grundlegende Konzept der Objektlokalisation ist in Abschnitt 4.3 beschrieben. Es behandelt alle notwendigen Schritte von der Bildaufnahme bis zur Ausgabe der 6 DoF Lage. Das Verfahren ist allerdings nicht auf diese Vorgehensweise beschränkt, sondern hat Potential für Erweiterungen, die optional in das System integriert werden können.

Der folgende Abschnitt beschäftigt sich mit der Bewertung von Regionen zur Laufzeit und mit dem Problem von ähnlichen Textur- oder Strukturelementen auf einem Objekt, welches bis zu diesem

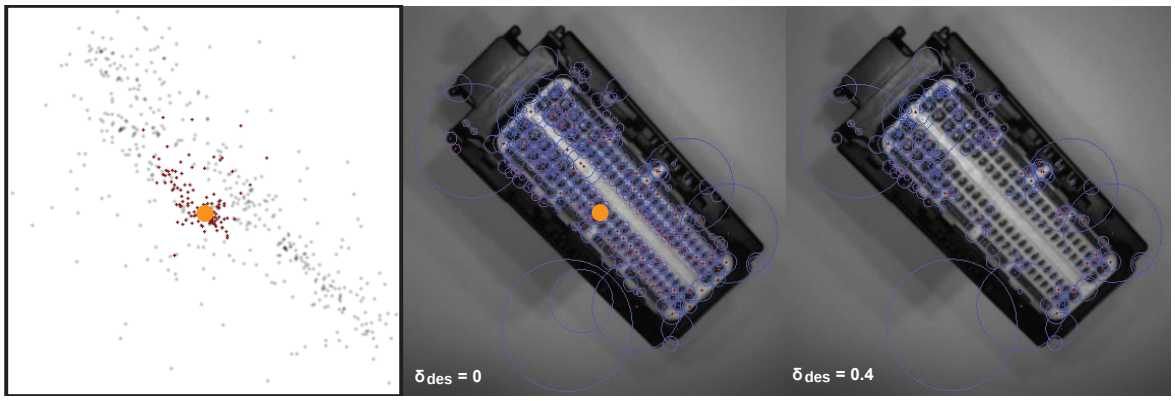

Abbildung 4.6: Eindeutigkeitsproblematik bei ähnlichen Objektstrukturen des Steckers während der Korrespondenzfindung. Links sind die Ursprünge der lokalen Lagen nach der Korrespondenzzuordnung für die detektierten Regionen im mittleren Bild dargestellt, orange der Ursprung der Ground-Truth-Lage. Die Trainingsansicht (nicht abgebildet) wurde exakt von oben aufgenommen. Leicht zu erkennen ist das verschmierte Cluster der lokalen Lagen, welches sich auf die Verwechslung der ähnlichen Regionen (die Löcher des Steckers) während der Korrespondenzfindung zurückführen lässt. Rechts sind dieselben Regionen abgebildet, allerdings wurden alle mehrdeutigen Regionen entfernt.

Zeitpunkt noch nicht adressiert wurde. In einem weiteren Abschnitt werden die Regionen der Modelldatenbank auf ihre Eignung während der Lokalisation untersucht und gegebenenfalls das Modell optimiert. Evaluiert werden beide optionalen Erweiterungen innerhalb des Kapitels 6.

4.5.1 Bewertung von Regionen während der Lokalisation

Eine Bewertung der gefundenen Regionen zur Laufzeit ist wesentlich schwieriger als während des Trainings, da auf keine Referenzergebnisse zurückgegriffen werden kann. Eine Möglichkeit besteht über die Konstruktionskriterien der Detektoren. Über Schwellwerte (siehe Abschnitt 3.2) lässt sich während der Detektion die Sensibilität bzgl. ausgeprägter Bildstrukturen beeinflussen und damit letztendlich auch die Güte der gefundenen Regionen. Andere Möglichkeiten ergeben sich nach der Korrespondenzfindung durch die Konstruktionsgüte der Modellregion, der Merkmalsdifferenz beider Regionen oder der Bewertung der Modellregion während des Trainings.

Diese Sichtweise beschäftigt sich allerdings immer nur mit einer einzelnen Such-Region. Es gibt auch eine Bewertungsgrundlage durch Betrachtung der Gemeinsamkeiten über alle Such-Regionen. Je ähnlicher sich zwei Regionen innerhalb einer Suchansicht sind, desto größer ist die Wahrscheinlichkeit einer Verwechslung während der Korrespondenzzuordnung. Dieser Effekt ist in Abb. 4.6 dargestellt. Die einfachste Möglichkeit besteht in der Eliminierung zu ähnlicher Regionen über einen Schwellwert $\delta_{\mathbf{des}}$. Die Ähnlichkeit wird dabei über das Distanzmaß $d_{\mathbf{des}}(\cdot)$ definiert. Die Auswirkungen dieser (optionalen) Maßnahme sind ebenfalls in Abb. 4.6 zu sehen. Diese Herangehensweise birgt aber eine Gefahr. Befinden sich mehrere Objekte unter einem ähnlichen Blickwinkel im Bild, können im schlimmsten Fall so viele Such-Regionen eliminiert werden, dass keine robuste Lokalisation mehr möglich ist.

Eine andere Möglichkeit Mehrdeutigkeiten zu adressieren, bezieht sich direkt auf die Korrespondenzfindung. Anstatt nur die ähnlichste Modell-Region über den NN im Merkmalsraum zu suchen, können auch die k_{NN} ähnlichsten Modell-Regionen über den kNN ermittelt werden. Dies führt zwar einerseits zu mehr Fehlkorrespondenzen, erhöht andererseits aber auch die Wahrscheinlichkeit einer korrekten Zuordnung bei Mehrdeutigkeiten. Das System wird dabei aber stark ausgebremst, da

sich die Anzahl der zu verarbeitenden Korrespondenzen um den Faktor k_{NN} erhöht. Der folgende Abschnitt stellt daher eine effizientere Methode vor, die zwar mehr Informationen während des Trainings benötigt, dafür den kNN aber nur noch bei verwechselbaren Such-Regionen anwendet.

4.5.2 Optimierung der Modelldatenbank

Der wichtigste Teil eines Modells ist seine Datenbank an Modell-Regionen, die die Textur und Struktur des Objektes an salienten Stellen kodieren. In ihr sind alle Regionen aller Modellansichten enthalten, die im Detektorschritt gefunden wurden. Auch wenn die Modellansichten unter möglichst guten Bedingungen aufgenommen werden, so sind Schattenwurf, Glanzpunkte bzw. eine fehlerhafte Segmentierung bei einem autonomen Ablauf und damit fehlerhafte Regionen nicht zu vermeiden. Weiterhin enthält die Datenbank ebenfalls ähnliche, d. h. leicht verwechselbare Regionen. Ist dies bei unterschiedlichen Modellansichten durchaus beabsichtigt, da sie denselben Objektausschnitt aus einem anderen Blickwinkel beschreiben, kann dies innerhalb einer Suchansicht zu einem Problem führen. Abhilfe schafft auch in diesem Fall eine kNN Korrespondenzfindung zur Laufzeit oder eine Eliminierung wie in Abschnitt 4.5.1 beschrieben. Dies geht aber zur Lasten der Laufzeit oder Robustheit.

Während des Trainings stehen allerdings weitere Möglichkeiten offen, um eine ausführliche Bewertung der Modell-Regionen vorzunehmen und damit feinere Maßnahmen ergreifen zu können. Die Idee dabei ist, die Lage ${}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{O}}$ aus Gl. 4.29 als Referenzlage zu verwenden und über bekannte Blickwinkeländerungen die lokalen Lagen der Korrespondenzen damit zu vergleichen. Dies ist allerdings in der hier beschriebenen Form nur möglich falls das Training mittels Roboter verwendet wird und das Objekt während des gesamten Trainingsvorgangs inkl. der Bewertung nicht bewegt wird.

Sei das TCP-Koordinatensystem gleich dem Welt-Koordinatensystem W der Kameras, dann lässt sich die wahre Lage des Objekts in W unter einem neuen Blickwinkel an der Position ${}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP}}$ durch ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}} = {}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{TCP}}^{-1}\,{}^{\mathrm{BASE}}\check{\mathbf{H}}_{\mathrm{O}}$ des Roboters angeben. Es erfolgt unter dem neuen Blickwinkel eine normale Lokalisation, allerdings wird während des Clusterschritts nicht nach Häufungen der lokalen Korrespondenzen gesucht, sondern nur das Referenzcluster $\mathcal{G}_R$ mit Zentrum ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ bestimmt. Wird diese Vorgehensweise für viele verschiedenen Blickwinkel durchgeführt, lässt sich damit eine Statistik über die Modell-Regionen aufbauen. Für jede Modellregion $\mathcal{R}_k$ werden zwei Größen bestimmt, die Anzahl $a(k)$ wie oft die Modell-Region einer beliebigen Such-Region während der Korrespondenzfindung zugeordnet wird und die Anzahl $b(k)$ wie oft die daraus ermittelte lokale Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},k}$ dem Referenzcluster $\mathcal{G}_R$ zugehörig ist. Darüber können die Modell-Regionen in mehrere Klassen unterteilt und unterschiedlich behandelt werden:

- Ist $a(k) < \tau_a$ sehr klein, handelt es sich bei $\mathcal{R}_k$ um eine Region, die nur unter ganz bestimmten Blickwinkeln mit einer Such-Region korrespondiert. Dies deutet auf einen Glanzpunkt oder einen Schattenwurf während des Trainings hin und ist daher für die Lokalisation (meist) unbrauchbar. $\mathcal{R}_k$ wird daher aus der Modelldatenbank entfernt, um die Korrespondenzzuordnung zu beschleunigen.

- Ist das Verhältnis $\frac{b(k)}{a(k)} < \tau_r$ klein, wird $\mathcal{R}_k$ zwar häufig zugeordnet, allerdings meistens falsch. Findet sich innerhalb derselben Modellansicht eine sehr ähnliche Region, deutet dies auf eine häufige Verwechslung, ansonsten auf eine schwache zugrunde liegende Objektstruktur hin. In beiden Fällen wird $\mathcal{R}_k$ in der Datenbank belassen, im ersten Fall als Mehrdeutigkeit markiert, im zweiten Fall als schwache $\mathcal{R}_k$. Die schwachen Regionen werden zwar bei der Korrespondenzfindung berücksichtigt um andere Fehlzuordnungen zu vermeiden, allerdings werden die lokalen Hypothesen im Anschluss verworfen. Bei mehrdeutigen Modell-Regionen wird für die *Such-Region* nach der ersten gefundenen Korrespondenz nochmals eine kNN-Anfrage durch-

geführt, im Gegensatz zu Abschnitt 4.5.1 allerdings diesmal nur wenn es tatsächlich sinnvoll ist.

- Ist das Verhältnis $\frac{b(k)}{a(k)} \geq \tau_r$ dagegen groß, handelt es sich um eine verlässliche Region die genau so in der Datenbank erhalten bleibt. Allerdings kann über den gemittelten Fehler bzgl. ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ eine Güte angegeben werden und während des Clusterns bzw. während der globalen Lagebestimmung als Gewicht genutzt werden.

COMPUTER GEHORCHEN
DEINEN BEFEHLEN,
NICHT DEINEN ABSICHTEN.

Unbekannter Autor

Kapitel 5

Softwaretechnische Umsetzung

Dieses Kapitel gibt einen Überblick über die gesamte, im Rahmen dieser Arbeit entstandene Software und verknüpft die schriftliche Ausarbeitung mit den entwickelten SW-Komponenten. Bei der Umsetzung wurde ein besonderes Augenmerk auf eine performante Implementierung inkl. einer vollständigen Parallelisierung aller laufzeitkritischen Komponenten gelegt. Die Umsetzung folgt einem streng objektorientiertem Design in der Sprache C++ und erlaubt einen flexiblen Einsatz der Software sowie eine einfache Erweiterbarkeit. Die zentrale Struktur ist dabei die Klasse, sie wird im Quellcode mit einem vorangestellten `C...` gekennzeichnet. Schnittstellen ohne vollständige Implementierung werden mit *abstrakten* Klassen realisiert und im Quellcode mit `CAbstract...` bezeichnet. Methoden und Variablen werden klein geschrieben. Zur Parallelisierung wird *OpenMP* (Ope10) verwendet, als zentrale Bildverarbeitungs- und Mustererkennungsbibliotheken *OpenCV* (Ope09), *IVT* (IVT10) sowie *Halcon* (MVT10).

Auf der obersten Abstraktionsebene lassen sich in Abb. 5.1 die drei Software-Pakete identifizieren. Sie beschäftigen sich mit dem erstellten Framework FRoVis[1], der Hardwareansteuerung und der in dieser Arbeit vorgestellten Objektlokalisation. Jedes der SW-Pakete besitzt mindestens eine zentrale Schnittstelle: `CRVApplication`, `CAbstractModule`, `CAbstractModel` bzw. `CAbstractRobot`. Die Klasse `CRVApplication` bündelt dabei die gesamte Funktionalität und ist die einzige Schnittstelle zu einer übergeordneten, externen Anwendung. In den folgenden drei Abschnitten werden die einzelnen SW-Pakete detaillierter auf Klassenebene vorgestellt und der Bezug der Implementierung zu der vorliegenden schriftlichen Ausarbeitung hergestellt.

5.1 Framework FRoVis

Das Framework FRoVis beinhaltet alle Klassen zur Verwaltung der entstandenen Software und beinhaltet u. a. Funktionalität zum Fehlermanagement, Logging, Visualisierung, Konfigurationsmanagement und zur Modellverwaltung. Es ist im Rahmen dieser Arbeit entstanden und wird mittlerweile innerhalb des Projekts als zentrales Bildverarbeitungs-Framework gepflegt und erweitert. Es ist (ausschnittsweise) in Abb. 5.2 als Klassendiagramm dargestellt.

FRoVis beinhaltet die Schnittstelle `CAbstractModule`, von der alle wichtigen Klassen der drei Pakete abgeleitet sind (ohne Darstellung in den Abbildungen). `CAbstractModule` ermöglicht Zugang zu den elementaren Funktionalitäten von FRoVis. Dies ist über die Schnittstelle `CAbstractDataLogger` das zentrale Logging von Informationen und Fehlern, über `CAbstractVisualization` die Visualisierung von 2D-Bildern (`CProcessVisualization`) bzw. mittels *CluViz* (Ray10) auch 3D-Daten (`CCLUVizContainer`) sowie das Konfigurationsmanagement. Letzteres kann über das globale Singleton `CLibConfig` von allen Klassen erreicht werden.

[1] Framework Robot Vision

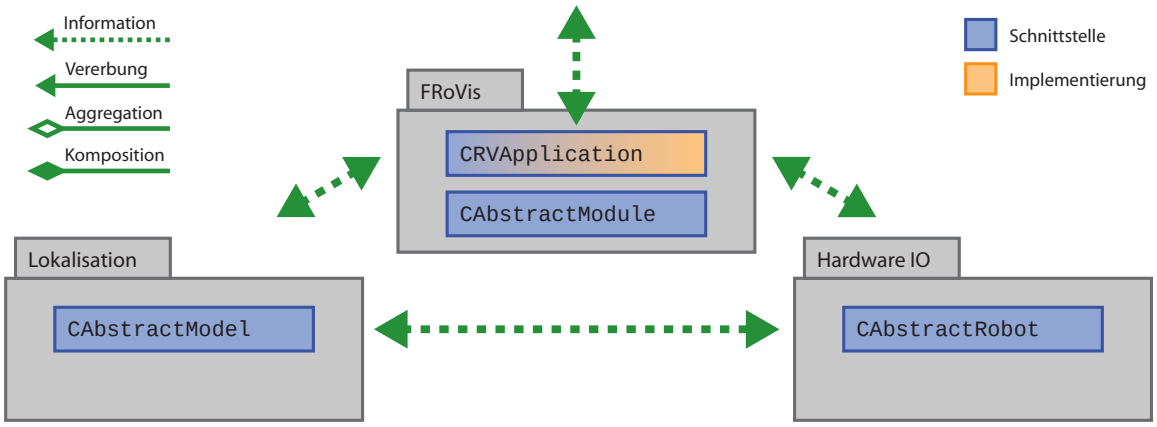

Abbildung 5.1: Darstellung der drei übergeordneten Software-Pakete und ihrer zentralen Schnittstellen.

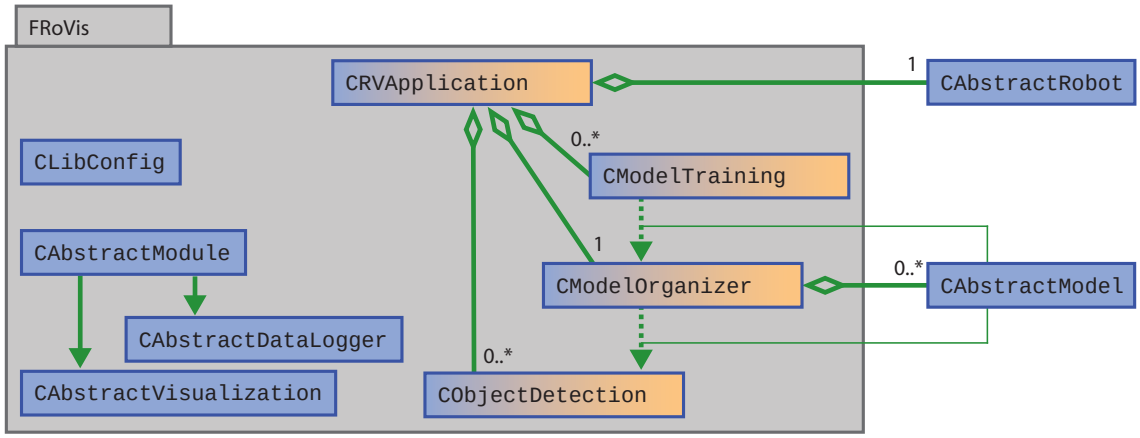

Abbildung 5.2: UML-Klassendiagramm (Ausschnitt) des Frameworks FRoVis.

`CRVApplication` ist die zentrale Klasse der gesamten Software und stellt Methoden zur Initialisierung, Finalisierung, zum Modelltraining und zur Objektlokalisation einer übergeordneten Anwendung zur Verfügung. Sie enthält und initialisiert die Schnittstelle `CAbstractRobot` zur Hardware und verwaltet über die Klasse `CModelOrganizer` alle Modelle. Es können beliebig viele Modelle simultan trainiert bzw. Objekte lokalisiert werden. Dafür wird jeweils eine Klasse `CModelTraining` oder `CObjectDetection` instantiiert, in der der weitere Ablauf definiert ist. Beide Klassen greifen dazu auf die Schnittstelle `CAbstractModel` zurück, in der die gesamte Trainings- bzw. Lokalisationsfunktionalität der Modelltypen spezifiziert ist.

Die Klasse `CModelOrganizer` verwaltet alle bekannten Modelle sowohl persistent auf der Festplatte als auch je nach Zugriff im Hauptspeicher. In jedem Trainingsvorgang wird durch `CModelTraining` ein neues Modell durch eine `CAbstractModel`-implementierende Instanz erzeugt, mit einer eindeutigen GUID versehen und bei erfolgreichem Training der `CModelOrganizer`-Instanz übergeben. Diese legt das Modell dann auf der Festplatte ab, lädt es bei Bedarf wieder in den Hauptspeicher und stellt es dann einer `CObjectDetection`-Instanz zur Verfügung.

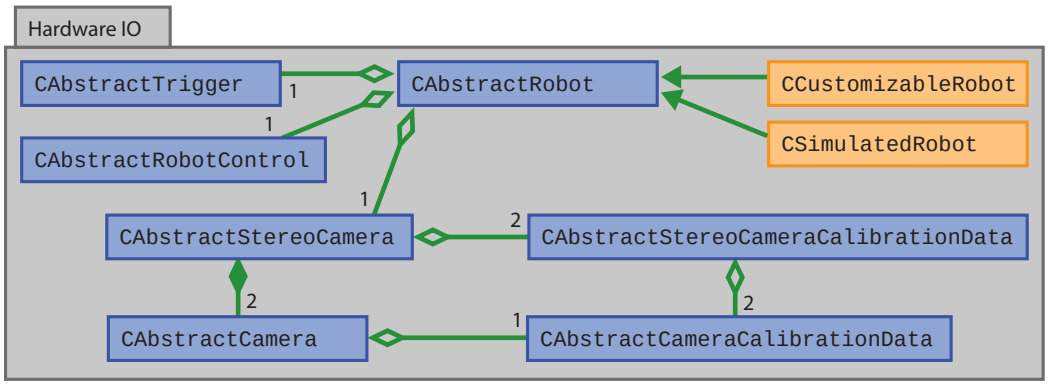

Abbildung 5.3: UML-Klassendiagramm (Ausschnitt) für die Hardwareansteuerung.

5.2 Hardwareansteuerung

Die Funktionalität der Hardwareansteuerung wird in `CAbstractRobot` gebündelt und ist in der Abb. 5.3 (ausschnittsweise) dargestellt. Der Begriff Roboter verkörpert in diesem Zusammenhang die gesamte Sensorik und Aktorik des Systems.

`CAbstractRobot` besitzt die Schnittstelle `CAbstractRobotControl` zur Robotersteuerung, welche u. a. Zugriff auf die Roboterlage ${}^{\text{BASE}}\check{\mathbf{H}}_{\text{TCP}}$ bietet. Verfahren kann man den Roboter damit allerdings nicht, dies geschieht aus Sicherheitsgründen durch die übergeordnete, externe Anwendung.

Weiterhin enthält `CAbstractRobot` eine Schnittstelle `CAbstractStereoCamera` zu der auf dem Roboter montierten Stereokamera. Diese besteht aus zwei Einzelkameras, die wiederum über `CAbstractCamera` angesteuert werden können. `CAbstractStereoCamera` enthält zweimal die Schnittstelle `CAbstractStereoCameraCalibrationData`, jeweils einmal zu den *unrektifizierten* und den *rektifizierten* Kalibrierdaten (vgl. Anhang C.2). In beiden Fällen stehen die intrinsischen und extrinsischen Kalibrierdaten jeweils für die rechte und linke Kamera in der untergeordneten Schnittstelle `CAbstractCameraCalibrationData` zur Verfügung. Letztere enthalten u. a. die Modelle $\mathbf{h}(\cdot)$ bzw. $\mathbf{h}^{-1}(\cdot)$ der jeweiligen Kamera. Das Weltmodell W ist im unrektifizierten Fall der TCP des Roboters, im rektifizierten Fall das Kamera-Koordinatensystem C der *unrektifizierten* Kamera. `CAbstractCameraCalibrationData` stellt ebenfalls Funktionen zur Ver- und Entzerrung der Kamerabilder zur Verfügung. Im unrektifizierten Fall werden damit Linsenverzerrungen ausgeglichen, im rektifizierten Fall die Rektifizierung berechnet.

Alle Sensordaten werden synchron in einem Zeitschritt verarbeitet. Dazu ist in `CAbstractRobot` eine Schnittstelle `CAbstractTrigger` vorhanden, deren Implementierung in einem vorbereitendem Schritt einen Triggerimpuls zur Datenaufnahme an die Kameras, die Beleuchtung und die Robotersteuerung schickt. Während des eigentlichen Zeitschritts werden dann alle Sensordaten, wie Bilder und Roboterlage aus den Hardwarekomponenten ausgelesen und in den jeweiligen Softwarekomponenten in einem Ringpuffer gespeichert. Dem System stehen danach die synchronen Daten der aktuellsten n_t Zeitschritte zur Verfügung. Alle Parameter zur Konfiguration der Sensoren, wie z. B. die Belichtungszeit einer Kamera, werden für jede Komponente in einer abgeleiteten Klasse von `CSensorParameters` gekapselt und für den gesamten Roboter in `CRobotSensorParameters` gebündelt. Jede `CAbstractModel`-Instanz enthält einen eigenen Parametersatz und kann damit die Datenaufnahme individuell vor dem Triggerimpuls konfigurieren.

Es existieren z. Z. zwei Implementierungen von `CAbstractRobot`. `CCustomizableRobot` realisiert einen konfigurierbaren Roboter, der für alle Schnittstellen Implementierungen zur Ansteuerung realer Hardware zur Verfügung stellt. Die Triggerung wird dabei über `CWascoTrigger` mittels einer

WASCO-Karte erzeugt, die Kameraansteuerung über `CHalconStereoCamera` bzw. `CHalconCamera` mittels Halcon und die Roboteransteuerung über `CExternalFillRobotControl` mittels einer projektspezifischen externen Anwendung. Die Kalibrierung erfolgt ebenfalls durch eine externe Anwendung mittels Halcon und wird über `CHalconStereoCameraCalibrationData` bzw. `CHalconCameraCalibrationData` aus abgelegten Dateien importiert.

Des Weiteren existiert mit `CSimulatedRobot` eine Implementierung von `CAbstractRobot` zu Entwicklungszwecken. Sie simuliert einen realen Ablauf durch auf der Festplatte gespeicherte Datensequenzen. Dazu kann in `CCustomizableRobot` ein `CRobotRecorder` instantiiert werden, der die Daten jedes Zeitschritts protokolliert und in chronologischer Reihenfolge auf der Festplatte ablegt.

5.3 Lokalisation

Innerhalb des SW-Pakets Lokalisation wird das gesamte in Kap. 4 entworfene Verfahren umgesetzt. Das Paket ist allerdings so allgemein gehalten, dass weitere, innerhalb des Projekts entwickelte Verfahren auf einfache Weise integriert werden können und die gesamte Funktionalität von FRoVis nutzen können. Einen Überblick über die wichtigsten Klassen ist in Abb. 5.4 zu sehen.

`CAbstractModel` ist die zentrale Schnittstelle und kapselt den Zugriff auf alle Modelldaten inkl. Laden und Speichern von der Festplatte sowie die Funktionalität des Trainings. Jedes eigenständige Verfahren ist ein eigener Modelltyp und leitet von `CAbstractModel` eine eigene Klasse ab, in der die konkrete Umsetzung implementiert ist. Für die 6 DoF Lageerkennung mit lokalen, kovarianten Merkmalen ist dies `CAffineLocalFeatureModel` und davon abgeleitet `CSimpleAffineLocalFeatureModel`. Ersteres enthält die gesamte Funktionalität, Letzteres dient nur als Factory-Klasse und erzeugt je nach Konfiguration unterschiedliche, aber konkrete Implementierungen für alle benötigten Schnittstellen. Eine Besonderheit ist der Modelltyp `CALFMetaModel`. Er ist eine Ergänzung von `CAffineLocalFeatureModel` um die in Abschnitt 4.5.2 vorgestellten konzeptionellen Erweiterungen, die bei Bedarf aktiviert werden können.

`CAffineLocalFeatureModel` enthält eine Schnittstelle `CAbstractIndexStructure` auf die Datenbank der lokalen Merkmale, wobei deren Deskriptorvektor $\mathbf{o}$ als hochdimensionaler Index innerhalb der DB verwendet wird. `CAbstractIndexStructure` ermöglicht über diesen Index den Zugriff auf die Merkmale, insbesondere durch eine NN- bzw. kNN-Anfrage. Im Laufe dieser Arbeit sind vier unterschiedliche Implementierungen entstanden: `CSimpleArrayIndexStructure` realisiert eine Bruteforce-Anfrage über alle Merkmale der DB, allerdings hochoptimiert mittels Assembler, SSE3 SIMD-Befehlen und vollständig parallelisiert; `CSimpleCUDAIndexStructure` setzt dieselbe Funktionalität nach einer Implementierung von (GDB08; Zoe10) auf der GPU um; `CVAFileIndexStructure` nutzt die Methodik des VA-Files (Tra09) und `CXTreeIndexStructure` nutzt einen hierarchischen X-Tree (Tra09).

`CAbstractModel` beinhaltet die drei Schnittstellen `CAbstractFeatureExtractor`, `CAbstractStereoDepthExtractor` und `CAbstractFeatureMatcher`, die den Zugriff auf separate Funktionalität kapseln und in den folgenden Paragraphen behandelt werden. Dabei wird über ein Flag unterschieden, ob die Funktionalität während des Trainings oder der online Lokalisation benötigt wird. Bis auf wenige Abhängigkeiten sind alle im Folgenden beschriebenen Verfahren beliebig miteinander kombinierbar.

`FeatureExtractor` Die Schnittstelle `CAbstractFeatureExtractor` kapselt die gesamte Funktionalität die sich mit der Verarbeitung eines *einzelnen* Bildes beschäftigt. Für die 6 DoF Lokalisation bedeutet dies die Verarbeitung von lokalen, kovarianten Regionen, deren Beschreibungen jeweils in einer Instanz der Klasse `CAffineLocalFeature` unabhängig von dem Verfahren gekapselt sind. Dies sind u. a. der Merkmalsvektor $\mathbf{o}$ in `index`, das Zentrum $\mathbf{u} = (u,v)^{\mathrm{T}}$ sowie der zugehörige Tiefenwert

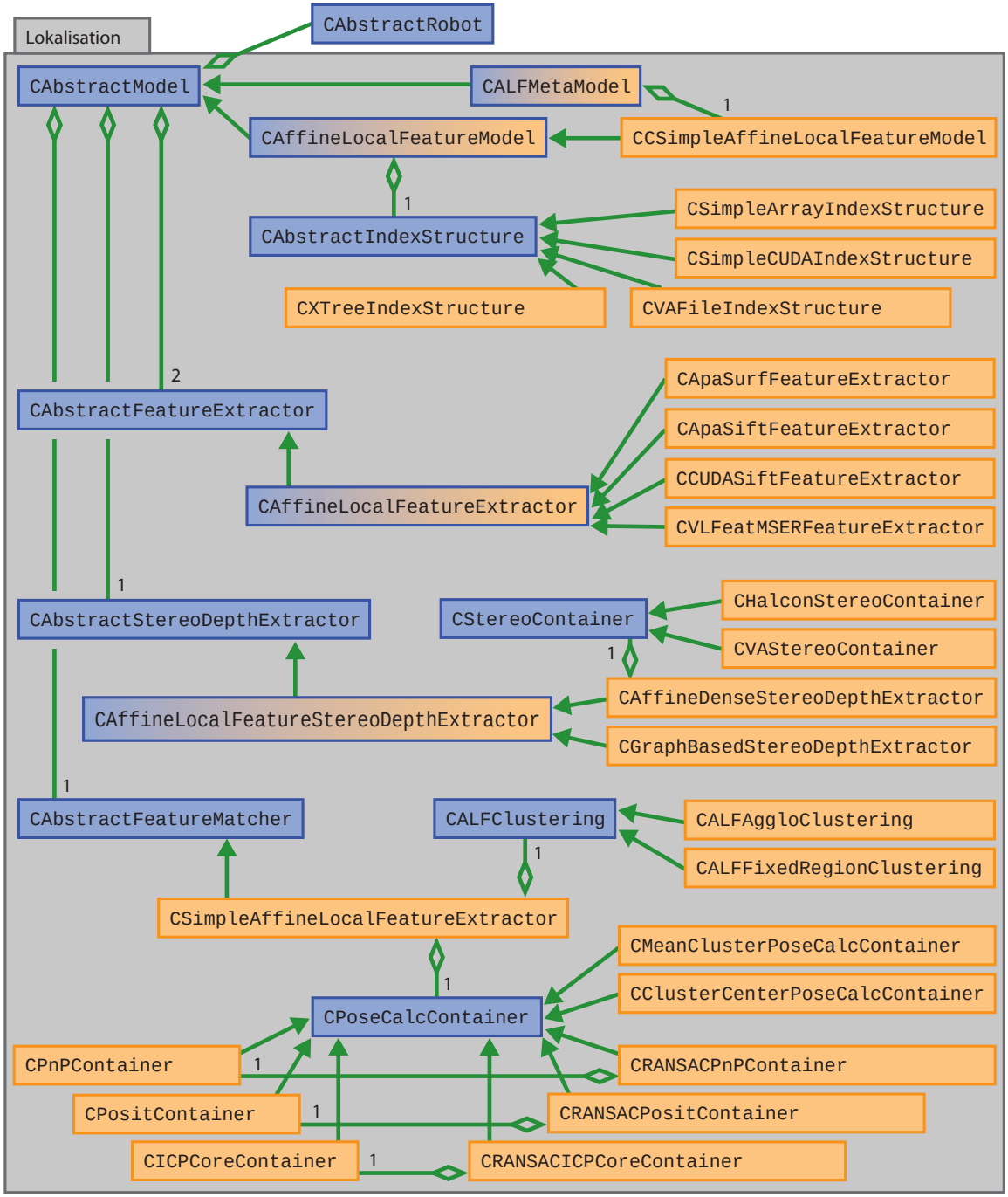

Abbildung 5.4: UML-Klassendiagramm (Ausschnitt) der Objektlokalisation.

p_z in `u`, `v`, `depth`, die Matrix $\mathbf{A}_N$ bzw. **KD** aus Abschnitt 4.3.1 in `CV_mA_N` bzw. `CV_mA_UCV` sowie die Ebenenparameter in `mu` bzw. `mv`.

Die Detektion und Deskription der Regionen wird allgemein in `CAffineLocalFeatureExtractor` definiert und dann konkret von abgeleiteten Klassen implementiert. Es sind im Rahmen dieser Arbeit mehrere Verfahren mit verschiedenen Detektoren entstanden, allerdings verwenden alle SIFT als Deskriptor: `CApaSiftFeatureExtractor` setzt eine (interne) DoG-Implementierung als

Detektor ein; `CCUDASiftFeatureExtractor` realisiert das gleiche Verfahren nach einer Implementierung von (Wu09; Zoe10) auf der GPU; `CApaSurftFeatureExtractor` setzt eine (interne) SURF-Implementierung um und `CVLFeatMSERFeatureExtractor` verwendet eine MSER-Implementierung von (VF10).

Eine Unterscheidung zwischen Training und Lokalisation gibt es bis auf unterschiedliche Parametersätze für die Detektoren nicht.

`StereoDepthExtractor` Die Schnittstelle `CAbstractStereoDepthExtractor` kapselt die Extraktion von Informationen aus *beiden* Bildern. Für das Training bedeutet dies die Berechnung der Tiefe und der Ebenenparameter der zuvor durch `CAffineLocalFeatureExtractor` gewonnenen Regionen. Während der Lokalisation muss in Abhängigkeit der globalen Lageschätzung nichts oder die Tiefe der Regionenzentren berechnet werden.

Allgemein wird diese Aufgabe in der abstrakten Klasse `CAffineLocalFeatureStereoDepthExtractor` definiert. Konkret wurden zwei unterschiedliche Verfahren umgesetzt. Zum einen in `CGraphBasedStereoDepthExtractor` ein Fluss-basiertes dünnes Stereoverfahren (Sch08), welches die korrespondierenden Regionen in beiden Bildern sucht und darüber deren Tiefe ableitet. Zum anderen in `CAffineDenseStereoDepthExtractor` ein Verfahren, welches sowohl Tiefe als auch Ebenenparameter der Regionen mit Hilfe von einem dichten Tiefenbild schätzt. Dazu wird ein `CStereoContainer` eingebunden, dessen Implementierungen aus rektifizierten Stereobildern eine dichtes Tiefenbild berechnen. Konkret existieren z. Z. zwei Umsetzungen: `CVAStereoContainer` verwendet ein intern vorhandenes Verfahren und `CHalconStereoContainer` ein hierarchisches, korrelations-basiertes Verfahren von Halcon.

Während der Lokalisation kann sowohl `CGraphBasedStereoDepthExtractor` als auch `CAffineDenseStereoDepthExtractor` eingesetzt werden. Für das Training steht allerdings nur letztere Umsetzung zur Verfügung, da sie als einziges auch die Ebenenparameter schätzen kann. Während der Lokalisation arbeiten beide Verfahren parallel bis zur Anzahl der übergebenen Regionen.

`FeatureMatcher` Die Schnittstelle `CAbstractFeatureMatcher` bündelt die eigentliche Lokalisation, d. h. die Zuordnung der Informationen aus den zwei `FeatureExtractoren` und dem `StereoDepthExtractor` zu dem Modell und die Ableitung einer oder mehrerer Objektlagen.

Für die 6 DoF Lageerkennung mit lokalen, kovarianten Regionen bedeutet dies in `CSimpleAffineLocalFeatureExtractor` die Zuordnung der Suchregionen zu der Modelldatenbank mit Hilfe der Schnittstelle `CAbstractIndexStructure`, der `CAffineLocalFeatureModel`-Instanz, der Ableitung der lokalen Pose ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ jeder Korrespondenz (parallel implementiert) aus Abschnitt 4.3.1 in der Funktion `calulate3DTransformationFrom2DAffineDistortionOfSOPRegions(...)`, dem Clustern mittels `CALFClustering` und der globalen Lageberechnung ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ jedes gefundenen, in `CALFCluster` gekapseltes Clusters $\mathcal{G}$ durch die Schnittstelle `CPoseCalcContainer`.

Während dieser Arbeit sind zwei verschiedene Implementierungen der `CALFClustering`-Schnittstelle entstanden. `CALFAggloClustering` realisiert ein agglomeratives Clustern der Regionenzentren in der Bildebene. Es wurde aber durch den verbesserten Ansatz im 6D-Raum der Lagen aus Abschnitt 4.3.3.2 in `CALFFixedRegionclustering` ersetzt und wird z. Z. nicht mehr eingesetzt. `CALFFixedRegionclustering` arbeitet parallel bis zur Anzahl der übergebenen lokalen Posen.

Für die Schnittstelle `CPoseCalcContainer` wurden alle in Abschnitt 4.3.4 beschriebenen Verfahren zur Berechnung der robusten, globalen Lage ${}^{\mathrm{W}}\bar{\mathbf{H}}_{\mathrm{O}}$ umgesetzt. Die Klasse `CMeanClusterPoseCalcContainer` implementiert die Lagebestimmung mittels Durchschnittsbildung, `CClusterCenterPoseCalcContainer` die Lagebestimmung mittels Medianbildung, `CICPCoreContainer` die Lagebestimmung mit 3D-3D-Punkt-Korrespondenzen und `CPositContainer` bzw. `CPnPContainer`

die Lagebestimmung mittels 2D-3D-Punkt-Korrespondenzen. Ersteres setzt dabei den POSIT-Algorithmus ein, Letzteres den OI-Algorithmus.

Des Weiteren existieren für die Bestimmung mittels Punkt-Korrespondenzen dieselben Algorithmen inklusive einer ausreißertoleranten RANSAC-Berechnung. Der RANSAC wird in den Klassen `CRANSACICPCoreContainer`, `CRANSACPositContainer` bzw. `CRANSACPnPContainer` umgesetzt und greift auf die ursprünglichen Implementierungen zur Lagebestimmung zurück. Allerdings wird diese dort nur mit der zufällig bestimmten, reduzierten Menge an Korrespondenzen aufgerufen. Die RANSAC-Implementierungen arbeiten parallel bis zur Anzahl der RANSAC-Zyklen.

IN THEORY, THERE IS NO DIFFERENCE
BETWEEN THEORY AND PRACTICE.
BUT, IN PRACTICE, THERE IS.

Jan L.A. van de Snepscheut

Kapitel 6

Experimente und Verifikation

Dieses Kapitel untersucht anhand geeigneter Experimente das in Kap. 4 beschriebene und in Kap. 5 umgesetzte Verfahren zur 6 DoF Lokalisation mit lokalen, kovarianten Regionen. Dabei sollen zuerst in Abschnitt 6.1 die Einflüsse der Parameter auf das System ermittelt werden und ein geeigneter Parametersatz für die nachfolgenden Untersuchungen festgelegt werden. Danach wird in Abschnitt 6.2 das beste Verfahren zur globalen Lagebestimmung ermittelt und im weiteren Verlauf eingesetzt. In den Abschnitten 6.3 und 6.4 werden dann verschiedene Einflussgrößen und Erweiterungen untersucht. Anhand dieser Ergebnisse wird abschließend in Abschnitt 6.5 das sinnvollste System nach Ansicht des Autors festgelegt und ausführlich bzgl. Geschwindigkeit, Genauigkeit und Robustheit evaluiert.

Alle Experimente dieses Kapitels werden mit der in Abschnitt 1.3 vorgestellten Sensorik und Aktorik durchgeführt. Der Roboter dient dabei einerseits zum Einlernen der Objekte und andererseits zur Generierung von Ground-Truth-Daten über gesteuerte und damit bekannte Lageänderungen. Die absolute Genauigkeit des Roboters ist nicht bekannt, sollte aber in dem lokalen Bereich, in dem sich der TCP für die Messungen bewegt hat, unter 0.1 mm liegen. Alle Ergebnisse beziehen sich auf die in Abb. 6.1 dargestellten Objekte.

6.1 Wahl der Parameter

Dieser Abschnitt beschäftigt sich mit der Abhängigkeit des Systems von den einstellbaren Parametern. Ziel ist einerseits ein Gefühl für die Einflussgrößen zu bekommen und andererseits einen optimalen Satz an Parametern festzulegen, mit dem die restlichen Experimente durchgeführt werden können. Dabei ist eine komplette Optimierung über den gesamten Parametersatz aufgrund des exponentiellen Aufwands nicht möglich. Die meisten Parameter werden daher unabhängig voneinander optimiert, obwohl Abhängigkeiten zwischen verschiedenen Einstellungen grundsätzlich nicht ausgeschlossen werden können. Sind diese Abhängigkeiten allerdings bekannt, werden sie an entsprechender Stelle berücksichtigt oder zumindest beschrieben.

Abgeleitet werden die Einstellungen anhand der Objekte Box und Leiterplatte. Jedes Objekt wird mit einer Modellansicht zentral von oben auf einem homogenen Hintergrund eingelernt. Anschließend werden vor dem homogenen Hintergrund jeweils 324 Suchansichten mit dem Roboter aufgenommen, wobei bei diesen über die Roboterposition auch die Objektlage bekannt ist. Die Blickwinkeländerung bzgl. der Modellansicht beschränkt sich dabei auf den Bereich von ca. $\frac{1}{3}$ Bildgröße in der Translation, ± 15 mm in der Tiefe, 45° Rotation in der Bildebene und max. 15° Verkippung aus der Bildebene. Eingelernt wurde in einem Abstand von 170 mm. Die Suchansichten werden als Referenzaufnahmen mit Ground-Truth-Informationen verwendet und dienen zur Bewertung der unterschiedlichen Parametereinstellungen. Beispiele der Ground-Truth-Sequenz findet sich in Abb. 6.2.

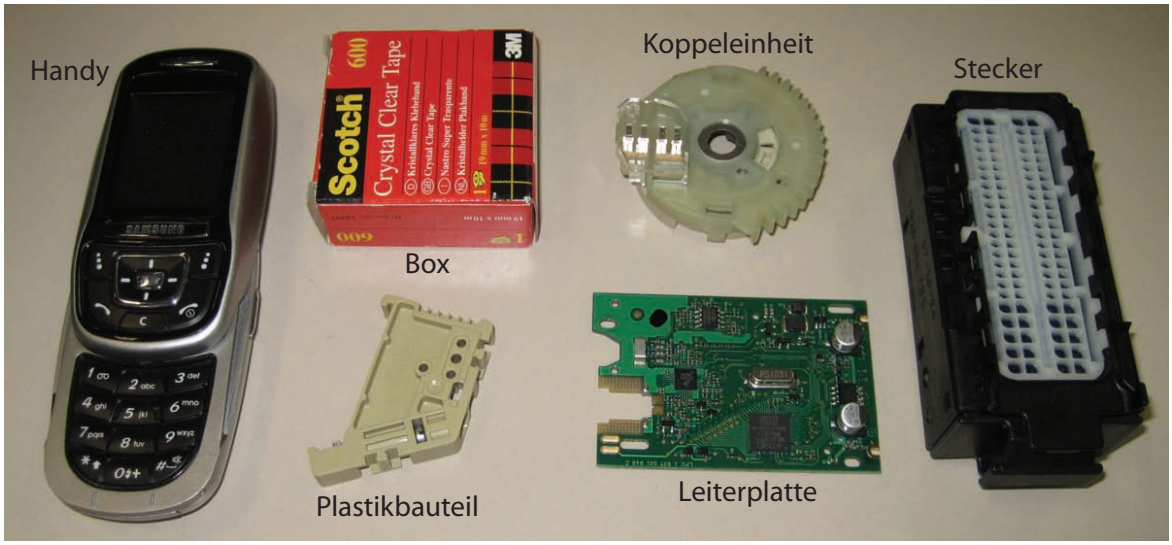

Abbildung 6.1: Alle für die Experimente verwendeten Objekte. Die Dimensionen (Länge × Breite × Höhe) variieren von dem größten Objekt, dem Handy mit Maßen von 118 mm × 44 mm × 21 mm bis zum kleinsten Objekt, dem Plastikbauteil mit Maßen von 45 mm × 27 mm × 7 mm.

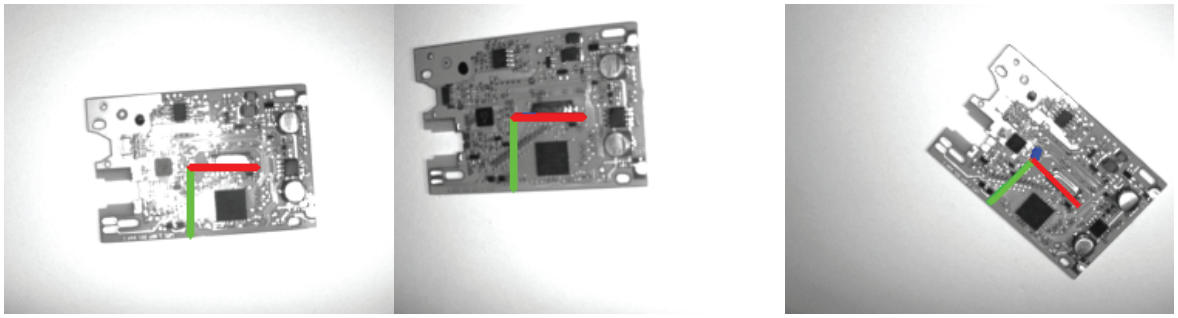

Abbildung 6.2: 3 von 324 Ansichten der Ground-Truth-Sequenz der Leiterplatte mit eingezeichnetem bekannten Objekt-Koordinatensystem. Die Achsen haben in allen Bildern dieser Arbeit die Länge 20 mm und sind farblich mit rot (X), grün (Y) und blau (Z) gekennzeichnet.

Die folgende Ausführung gliedert sich in zwei Unterabschnitte, die sich mit den Trainings- und Lokalisationsparametern beschäftigen.

6.1.1 Trainingsparameter

Die Parameter des Trainings beziehen sich in erster Linie auf die Einstellmöglichkeiten bei der Verarbeitung einer Modellansicht. Dies bezieht sich insbesondere auf den Detektor- und Deskriptorschritt sowie den Ebenenfit für jede Region. Weiterhin muss schon während des Trainings der Datenbankzugriff festgelegt werden, da u. U. die Merkmalsvektoren der Modellregionen in eine Indexstruktur einsortiert werden müssen. Die folgenden Paragraphen beschäftigen sich mit den entsprechenden Parametern.

Nicht behandelt wird in diesem Abschnitt das Einlernen mehrerer Modellansichten und auf wel-

che Weise man dies im Optimalfall tun sollte. Auf diesen Aspekt wird in Abschnitt 6.3.2 eingegangen.

Detektor Als Detektor wird in dieser Arbeit entweder eine der beiden (von den Ergebnissen und Parametern) identischen SIFT-Implementierungen oder die MSER-Implementierung verwendet. Sie sind in ihrer Klasse - ähnlich- bzw. affin-kovariant (vgl. Abschnitt 3.2.5.2) - jeweils die am besten bewerteten Detektoren und wurden aufgrund dessen für diese Arbeit ausgewählt.

Sowohl bei SIFT als auch MSER werden die in Abschnitt 3.2.3.3 bzw. 3.2.4.1 beschriebenen und in der Literatur üblichen Standardeinstellungen der Parameter verwendet. Die MSER-Implementierung verwendet allerdings einen leicht modifizierten Algorithmus zur Bestimmung der Q_{MSER} (vgl. Abschnitt 3.2.3.3). Anstatt das Minimum der relativen Größenänderung $q(i)_{\text{MSER}}$ mit $\Delta_{\text{MSER}} = 5$ zu bestimmen, werden alle Regionen unter einem Schwellwert $q(i)_{\text{MSER}} < \varepsilon_{\text{MSER}}$ herangezogen. Allerdings werden zu kleine Regionen mit einer Fläche $|Q| < a_{\text{min,MSER}} = 30\,\text{pix}$ als auch zu große Regionen mit $|Q| < a_{\text{max,MSER}} = 0.75 a_{\text{i}}$ bzgl. der Bildfläche a_{i} ignoriert. Weiterhin wird jeweils die instabilere Region unterdrückt, wenn zwei Regionen mehr als $a_{\text{div,MSER}} = 0.5|Q|$ der Pixel gemeinsam haben.

In beiden Fällen lässt sich über den Schwellwert $\varepsilon_{\text{SIFT}}$ bzw. $\varepsilon_{\text{MSER}}$ die Anzahl der detektierten Regionen steuern. Je nach Einstellung werden dabei weniger, aber robuste oder mehr, dafür weniger stabile Regionen gefunden. Es stellt sich natürlich für das Training die Frage, welche Einstellung für das System am besten ist. Dafür wurden anhand der zwei Objekte Leiterplatte und Box jeweils 10 Modelle mit unterschiedlichen Schwellwerteinstellungen beider Verfahren trainiert und jedes Modell mittels den 324 Suchansichten der Ground-Truth-Sequenz evaluiert. Die Einstellungen der Lokalisation spielen dabei keine Rolle, da nur der relative Vergleich entscheidend ist. Abb. 6.3 zeigt die Ergebnisse.

Beginnt man mit den restriktivsten Schwellwerteinstellungen $\varepsilon_{\text{SIFT}} = 30.0$ bzw. $\varepsilon_{\text{MSER}} = 0.05$ so wird die Erkennungsrate bis zu einer gewissen Grenze mit steigender Anzahl Regionen immer besser. Danach tritt ein Sättigungseffekt ein, bei der das System aus den zusätzlichen Regionen keine neuen Informationen gewinnen kann. Ähnlich verhält es sich mit den Lagefehlern, allerdings steigen diese sogar innerhalb der Sättigung manchmal wieder an. Dies lässt sich damit begründen, dass immer mehr instabile Regionen hinzukommen, sich daher auch mit einer höheren Wahrscheinlich Fehlkorrespondenzen innerhalb des gefundenen Clusters befinden und die globe Lageermittlung gestört wird. Weiterhin benötigt das Verfahren mit mehr Regionen auch eine längere Verarbeitungszeit.

Die optimalen Einstellungen liegen daher an der Sättigungsgrenze. Diese Grenze ist prinzipiell objektabhängig, allerdings ist der Einfluss der beiden untersuchten Objekten gering. Die Einstellungen werden daher objektunabhängig auf $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$ festgelegt und im weiteren Verlauf verwendet. Dies entspricht im übrigen auch den in der Literatur üblichen Werten.

Deskriptor Als Deskriptor wird während der gesamten Experimente das SIFT-Verfahren aus Abschnitt 3.3.1 verwendet. Alle Parameter entsprechen den in diesem Abschnitt angegebenen Referenzeinstellungen.

Ebenenfit Wie in Abschnitt 4.4.1 beschrieben, muss für jede Region eine Ebene an die entsprechende Szenenteilmenge angefittet werden. Dazu wird RANSAC verwendet, der als Parameter die Anzahl der Zyklen m_{RAN} sowie die Anzahl der gezogenen Punkte n_{RAN} benötigt. Da während des Trainings die Laufzeit eine untergeordnete Rolle spielt, wird die Anzahl der Zyklen auf $m_{\text{RAN}} = 1000$ eingestellt. Dies ermöglicht anstatt der 3 minimal benötigten Punkte für den Ebenenfit pro Zyklus, den höheren Wert $n_{\text{RAN}} = 5$ zu wählen und dennoch mit einer Wahrscheinlichkeit größer 99.9% bei 60% maximal angenommener Ausreißer eine korrekte Ebene zu erhalten (FB81).

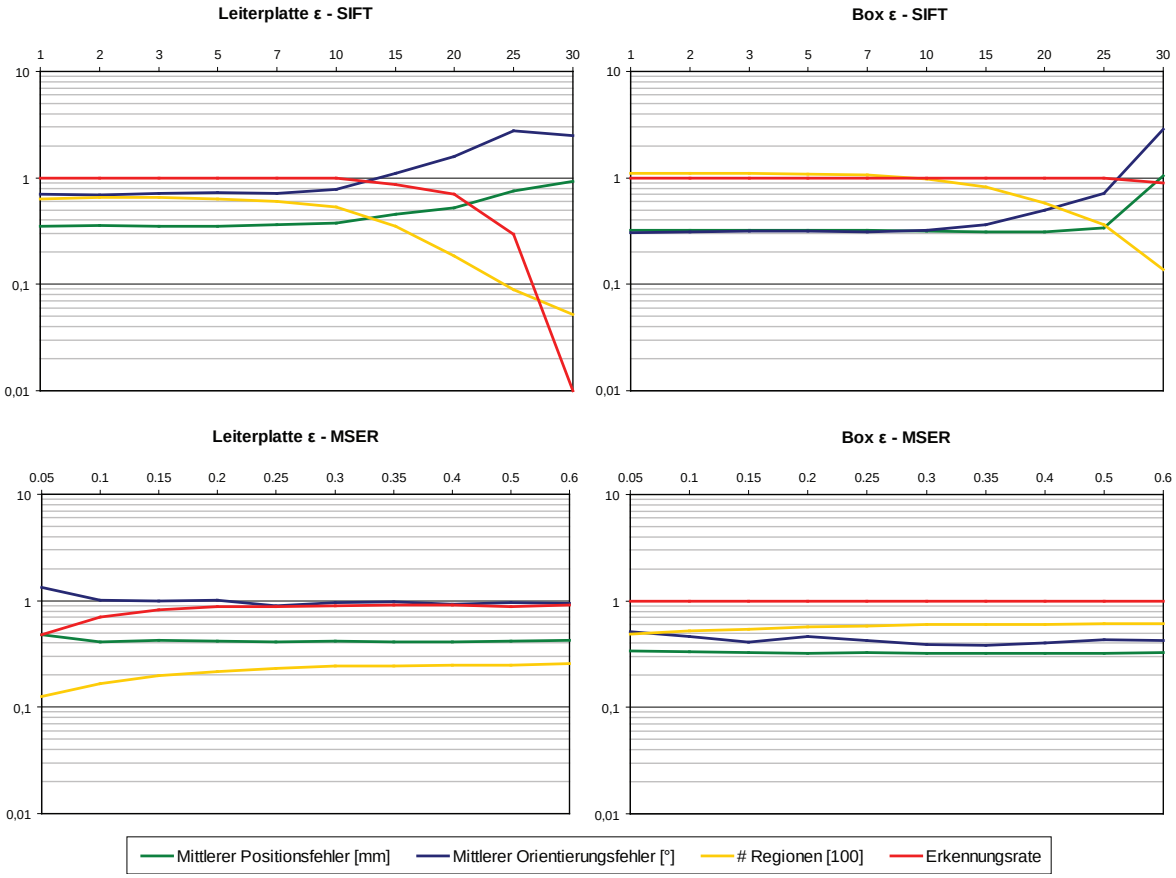

Abbildung 6.3: Einfluss unterschiedlicher Schwellwerte $\varepsilon_{\text{SIFT}} \in [1,30]$ bzw. $\varepsilon_{\text{MSER}} \in [0.05,0.6]$ der Detektoren während des Trainings eines Modells. Dargestellt sind jeweils der mittlere Translationsfehler, der mittlere Orientierungsfehler, die mittlere Anzahl geclusterter Regionen und die Erkennungsrate bei der Evaluation der Modelle mit den 324 Suchansichten der Ground-Truth-Sequenz. Werden zuwenig Regionen gefunden, bricht die Erkennungsrate des Systems ein (und die restlichen dargestellten Größen verlieren an Aussagekraft), werden zuviele Regionen gefunden, gerät das System in eine Sättigung und die Genauigkeit verbessert sich trotz steigendem Rechenbedarf nicht mehr. Die besten Einstellungen ergeben sich an der Sättigungsgrenze zu $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$.

Für die Bewertung wird weiterhin der Parameter ε benötigt, der angibt in welchem Abstand von der gefitteten Ebene ein Punkt noch der Ebene zugehörig zählt. Da die lokalen Regionen (bei der verwendeten Sensorik) teilweise nur eine Ausdehnung von $b \approx 1\,\text{mm}$ in der Szene besitzen, muss ε signifikant kleiner gewählt werden. Hier wird $\varepsilon = \frac{1}{100} b \approx 10\,\mu\text{m}$ verwendet.

Datenbankzugriff Die Bestimmung des Nächsten-Nachbarn (NN) jeder Suchregion im Merkmalsraum des Deskriptors über eine NN-Suche in der Modelldatenbank ist konzeptionell keine Herausforderung. Allerdings wird in Abhängigkeit der Anzahl der Suchregionen h_l und der Modellregionen n_k in der Datenbank ein Großteil der Rechenzeit der gesamten Verarbeitungskette für die NN-Suche verwendet. Es wurden daher im Rahmen dieser Arbeit zwei Diplomarbeiten betreut, die sich einerseits mit intelligenten, hierarchischen Datenstrukturen (Tra09) und hardwareoptimierten Bruteforce-Implementierungen (Zoe10) beschäftigt haben. An dieser Stelle soll nur die Quintessenz beider Arbeiten präsentiert werden, die Details können in (Tra09; Zoe10) nachgeschlagen werden.

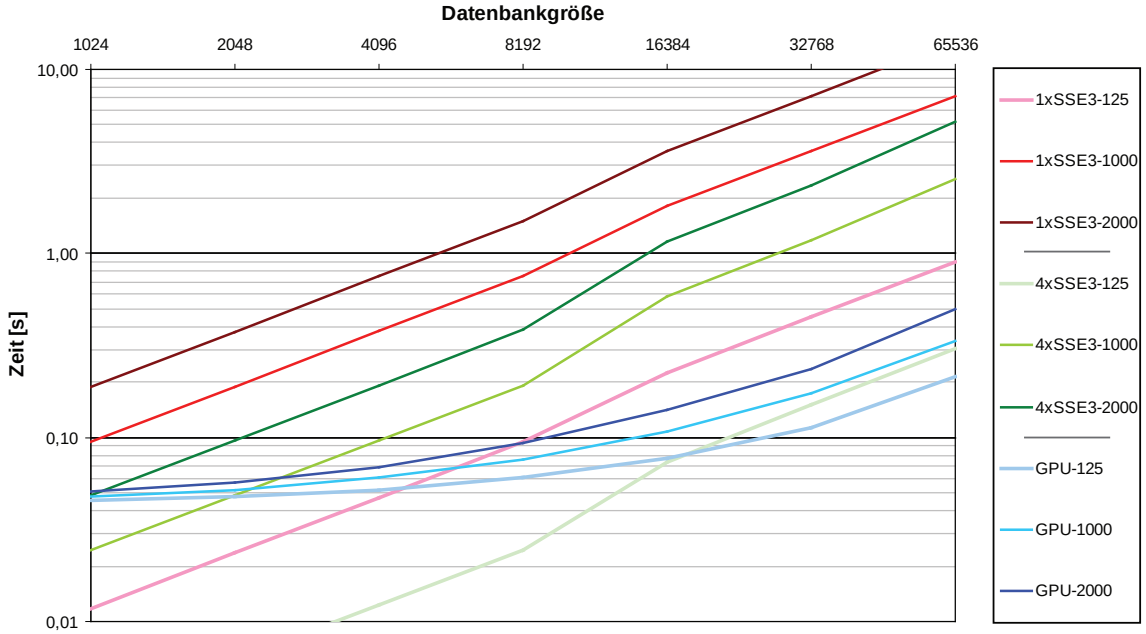

Abbildung 6.4: Performance-Vergleich zwischen drei verschiedenen Bruteforce-NN-Implementierungen des Matchers. Dargestellt ist eine assembleroptimierte SIMD-Implementierung mittels SSE3 (1xSSE3, rot), dessen parallelisierte Variante (4xSSE3, grün) und eine GPU Variante (GPU, blau). Die Zeiten beziehen sich auf die NN-Suche von $n_l = 125$ (hell), 1000 (normal), 2000 (dunkel) Suchregionen zu unterschiedlichen Datenbankgrößen n_k. Details können (Zoe10) entnommen werden, als Hardware wurde ein Intel Core 2 Quad Q9450 mit $4 \times 2.66\,\text{GHz}$ und eine NVIDIA GTX 260 verwendet.

Das Hauptproblem der NN-Suche ist der 128-dimensionale Deskriptorraum des SIFT-Deskriptors. Das führt dazu, dass der Raum nur sehr dünn von Merkmalsvektoren besetzt ist. Selbst die, für hochdimensionale Räume optimierten, hierarchischen Datenstrukturen wie der *X-Tree* (BKK96) werden daher im Vergleich zu den anderen betrachteten Verfahren sehr ineffizient (Tra09) und wurden daher nicht weiter verfolgt. Ebenso wurde die Möglichkeit verworfen, den Deskriptorraum mittels PCA auf weniger Dimensionen herunterzubrechen, da dann der Deskriptor seine Eindeutigkeit verliert (MS05).

Abhilfe schaffen approximative Techniken wie das *VA-File* (WSB98). Dieses benötigt eine Vorverarbeitungszeit zum Bestimmen von geeigneten Datenpunkt-Kandidaten, kann dann allerdings die NN-Suche schneller verarbeiten. Ab einer Datenbankgröße von $n_k = 50000$ zahlt sich in (Tra09) dieses Vorgehen im Vergleich zu einer (nicht hardware-optimierten) Bruteforce-Implementierung aus.

Die Bruteforce-Implementierung ist allerdings algorithmisch sehr einfach und lässt sich daher mittels Assembler, SIMD-Befehlssätzen wie SSE3 und Parallelisierung auf CPU-Ebene einfach, aber sehr stark beschleunigen. Mittels dieser SSE3-Implementierung konnte aufgrund von Speicherlimitierungen nicht mehr die Datenbankgröße ermittelt werden, bei der das VA-File effizienter wird. Sie liegt aber schätzungsweise bei ca. 10^6 Datenpunkten, einer DB Größe die innerhalb des Projektes nie benötigt wurde. Aufgrund dessen wurden die intelligenten Datenstrukturen nicht mehr weiter verfolgt.

In (Zoe10) wurde daher versucht, mittels dem massiv parallelem Einsatz von einfachen Recheneinheiten auf der GPU eine weitere Beschleunigung des Bruteforce-Ansatzes zu erzielen. Die Ergeb-

nisse für unterschiedliche n_l und n_k sind in Abbildung 6.4 zu sehen. Gut erkennbar ist die Komplexität von $O(n_l n_k)$ der SSE3-Bruteforce-Implementierung. Es wurde auch gleichzeitig der Speedup $s(4)$ der SSE3-Implementierung zwischen der sequentiellen Ausführung mit 1 Thread bzw. Kern und 4 Rechenkernen untersucht. Interessanterweise liegt dieser mit $s(4) > 4$ im superlinearen Bereich bei $n_k \leq 8192$, was sich durch eine Zunahme der insgesamten Cachegröße bei 4 Kernen erklären lässt. Ab $n_k > 8192$ scheint dieser Vorteil aufgrund der größeren DB aufgebraucht zu sein, gut zu erkennen an dem Knick des ansonsten geraden Verlaufs.

Auch die GPU-Variante besitzt theoretisch eine Komplexität von $O(n_l n_k)$, allerdings ist diese aufgrund des Auslastungsgrads der einzelnen GPU-Rechenkerne in dem betrachteten Bereich nicht mehr ersichtlich. Gut zu erkennen ist dagegen die Initialisierungszeit von ca. 50 ms. Allerdings zeigt sich, dass sich der Aufwand selbst bei kleinem $n_l = 125$ schon ab einer Datenbankgröße von $n_k = 16384$ lohnt bzw. bei ca. $n_{lk} = n_l n_k = 125 \cdot 16384 \approx 2 \cdot 10^6$ zu vergleichenden Merkmalsvektoren.

Welche Implementierung verwendet wird, hängt daher von der Datenbankgröße und Suchanfrage ab. Unter $n_{lk} \approx 2 \cdot 10^6$ zu vergleichenden Merkmalsvektoren empfiehlt sich die SSE3-Variante, ansonsten ist die GPU-Implementierung vorzuziehen. Nur für extrem große Modelldatenbanken sind die intelligenten Datenstrukturen geeignet, dann muss allerdings auch mit einer Zeit der NN-Suche im Minutenbereich gerechnet werden. Würde man sie allerdings auf gleiche Weise optimieren und parallelisieren wären sie wieder sehr attraktiv. Allerdings ist der Aufwand aufgrund der algorithmischen Komplexität extrem hoch. In dieser Arbeit werden je nach DB-Größe beide Bruteforce-Implementierungen eingesetzt.

6.1.2 Lokalisationsparameter

Die Parametereinstellungen der Lokalisation müssen hinsichtlich zweier, meist gegensätzlicher Kriterien bewerten werden. Am wichtigsten ist eine hohe Genauigkeit und Robustheit des Systems, allerdings ist ebenfalls eine kurze Laufzeit von Bedeutung. Die Parametereinstellungen sind daher oftmals ein Kompromiss zwischen beiden Optimierungszielen.

Zuerst werden die Parameter des Gruppierungsschritts aus Abschnitt 4.3.3 untersucht, da dort auch die Kovarianzmatrix des Ähnlichkeitsmaßes geschätzt wird. Dieses wird u. a. benötigt um im nächsten Abschnitt die Parameter der Korrespondenzfindung festzulegen. Dabei handelt es sich um die ersten drei Schritte aus Abb. 4.3. Abschließend werden dann die Parametereinstellungen der globalen Lagebestimmung aus Abschnitt 4.3.4 behandelt. Die lokale Lagebestimmung besitzt keine Parameter, daher wird sie hier auch nicht behandelt.

6.1.2.1 Parameter des Gruppierungsschritts

Der Gruppierungsschritt aus Abschnitt 4.3.3 dient dazu, Cluster in einer Menge von lokalen Lagen ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ zu finden. Dazu muss u. a. die Ähnlichkeit zwischen einer lokalen Lage ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O}}$ und einem Clusterzentrum ${}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{O},c}$ mittels des in Gl. 4.15 angegebenen Distanzmaßes $d_{\mathbf{P}}(\cdot)$ bestimmt werden. Es setzt sich aus der Translationsdifferenz $d_t(\cdot)$ und der Orientierungsdifferenz $d_\alpha(\cdot)$ zusammen und verknüpft beide Größen über die Mahalanobisdistanz miteinander. Als Gewichtung wird dabei die Kovarianzmatrix $\mathbf{P}$ genutzt, die im Folgenden geschätzt werden soll.

Über die Ground-Truth-Daten kann für jede Suchansicht das exakte Clusterzentrum angegeben werden und damit für jede lokale Lage i die exakten Differenzen für $d_t(\cdot)$ und $d_\alpha(\cdot)$, zusammengefasst in dem 2D-Distanz-Vektor $\mathbf{d}_i(d_t(\cdot), d_\alpha(\cdot))^{\mathrm{T}}$ angegeben werden. Werden n lokale Lagen mit diesem Mechanismus ausgewertet, kann man darüber Statistik treiben und die Kovarianzmatrix mittels $\mathbf{P} = (n-2)^{-1} \sum_{i=1}^{n} \mathbf{d}_i \mathbf{d}_i^{\mathrm{T}}$ angeben.

Die Kovarianzmatrix $\mathbf{P}$ soll die inhärenten Eigenschaften der lokalen Lagerekonstruktion ohne objektspezifische bzw. ansichtsspezifische Einflüsse widerspiegeln. Es werden daher lokale Lagen

	SIFT	SIFT & 3D	MSER	MSER & 3D
P				
p_{11}	0.00014	0.00002	0.00052	0.00043
p_{12}, p_{21}	0.00229	0.00088	0.01545	0.02049
p_{22}	0.04123	0.04716	0.50170	1.00372
$\mathbf{Q} = \mathbf{P}^{-1}$				
q_{11}	81810	405926	25652	104601
q_{12}, q_{21}	-4556	-7550	-789	-2135
q_{22}	278	161	26	45
abgeleitete Größen				
$\tau_{\mathbf{P}}$	0.96	0.94	0.96	0.98
$r_{\mathbf{P}}\left[\frac{^\circ}{\mathrm{mm}}\right]$	0.98	2.87	1.79	2.76

Tabelle 6.1: Geschätzte Kovarianzmatrizen der Differenzen der lokalen Lagen zum Ground-Truth Clusterzentrum für die Detektoren SIFT und MSER sowie unterschiedlicher Tiefenrekonstruktionen über Skalierung oder 3D-Stereo-Tiefenrekonstruktion. Ebenfalls ist der Korrelationskoeffizient $\tau_{\mathbf{P}}$ sowie zur besseren Interpretation das Gewichtungs-Verhältnis $r_{\mathbf{P}}$ der Translations- und Orientierungsdifferenz (bei Vernachlässigung der Korrelation) der Mahalanobisdistanz angegeben. Man beachte dass $r_{\mathbf{C}}$ in mm und $^\circ$ angegeben ist, die Kovarianzmatrix allerdings in projektüblichen SI-Einheiten m und rad.

von allen Objekten und allen 324 Suchansichten mit Ground-Truth-Informationen ausgewertet, was zu $n \approx 10^6$ verwendeten Distanz-Vektoren führt. Allerdings soll $\mathbf{P}$ das Rauschen der lokalen Lagen um ein Clusterzentrum beschreiben und nicht durch lokale Lagen von Fehlkorrespondenzen beeinflusst werden. Es werden daher nur die 20% besten, d. h. am nächsten zum Clusterzentrum befindlichen Lagen für die Kovarianzschätzung herangezogen. Da hierfür allerdings schon die Kenntnis von $\mathbf{P}$ für das Distanzmaß $d_{\mathbf{P}}(\cdot)$ benötigt wird, muss folgender iterativer Ansatz ausgewählt werden:

Schritt 0 Initialisiere $\mathbf{P}^{[0]} = \mathbf{I}_{2\times 2}$

Schritt 1 Ermittle mittels $d_{\mathbf{P}^{[k]}}(\cdot)$ die 20% am nächsten zum Ground-Truth-Clusterzentrum liegenden lokalen Lagen. Bestimmte damit $\mathbf{P}^{[k+1]}$.

Schritt 2 Falls $(\mathbf{P}^{[k]})^{-1}\mathbf{P}^{[k+1]} \approx \mathbf{I}_{2\times 2}$ breche ab und gib $\mathbf{P}^{[k+1]}$ zurück, ansonsten erhöhe $k \leftarrow k+1$ und springe zu Schritt 1.

Das Verfahren wird separat für den SIFT-Detektor und den MSER-Detektor angewendet, da deren unterschiedliche Konstruktionsmethoden die Eigenschaften der lokalen Lagerekonstruktion am meisten beeinflussen. Alternativ wird anstatt der Tiefenbestimmung ${}^{\mathrm{B}}p_z = {}^{\mathrm{A}}p_z s^{-1}$ aus Abschnitt 4.3.1.2 mittels der Skalierung s die (robustere) Tiefe aus einem triangulationsbasiertem dichten 3D-Tiefenbild auf Stereobasis verwendet. Die unterschiedlichen Ergebnisse finden sich in Tab. 6.1.

In allen Fällen ist der Korrelationskoeffizient $\tau_{\mathbf{P}} = \frac{p_{12}}{\sqrt{p_{11}}\sqrt{p_{22}}} \approx 1$, d. h. die Translations- und Orientierungsdifferenz stehen in einem positiven linearen Zusammenhang. Dies ist auch nicht weiter verwunderlich, da Fehler in der kovarianten Regionendetektion sich auf alle Parameter und daher sowohl auf die rekonstruierte Orientierung als auch Translation auswirken.

Man beachte weiterhin das Verhältnis $r_{\mathbf{P}} = \frac{\sqrt{q_{11}}}{\sqrt{q_{22}}}$ zwischen den Gewichtungsfaktoren der Translations- und Orientierungsdifferenz der Mahalanobisdistanz (ohne Berücksichtigung der Korrelation). Es liegt für SIFT bei ca. $1\frac{^\circ}{\mathrm{mm}}$, d. h. 1° Orientierungsabweichung hat den gleichen Einfluss auf das Distanzmaß $d_{\mathbf{P}}(\cdot)$ wie 1 mm Translationsabweichung. Für MSER verdoppelt sich dieser Wert fast, was darauf deutet, dass die Orientierungen der lokalen Lagen bei MSER einem stärkeren Rauschen

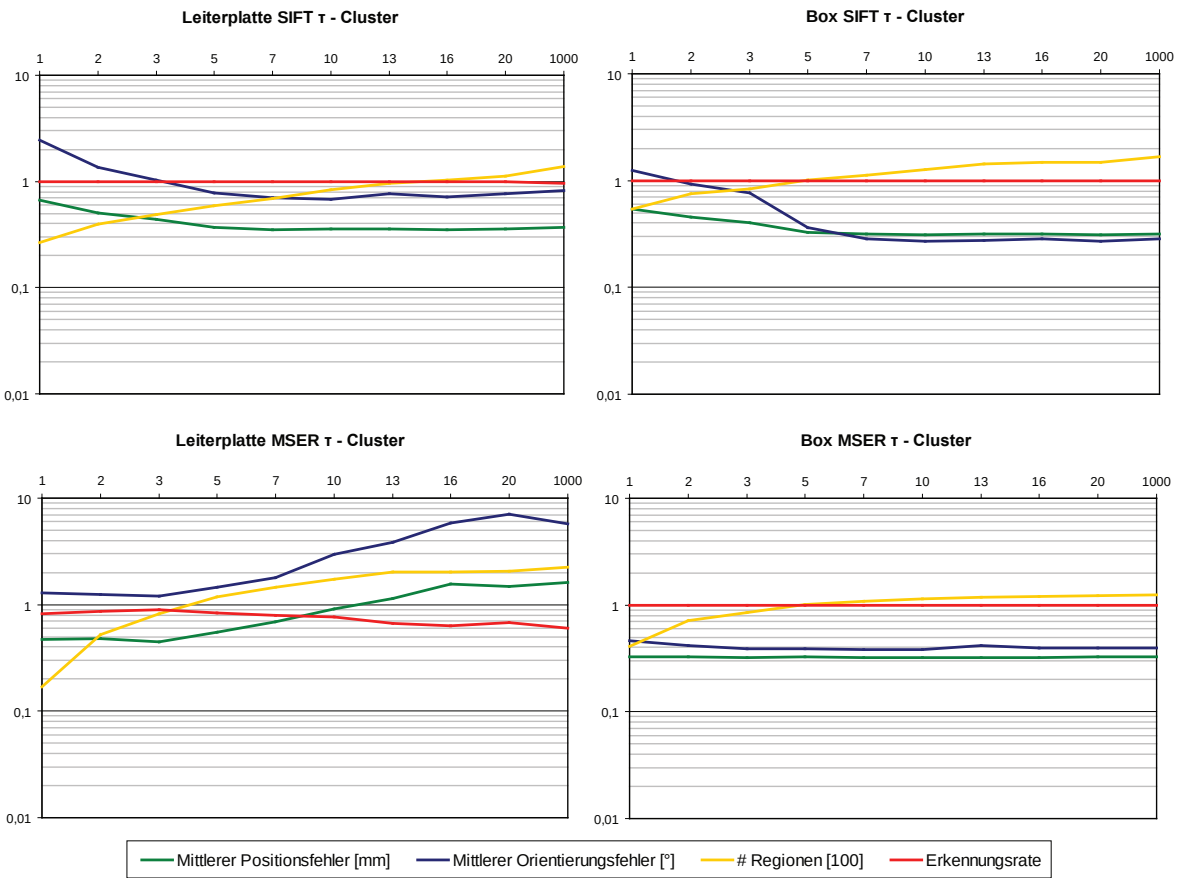

Abbildung 6.5: Verhalten des Gesamtsystems mit 3D-3D-Lagebestimmung inkl. Stereo auf verschiedene Clustergrößen, angegeben durch den Parameter τ in, durch **P** definierte ellipsenförmige 2D-Standardabweichungen um das Clusterzentrum.

unterliegen. Dies ist umso erstaunlicher, da bei den ähnlich-kovarianten SIFT-Korrespondenzen keine Verkippungen geschätzt werden können und ein durchschnittlicher Fehler *der lokalen Lagen* von über 7.5°[1] zu erwarten ist. Für affin-kovariante MSER-Korrespondenzen kann daher ein fast doppelt so großes Orientierungsrauschen der rekonstruierten lokalen Lagen angenommen werden. Die Tiefenwerte durch 3D-Informationen zu verbessern, reduziert die Translationsabweichungen im Schnitt um ca. die Hälfte, da diese im Vergleich zu den gleichbleibenden Orientierungsabweichungen stärker gewichtet werden. Dies ist eine experimentelle Bestätigung, dass die lokale Skalierungsänderung der korrespondierenden Regionen für eine stabile Tiefenrekonstruktion ungünstig ist.

Die Kovarianzmatrix beschreibt die lokale Struktur des Rauschens um das Clusterzentrum. Es sagt allerdings nichts über die tatsächliche Stärke des Rauschens aus, da **P** nur aus den besten 20% der Daten gewonnen wurde. In einem zweiten Schritt wird daher untersucht, wie das gesamte System auf unterschiedliche Clustergrößen τ aus Gl. 4.16 reagiert. τ gibt dabei die Distanz in, durch **P** definierte ellipsenförmige 2D-Standardabweichungen um das Clusterzentrum an, in denen sich die lokalen Lagen befinden dürfen um noch dem Cluster zugeordnet zu werden (vgl. Abschnitt 4.3.3.2). Abb. 6.5 zeigt die Ergebnisse für die globale Lagebestimmung mittels 3D-3D-Korrespondenzen auf der Leiterplatte und der Box (jeweils mit 3D-Tiefenstabilisierung), Abb. 6.6 die Ergebnisse auf der

[1] da die Suchansichten mit Ground-Truth-Informationen Verkippung bis 15° aufweisen. Zusätzlich kommt noch der Orientierungsfehler in der Kameraebene hinzu. Dieser ist allerdings im Schnitt um eine Größenordnung kleiner.

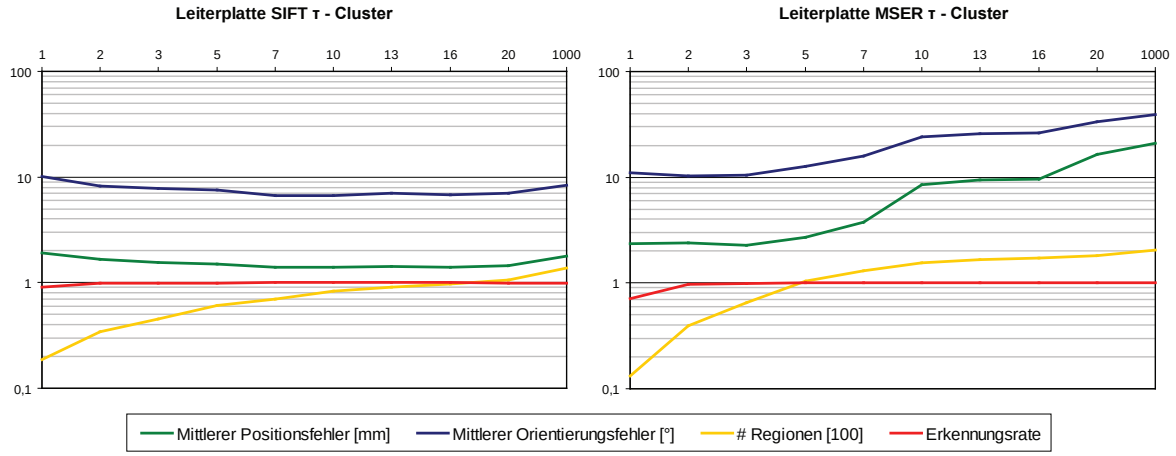

Abbildung 6.6: Der gleiche Versuch wie in Abb. 6.5, diesmal allerdings bei Verwendung der 2D-3D-Lagebestimmungen. Die besten Werte für τ, dies ist für SIFT ca. 7 und für MSER ca. 3 sind für beide globalen Lageauswertungen dieselben und werden daher objekt- und verfahrensunabhängig verwendet.

Leiterplatte für die globalen Lagebestimmung mittels 2D-3D-Korrespondenzen.

Je größer τ, desto größer die Anzahl der lokalen Lagen im Cluster. Dies ist für die globale Lagebestimmung allerdings nicht nur von Vorteil, da ab einer bestimmten Größe die Wahrscheinlichkeit steigt, auch Fehlkorrespondenzen mit zu gruppieren. Ebenfalls steigt die Gefahr, nahe beieinander liegende Objekte nicht mehr trennen zu können, wie z. B. in Abb. 6.33 in der untersten Szene ersichtlich. Eine obere Grenze für τ sollte daher die halbe Distanz zwischen den Clusterzentren zweier dicht aneinander liegender Objekte sein, falls der Ursprung des Objekt-Koordinatensystems im Zentrum des Objektes eingelernt wurde. Dies ist bei gleicher Orientierung der Objekte ca. $\tau < \frac{l}{\sqrt{p_{11}}}$, wobei l der kleinste Abstand des Ursprungs zu einer Außenkante des Objektes ist und $\sqrt{p_{11}}$ die ermittelte Standardabweichung der Translation. Für unsere Objekte ist $l \approx 20\,\text{mm}$, was z. B. bei SIFT mit 3D-Informationen zu $\tau < 5$ führt. Werden allerdings zu wenige lokale Lagen geclustert, wird die globale Lagebestimmung ebenfalls unrobust. Die Genauigkeit verläuft daher in Abhängigkeit von τ je nach Objekt und Detektor in einer mehr oder weniger ausgeprägten Badewannenkurve.

Interessanter weise liegen die besten Werte für τ, dies ist für SIFT ca. 7 und für MSER ca. 3 über der oberen Grenze. Es muss daher ein Kompromiss getroffen werden, zwischen Trennbarkeit bei mehreren Objekten und genauer Lagerekonstruktion bei einem Einzelobjekt. Hier werden für die weitere Evaluation obig genannte Parametereinstellungen verwendet.

6.1.2.2 Parameter der Korrespondenzfindung

Mit der Korrespondenzfindung beschäftigen sich die ersten drei Schritten der Ablaufkette aus Abb. 4.3. Dies sind der Detektor kovarianter Regionen $\mathcal{R}$, der Deskriptor zum Beschreiben der invarianten Merkmalsvektoren $\mathbf{o} = \mathbf{des}(\mathcal{R})$ und der Matcher zum Auffinden der Regionenkorrespondenzen zwischen Suchansicht und Modellansichten über einer NN-Suche im Merkmalsraum mittels des Distanzmaßes $d_{\mathbf{des}}(\cdot)$. Die Parameter des Detektors und Deskriptors werden aus dem Training übernommen (vgl. Abschnitt 6.1.1), da das Ziel eine möglichst identische Regionenfindung und -beschreibung ist.

Einzig ein unterschiedlicher Schwellwert ε zur Festlegung der Stabilität bzw. Anzahl der gefundenen Regionen wird untersucht. Zur Bestimmung der Einstellung dieses Parameters zum Training wurde das System mit unterschiedlichen Werten trainiert und mit *denselben* Werten die Ground-

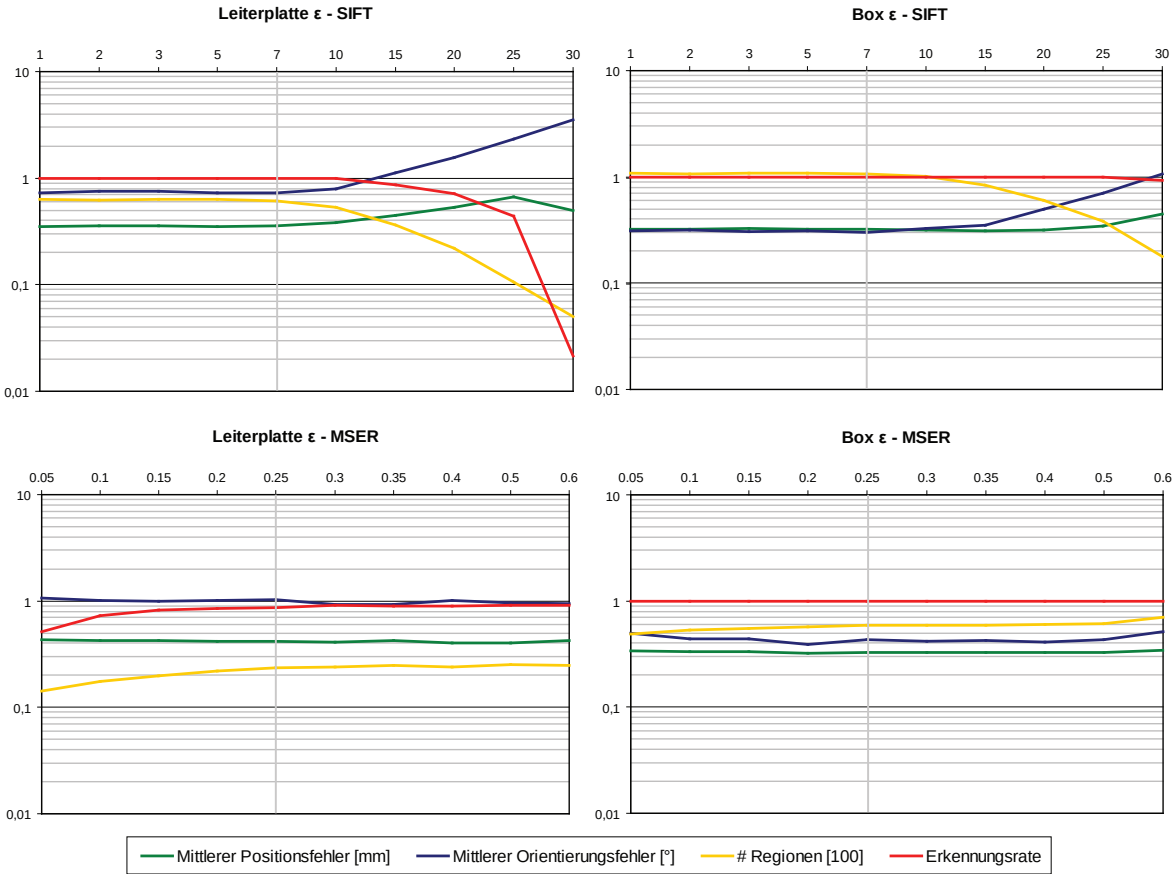

Abbildung 6.7: Einfluss unterschiedlicher Schwellwerte $\varepsilon_{\text{SIFT}} \in [1,30]$ bzw. $\varepsilon_{\text{MSER}} \in [0.05,0.6]$ der Detektoren während der online Lokalisation, wenn die grau markierten Einstellungen im Training verwendet wurden. Die besten Einstellungen ergeben sich wie beim Training (vgl. Abb. 6.3) an der Sättigungsgrenze zu $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$.

Truth-Sequenz evaluiert. Als beste Einstellung wurde dafür $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$ ermittelt. Es stellt sich allerdings die Frage ob es günstiger ist, während der online Lokalisation über einen kleineren (SIFT) bzw. größeren (MSER) Schwellwert mehr und damit instabilere Regionen zu finden als während des Trainings, den Schwellwert gleich zu belassen oder weniger Regionen zu detektieren. Das Verhältnis $r_{gb} = \frac{g}{b}$ der Anzahl g korrekter Regionen zu der Anzahl b Fehlkorrespondenzen nimmt dabei bzgl. obig beschriebener Reihenfolge bei der Wahl von ε zu.

Abb. 6.7 zeigt die Ergebnisse bei der Wahl von ε und einer Einstellung während des Trainings von $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$. Im Vergleich zu den Experimenten während der Bestimmung des Trainingsparameters in Abb. 6.3 zeigen sich nur geringfügige Änderungen. Es ist daher nicht der Unterschied zu den Trainingseinstellungen entscheidend, sondern die Einstellung des Parameters an sich. Dies ist für SIFT sehr deutlich an der Sättigungsgrenze, da man so restriktiv wie möglich sein möchte um zu viele Regionen auf dem Hintergrund zu vermeiden, aber gleichzeitig so viele Regionen wie nötig detektieren muss um eine robuste Detektion zu gewährleisten. Gleiches gilt auch für den MSER Parameter, allerdings fällt dort die Entscheidung nicht so eindeutig aus. Die besten Einstellungen sind sowohl für das Training als auch die Detektion $\varepsilon_{\text{SIFT}} = 7.0$ bzw. $\varepsilon_{\text{MSER}} = 0.25$.

Die Parametereinstellungen des Detektors und Deskriptors sind damit festgelegt. Es verbleiben die Einstellungen der NN-Suche des Matchers. Dort gibt es nur einen relevanten Parameter $\tau_{\mathbf{des}}$, die

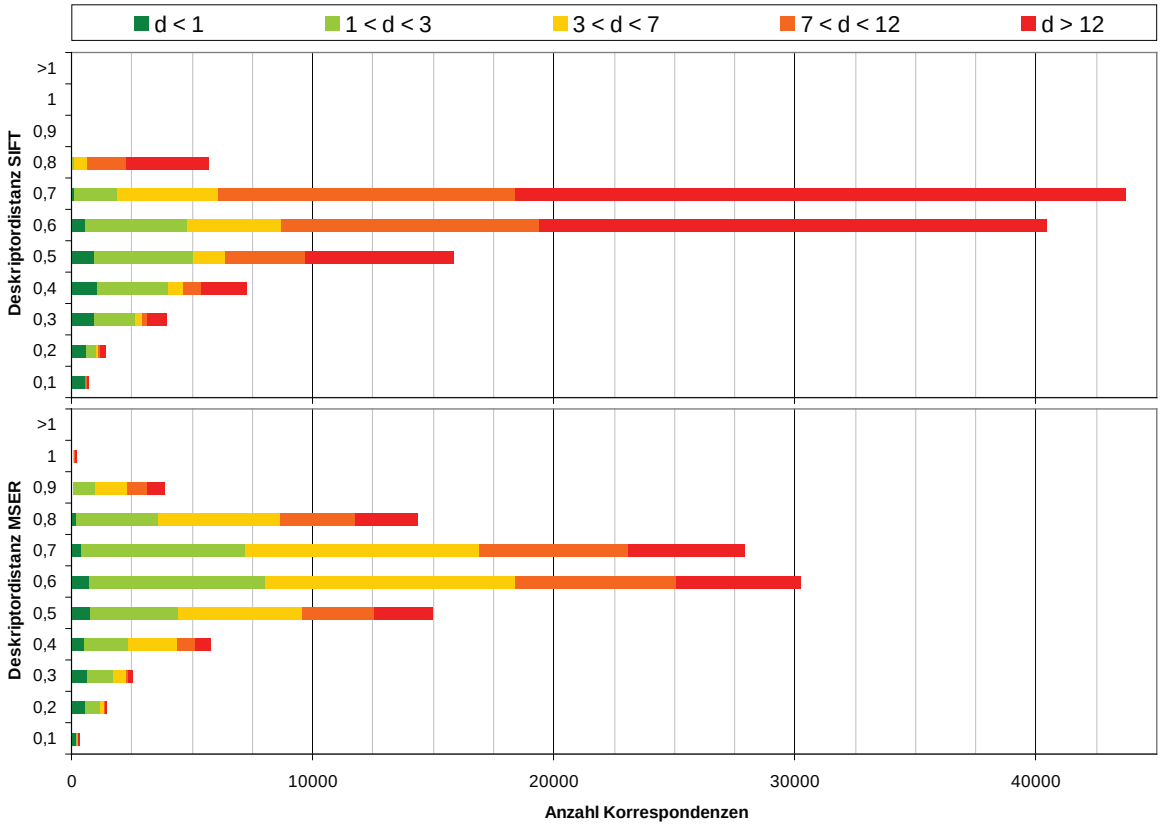

Abbildung 6.8: Histogramm über die Merkmalsabweichungen bzw. Deskriptordistanzen der gefundenen Korrespondenzen während der NN-Suche. Es sind jeweils die oberen Grenzen der Histogrammzellen auf der linken Achse angegeben. Weiterhin wird farblich zwischen der Güte der Korrespondenzen unterschieden. Die Güte ist dabei die Abweichung der abgeleiteten lokalen Lagen bzgl. der wahren Ground-Truth Lage, gemessen mit dem Distanzmaß $d = d_{\mathbf{P}}(\cdot)$ aus Gl. 4.15.

maximale Abweichung $d_{\mathbf{des}}(\mathbf{o}_1, \mathbf{o}_2)$ zweier Merkmalsvektoren $\mathbf{o}_1$ und $\mathbf{o}_2$ die für eine Korrespondenz akzeptiert wird. Je größer der Unterschied zweier korrespondierender Regionen, desto größer ist auch die Abweichung ihrer Merkmalsvektoren. Die Frage ist, ob auch der Umkehrschluss zutrifft, d. h. ob bei einer größeren Merkmalsabweichung auch die Güte der Regionenkorrespondenz schlechter wird.

Der Merkmalsvektor des für alle Regionen-Detektoren benutzte SIFT-Deskriptors ist auf 1 normiert. Daher gilt theoretisch $d_{\mathbf{des}}(\cdot) \in [0, \sqrt{2}]$, realistischer ist allerdings aufgrund der Verteilung der einzelnen Einträge $d_{\mathbf{des}}(\cdot) < 1$. Abb. 6.8 zeigt ein Histogramm über die Deskriptorabweichungen für die Leiterplatte bei der Evaluation der Ground-Truth-Sequenz. Weiterhin wird innerhalb des Histogramms farblich zwischen verschiedenen Güten der zugehörigen Regionenkorrespondenzen unterschieden. Die Güte ist dabei die Abweichung der daraus gewonnen lokalen Lagen bzgl. der wahren Ground-Truth Lage, gemessen mit dem Distanzmaß $d_{\mathbf{P}}(\cdot)$ aus Gl. 4.15.

Das Histogramm zeigt sowohl für SIFT als auch MSER eine Verteilung ähnlich einer Gaußglocke, mit einem Maximum zwischen 0.6 und 0.7. Während des Clusters werden für SIFT alle Korrespondenzen mit der Güte $d_{\mathbf{P}}(\cdot) < \tau = 7$ (alle grünen und gelben) bzw. für MSER mit $d_{\mathbf{P}}(\cdot) < \tau = 3$ (alle grünen) gruppiert (bei Vernachlässigung der Abweichung von gefundenem und wahren Clusterzentrum) und in der globalen Lageermittlung verwendet. Die restlichen Korrespondenzen (alle roten)

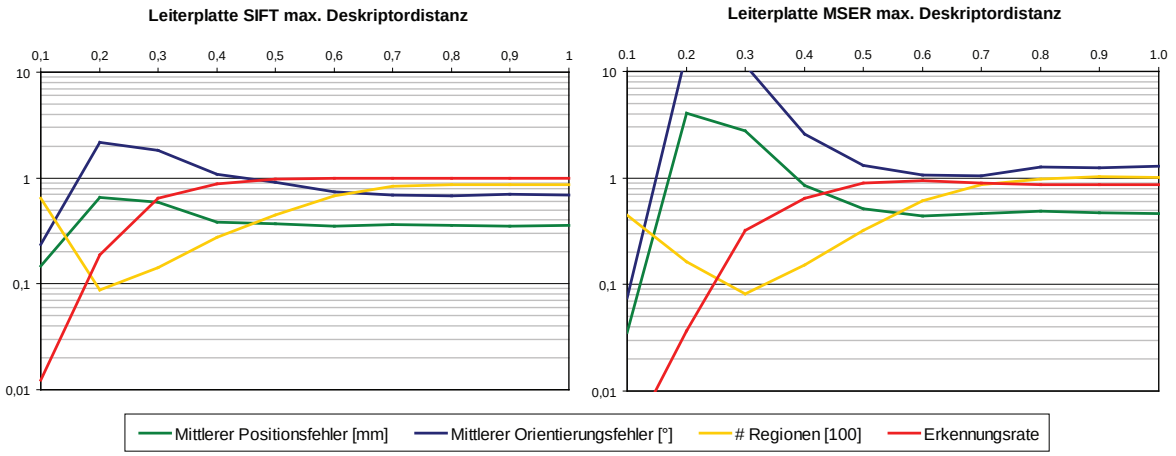

Abbildung 6.9: Einfluss der maximal erlaubten Merkmalsabweichung $\tau_{\mathbf{des}}$ der NN-Suche für SIFT und MSER auf das Gesamtsystem am Beispiel der Leiterplatte.

sind unerwünschte Fehlkorrespondenzen.

Man kann leicht erkennen, dass es keinen sinnvollen Zusammenhang zwischen Merkmalsabweichung und Korrespondenzgüte gibt. Zwar nimmt der Anteil der Fehlkorrespondenzen mit steigender Abweichung überproportional zu, allerdings befindet sich auch der Großteil der verwendbaren Korrespondenzen in dem Bereich $d_{\mathbf{des}}(\cdot) \in [0.4, 0.7]$. Es wurde daher untersucht, ob es sinnvoll ist, den Schwellwert der maximal akzeptierten Merkmalsabweichungen $\tau_{\mathbf{des}}$ möglichst klein zu wählen, mit der Gefahr, dass das System zu wenig Korrespondenzen für das Clustering und die robuste Lagebestimmung hat. Oder ob es besser ist, $\tau_{\mathbf{des}}$ groß zu wählen und über die vielen Fehlkorrespondenzen eine größere Ungenauigkeit und langsamere Verarbeitungsgeschwindigkeit zu riskieren.

Abb. 6.9 zeigt die Reaktion des Systems mit variierendem Schwellwert $\tau_{\mathbf{des}}$. Es ist leicht zu erkennen, das die Detektionsrate bei kleineren Werten völlig einbricht. Ebenfalls ist gut ersichtlich, dass die Genauigkeit mit zunehmendem Schwellwert wieder leicht schlechter wird und die besten Ergebnisse bei $\tau_{\mathbf{des}} = 0.7$ erzielt werden. Die Genauigkeiten bei $\tau_{\mathbf{des}} = 0.1$ sind aufgrund der extrem schlechten Detektionsrate zu vernachlässigen und beziehen sich auf die wenigen Suchbilder, die sehr Nahe an der Modellansicht liegen und daher fast optimale Korrespondenzen zulassen.

Schlussfolgernd kann man daher sagen, dass das System gut mit Fehlkorrespondenzen umgehen kann, da durch $\tau_{\mathbf{des}} = 0.7$ fast alle Korrespondenzen genutzt werden. Interessant ist dabei das Verhältnis $r_{gb} = \frac{g}{b}$ von korrekten Korrespondenzen zu Fehlkorrespondenzen. Es beträgt ca. 0.35 bei SIFT (grün und gelb zu rot) und 0.41 bei MSER (grün zu gelb und rot). Dennoch schneiden die SIFT-Detektoren bei der Genauigkeit besser ab. Vergleicht man allerdings das Verhältnis der sehr genauen Korrespondenzen zu den restlichen (dunkelgrün zu restlichen) erhält man fast identische 0.042 bzw. 0.043. Und dies obwohl die Kovarianzmatrix von SIFT in $d_{\mathbf{P}}(\cdot)$ wesentlich kleiner ist, d. h. SIFT exaktere Korrespondenzen finden muss und offensichtlich auch tut. Dies als ein statistisches Indiz für das bessere Abschneiden des SIFT-Detektors im Vergleich zu dem MSER-Detektor.

6.1.2.3 Parameter der globalen Lagebestimmung

Der letzte Schritt der Verarbeitungskette aus Abb. 4.3 ermittelt aus den geclusterten lokalen Lagen eine gemeinsame, robuste globale Lage (siehe Abschnitt 4.3.4). Dies sind die Durchschnitts- und Medianbildung sowie die Lageermittlung aus 3D-3D- und 2D-3D-Punktkorrespondenzen. Die ersten zwei Verfahren besitzen keinerlei relevante Parameter und werden daher hier nicht weiter behan-

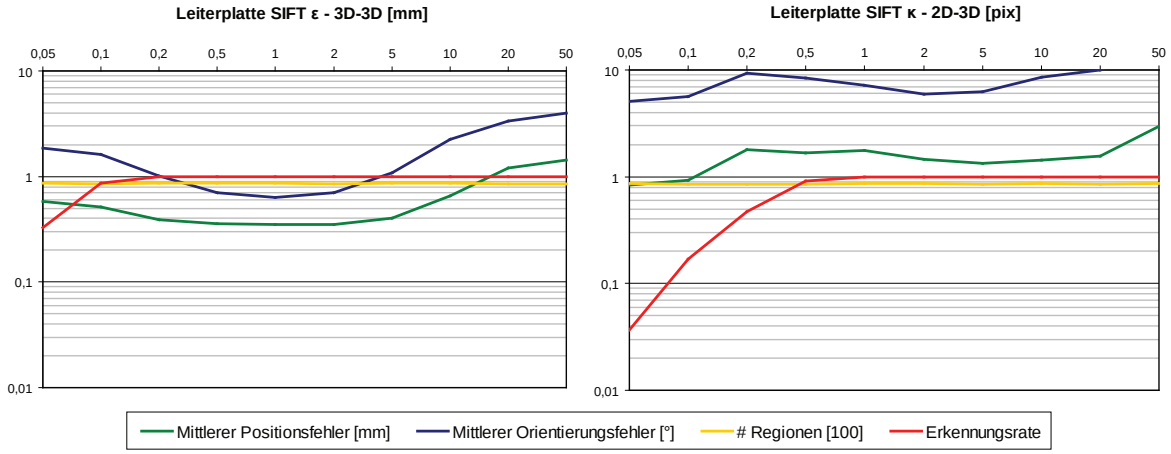

Abbildung 6.10: Verhalten des Systems bei unterschiedlichen maximalen Abweichungen ε (3D-3D-Punktkorrespondenzen) bzw. κ (2D-3D- Punktkorrespondenzen) in der Bewertungsfunktion $q_a(\cdot)$ der globalen Lagebestimmung. Ist die max. Abweichung zu niedrig gewählt, fließen zu wenige Punktkorrespondenzen in die Bewertung ein, ist die max. Abweichung zu groß gewählt, werden zuviele Fehlkorrespondenzen berücksichtigt. In beiden Fällen sinkt die Aussagekraft und der RANSAC liefert nicht unbedingt die beste Lage. Damit steigt die Ungenauigkeit und es bildet sich die ersichtliche Badewannenkurve. Der linke Knick bei den 2D-3D-Verfahren ist auf die niedrige Erkennungsrate zurückzuführen. Die besten Einstellungen sind bei $\varepsilon = 1\,\text{mm}$ bzw. $\kappa = 3\,\text{pix}$.

delt. Letztere Verfahren haben den Parameter ε bzw. κ der Bewertungsfunktion, die Anzahl n_{RAN} verwendeter Korrespondenzen pro RANSAC-Zyklus sowie die Anzahl m_{RAN} der RANSAC-Zyklen. Da hier nur die inhärenten Eigenschaften der unterschiedlichen globalen Lagebestimmungen untersucht werden sollen, wird stellvertretend für die 2D-3D-Punktkorrespondenzen-Verfahren der POSIT-Algorithmus untersucht.

Der Parameter ε aus Gl. 4.21 in $[\text{mm}]$ für die 3D-3D-Verfahren bzw. κ in $[\text{pix}]$ für die 2D-3D-Verfahren gibt die maximal erlaubte Abweichung zwischen den beobachteten Punkten und mittels der zu bewertenden Lage ins gleiche Koordinatensystem transformierten Modellpunkten an, damit diese noch in der Bewertung berücksichtigt werden. Ist die maximale Abweichung zu klein, fließen zu wenige Punktkorrespondenzen in die Bewertung mit ein und das Qualitätsmaß $q_a(\cdot)$ wird nicht aussagekräftig genug. Im schlimmsten Fall wird aufgrund eines zu kleinen $q_a(\cdot)$ die Lage sogar verworfen. Ist die maximale Abweichung allerdings zu groß gewählt, werden auch immer mehr weit abweichende Fehlkorrespondenzen berücksichtigt und $q_a(\cdot)$ spiegelt nicht unbedingt die wahre Güte der zu bewertenden Lage wieder. Dies führt dazu, dass von dem RANSAC-Verfahren nicht unbedingt die beste Lage zurückgegeben wird und die Genauigkeit sinkt.

Abb. 6.10 zeigt am Beispiel der Leiterplatte diesen Sachverhalt. Deutlich ist eine Badewannenkurve der Genauigkeiten mit den besten Einstellungen bei $\varepsilon = 1\,\text{mm}$ bzw. $\kappa = 3\,\text{pix}$ zu erkennen. Diese Einstellungen sind allerdings von der verwendeten Sensorik abhängig. Die Auflösung der Kameras hat einen Einfluss auf κ und indirekt durch die Rekonstruktionsgenauigkeit der Tiefe über die Stereotriangulation auch auf ε. Der größere Einfluss auf ε wird allerdings die Basisbreite des Stereosystems (siehe auch Anhang C.2) haben. Da keine alternative Sensorik im Projekt zur Verfügung stand, wurde dieser Einfluss nicht weiter untersucht.

Ein weiterer Parameter des RANSAC sind die Anzahl n_{RAN} verwendeter Korrespondenzen zur Ermittlung einer globalen Lage pro Zyklus. Bei 3D-3D-Punktkorrespondenzen sind dafür mindestens 3 nötig, bei 2D-3D-Punktkorrespondenzen im allg. mindestens 4. Werden zu viele Korrespondenzen

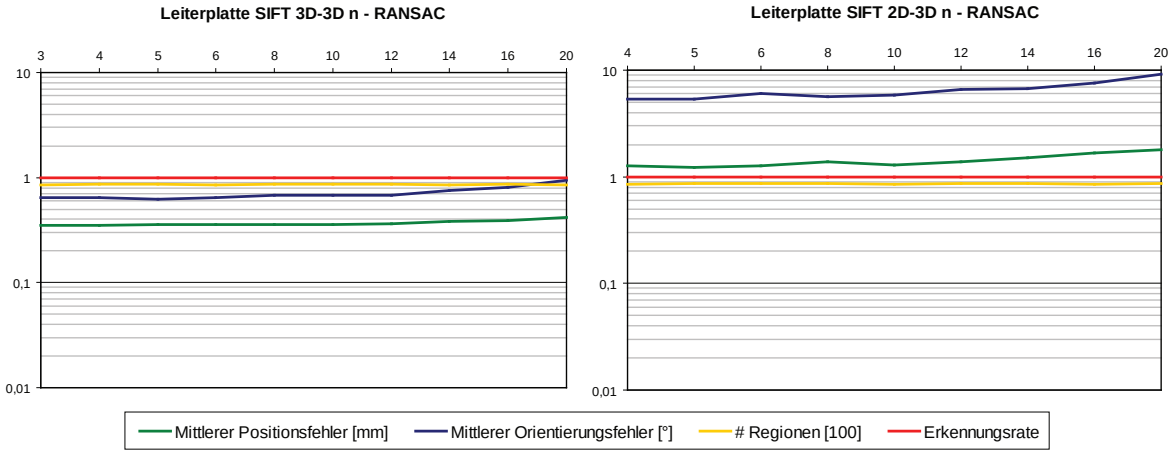

Abbildung 6.11: Einfluss der unterschiedlichen Anzahl verwendeter Korrespondenzen n_{RAN} pro RANSAC-Zyklus auf die Genauigkeit der globalen 3D-3D- bzw. 2D-3D-Lageermittlung. Bei großen Werten nimmt die Genauigkeit ab, es ist daher sinnvoller n_{RAN} nahe bei der minimal benötigen Anzahl (3 bzw. 4) an Korrespondenzen zu wählen.

pro Zyklus zur globalen Lageermittlung herangezogen, steigt die Wahrscheinlichkeit, dass sich in der verarbeiteten Menge ebenfalls Fehlkorrespondenzen befinden, bei zu wenigen Korrespondenzen kann dagegen die Genauigkeit leiden.

Abb. 6.11 zeigt den Einfluss der unterschiedlichen Einstellungen von n_{RAN} am Beispiel der Leiterplatte. Die Wahl dieses Parameters hat offensichtlich nicht den entscheidenden Einfluss, nur bei großen Werten nimmt die Genauigkeit deutlich ab. Als Einstellung in dieser Arbeit wird $n_{\mathrm{RAN}} = 5$ für die 3D-3D-Punktkorrespondenzen verwendet und $n_{\mathrm{RAN}} = 4$ für die 2D-3D- Punktkorrespondenzen. Die geringe Wahl bei 2D-3D ist, wie im Folgenden dargestellt, in der höheren Laufzeit und geringeren Anzahl an RANSAC-Zyklen begründet.

Als letzter RANSAC-Parameter verbleibt die Anzahl m_{RAN} der Zyklen. Je kleiner m_{RAN}, desto mehr bestimmt der Zufall das Ergebnis der globalen Lageermittlung und desto instabiler wird das System. Es ist daher sinnvoll m_{RAN} möglichst groß zu wählen, allerdings hängt die Rechenzeit in $O(m_{\mathrm{RAN}})$ von der Anzahl der Zyklen ab.

Abb. 6.12 zeigt diesen Zusammenhang am Beispiel der Leiterplatte. Bei den 3D-3D-Verfahren ist die lineare Abhängigkeit der Rechenzeit am besten zu erkennen, da dort jeder Zyklus eine feste Zeitspanne zur Berechnung benötigt. Bei den 2D-3D-Verfahren ist dies aufgrund der iterativen Berechnung nicht exakt gegeben, daher auch die leichten Abweichung der Laufzeiten von der Geraden. Ebenfalls lässt sich gut erkennen, das im Mittel das System mit zunehmender Anzahl an Zyklen genauer wird. Allerdings flacht dieser Effekt ab und nähert sich mit steigender Anzahl einer Asymptote an.

Die Wahl von m_{RAN} hängt also nur von der gewünschten Laufzeit ab. Hier werden die Werte so gesetzt, dass beide Verfahren ca. 100 ms zur Berechnung benötigen, d. h. $m_{\mathrm{RAN}} \approx 5000$ für die 3D-3D-Auswertung und $m_{\mathrm{RAN}} \approx 2500$ für die 2D-3D-Auswertung. Man beachte, dass sich die hier dargestellten Zeiten auf eine Testumgebung beziehen und nicht auf das endgültig optimierte System wie in Abschnitt 6.5.1. Bei einem Echtzeitsystem kann dieser Parameter natürlich in Abhängigkeit der verbleibenden Zeit eingestellt werden, da man insbesondere bei der 3D-3D-Auswertung den Zeitbedarf eines Zykluses kennt.

Mit diesen Einstellungen sind alle Parameter des Systems festgelegt. Daher können im nächsten Abschnitt die verschiedenen globalen Lagebestimmungen untersucht werden. Viele der folgenden

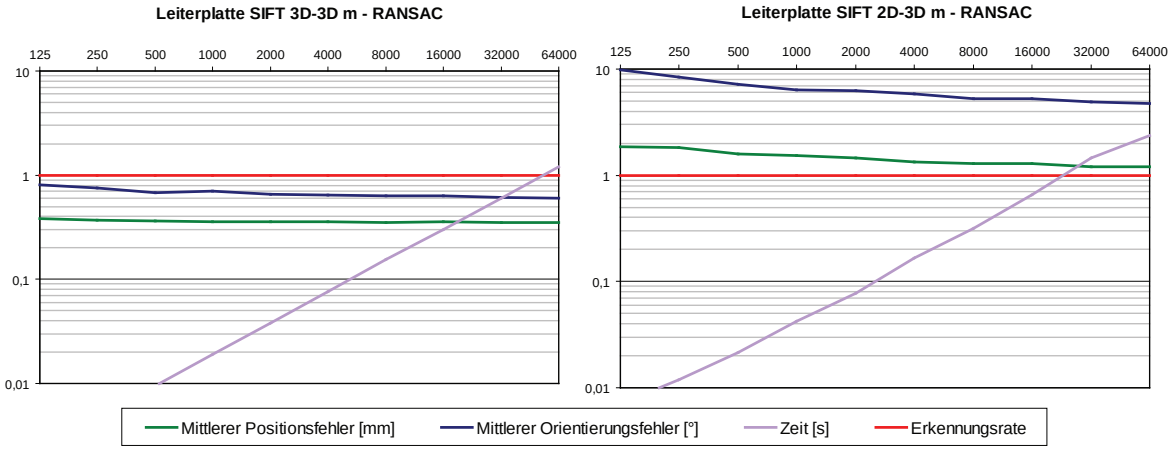

Abbildung 6.12: Genauigkeit und Geschwindigkeit der globalen Lageermittlung bei einer unterschiedlichen Anzahl m_{RAN} von RANSAC-Zyklen. Man beachte die doppelt logarithmische Darstellung, bei der der lineare Zusammenhang zwischen Rechenzeit und m_{RAN} weiterhin durch eine Gerade repräsentiert wird. Die Genauigkeit des Systems steigt mit zunehmender Anzahl, allerdings flacht dieser Effekt stark ab. Die Wahl von m_{RAN} hängt daher nur von der zur Verfügung stehenden Rechenzeit ab.

Ergebnisse sind auch schon in den vorangegangenen Diagrammen ersichtlich. Es wurde aber bis jetzt auf eine Diskussion darüber verzichtet um alle Algorithmen mit den besten ermittelten Einstellungen vergleichen zu können.

6.2 Evaluierung der globalen Lagebestimmung

In Abschnitt 4.3.4 werden verschiedene Methoden der globalen Lageermittlung aus den geclusterten lokalen Lagen vorgestellt. Dies sind eine iterative, algebraische Mittelung aus den lokalen Lagen, dem Clusterzentrum, welches eine Art 6D Median der lokalen Lagen darstellt, die Ermittlung der Lage aus 3D-3D-Punktkorrespondenzen der rückprojizierten Regionenzentren sowie aus 2D-3D-Punktkorrespondenzen. Im letzteren Fall werden nur die Bildpunkte der Regionenzentren im Suchbild genutzt. Dafür werden zwei Verfahren untersucht, der POSIT- und der OI-Algorithmus. Die Mittelung, der Median und das 3D-3D-Verfahren können zusätzlich noch mit einer Rekonstruktion der Tiefe über Stereotriangulation kombiniert werden, so dass die Tiefe der Regionenzentren nicht über den Skalierungsunterschied der Regionenzentren gewonnen werden muss, sondern über die robustere Stereoauswertung (vgl. dazu auch Abschnitt 6.1.2.1).

Die einzelnen Verfahren zur globalen Lageermittlung werden jeweils mit den besten, im vorangegangen Abschnitt 6.1 ermittelten Parametereinstellungen bzgl. Genauigkeit und benötigter Rechenzeit untersucht. Dazu werden die gleichen Ground-Truth-Sequenzen der Leiterplatte und der Box wie in Abschnitt 6.1 verwendet. Dabei sind Verkippungen der Suchansichten bzgl. der Modellansicht bis max. 15° enthalten. Abb. 6.13 zeigt die Ergebnisse der Auswertung, allerdings wird der übersichtshalber nur der mittlere Orientierungsfehler angegeben. Dieser korreliert stark mit dem nicht dargestellten Translationsfehler und ist daher aussagekräftig genug. Die dargestellten Zeiten beziehen sich wiederum auf die Testumgebung und nicht auf das endgültig optimierte System.

Vergleicht man zuerst einmal den ähnlich-kovarianten SIFT-Detektor mit dem affine-kovarianten MSER-Detektor, fällt auf, dass SIFT in fast allen Fällen besser abschneidet als MSER, bei 3D-3D inkl. Stereo oder 2D-3D-POSIT sogar deutlich. SIFT ist nur bei der Mittelung und dem Median der

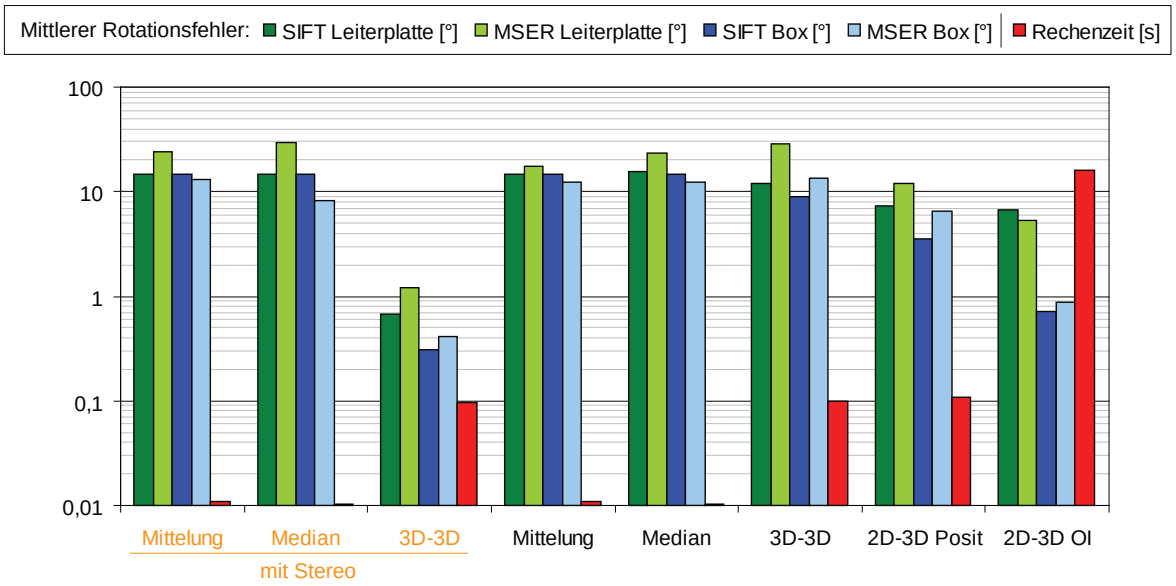

Abbildung 6.13: Vergleich der verschiedenen Algorithmen zur globalen Lageauswertung bei Verwendung von SIFT (dunkel) und MSER (hell) für die Leiterplatte (grün) und die Box (blau). Bei den ersten drei Verfahren wird die Tiefe der Suchregionen nicht über den Skalierungsunterschied sondern über eine robuste Stereotriangulation ermittelt. Man erkennt, dass die, auf Punktkorrespondenzen basierenden Algorithmen 3D-3D (inkl. Stereo), 2D-3D-POSIT bzw. OI den, auf den lokalen Lagen basierenden Verfahren Mittelung und Median deutlich überlegen sind.

Box zweimal geringfügig schlechter, allerdings sind bei diesen Verfahren alle Ergebnisse so schlecht, dass sie als Vergleich nicht zählen. Auch hier zeigt sich wieder wie in Abschnitt 6.1.2.2, dass, obwohl der 4 DoF SIFT-Detektor keine Verkippung aus der Kameraebene heraus schätzen kann, er dem 6 DoF MSER-Detektor mit voller Lagerekonstruktion überlegen ist. Die Anzahl der geclusterten Regionen für die globale Lageschätzung ist im Durchschnitt bei SIFT (83,5) und MSER (83,7) dieselbe. Das bessere Abschneiden von SIFT kann also nicht durch mehr detektierte Regionen oder eine signifikant geringere Anzahl Fehlkorrespondenzen erklärt werden, sondern muss mit dem Rauschen der Regionen zusammenhängen. Weiterhin werden von den drei überlegenen Verfahren, dem 3D-3D inkl. Stereo, 2D-3D-POSIT und 2D-3D-OI nur die Regionenzentren bei der globalen Lageermittlung ausgewertet und nicht die vollständige Region. Auch bei diesen Verfahren schneidet SIFT besser ab, es liegt also die Vermutung nahe, dass SIFT die Regionenzentren robuster detektiert. Eine mögliche Begründung dafür könnten die salienten Strukturen geben, auf die beide Detektoren reagieren, in Verbindung mit der hier verwendeten Beleuchtungseinheit. SIFT bevorzugt Ecken und kleine Blobs während MSER homogene Flächen benötigt. Aufgrund der flächigen Ring-Beleuchtung treten verstärkt großfläche Glanzpunkte auf, die zwar die Ecken und starken Blobs meist erkennbar lassen, die homogenen Flächen aber stark stören.

Betrachtet man das Abschneiden der unterschiedlichen Algorithmen zur Ermittlung der globalen Lage, fällt auf, dass die Verfahren die direkt auf einer Auswertung der lokalen Lage beruhen, d. h. die Mittelung und der Median sehr schlecht abscheiden bzw. ein extrem hohes Rauschen aufweisen. Der mittlere Orientierungsfehler ist sogar größer als die maximal zu detektierende Verkippung der Ground-Truth-Sequenzen, d. h. in dem untersuchten Bereich sind Mittelung und Median unbrauchbar. Dies ist auf die lokalen Lagen zurückzuführen, die aufgrund des stark begrenzten Szenenausschnitts ihrer zugehörigen Regionenkorrespondenzen eine zu große Ungenauigkeit aufweisen. Für

den Gruppierungsschritt spielt dies eine untergeordnete Rolle, da dort aufgrund der 6 DoF's beim Clustern größere Toleranzen akzeptabel sind, als bei der exakten Lageauswertung. Möchte man die Genauigkeit der Lagen weiter verbessern, kann man den Ansatz von (KK08) verwenden, um die korrespondierenden Regionen *nach* der Korrespondenzfindung iterativ weiter zu verfeinern und eine exaktere lokale Lage zu erhalten. Dies ist allerdings aufwendig, daher wurden in dieser Arbeit als Alternativen die globalen Lageauswertungen auf Basis von Punktkorrespondenzen entwickelt. Sie greifen einerseits nur auf die Regionenzentren zurück und verwenden andererseits mehrere Korrespondenzen, die im Idealfall über das gesamte Objekt verstreut sind. Sie nutzen daher nicht nur einen kleinen Ausschnitt der Szene, sondern verwenden Informationen die über einen großen, globalen Bereich verteilt sind.

Für die Genauigkeit des Algorithmus mit 3D-3D-Punktkorrespondenzen ohne Stereo, d. h. mit einer Tiefenschätzung der Regionenzentren über den lokalen Skalierungsunterschied der korrespondierenden Regionen, gilt dieselbe Argumentation wie bei der Mittelung und dem Median. Stabilisiert man dagegen die Tiefe mit einer stereobasierten Triangulation, erhält man den genauesten Algorithmus, der in dieser Arbeit entwickelt wurde. Er erreicht auf der Ground-Truth-Sequenz der Box einen mittleren Orientierungsfehler von 0.3° und ist mit einer mittleren Rechenzeit von 100 ms auch noch sehr schnell. Allerdings benötigt die Stereo-Tiefenrekonstruktion eine deutlich längere Zeit zur Berechnung des Tiefenbildes, was bei den hier präsentierten Zeiten nicht berücksichtigt ist. Deshalb wurden alternativ die Verfahren auf Basis von 2D-3D-Punktkorrespondenzen untersucht, die zwar aufgrund ihrer iterativen Struktur pro RANSAC-Zyklus langsamer sind, dafür aber auch ohne Stereo-Tiefenrekonstruktion auskommen. Ihre Genauigkeit ist allerdings um mind. eine Größenordnung geringer, wobei der OI-Algorithmus bei gleichen Einstellungen POSIT deutlich überlegen ist. Er ist aber auch um ca. den Faktor 100 langsamer als POSIT und mit einer mittleren Auswertezeit von über 10 s (in der nicht optimierten Testumgebung) auch deutlich langsamer als die 3D-3D-Auswertung inkl. Stereo.

Es zeigt sich weiterhin, dass die Leiterplatte in allen relevanten Fällen bei der Genauigkeit schlechter ausfällt als die Box. Dies liegt u. a. an der Beschaffenheit der Platte, sie glänzt wesentlich mehr und erzeugt daher größere und stärkere Glanzpunkte als die relative matte Oberfläche der Box.

Als Fazit lässt sich daher sagen, dass die 3D-3D-Auswertung inkl. Stereo-Tiefenrekonstruktion bzgl. der Genauigkeit allen anderen Algorithmen überlegen ist. Sie ist allerdings bei Berücksichtigung der Rechenzeit der Stereoauswertung um mind. den Faktor 5 langsamer als die 2D-3D-Auswertung mit POSIT. Letztere ist dagegen um den Faktor 10 schlechter in der Genauigkeit. Steht einem keine Stereoauswertung zur Verfügung (da man z. B. nur eine einzelne Kamera zur Verfügung hat), kann mittels dem OI-Algorithmus mit einer wesentlich höheren Laufzeit dennoch eine bessere Genauigkeit wie mit POSIT erzielt werden. Welches System nun bevorzugt wird, hängt von den Rahmenbedingungen und der Aufgabe ab. Es kann daher an dieser Stelle keine endgültige Empfehlung gegeben werden. Die endgültigen Zeiten der optimierten Systeme werden in Abschnitt 6.5.1 ermittelt.

6.3 Einfluss mehrerer Ansichten und Detektoren

Bei allen vorangegangenen Experimenten wurde jeweils nur ein einzelner Detektor und eine einzelne Modell- bzw. Suchansicht verwendet. In diesem Abschnitt soll nun für die besten Systeme (3D-3D inkl. Stereo und 2D-3D-OI) der Einfluss mehrerer kombinierter Regionendetektoren bzw. multipler Modell- und Suchansichten untersucht werden. Insbesondere das integrierte Behandeln mehrerer Ansichten ist ein zentraler Punkt der Konzeption (vgl. Abschnitt 4.2 und folgende) und unterscheidet den Ansatz von (Low99), wo für jede einzelne Kombination von Modell- und Suchansicht eine eigene Lokalisation durchgeführt werden muss.

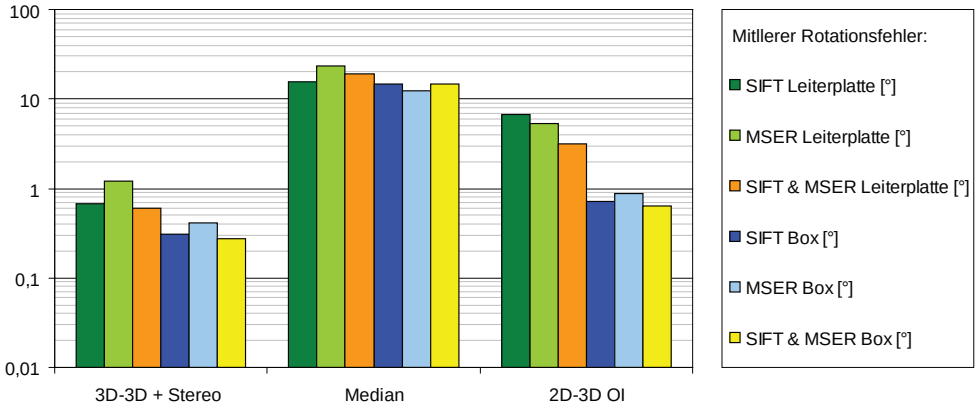

Abbildung 6.14: Das gleiche Experiment wie in Abb. 6.13 für die Leiterplatte und die Box, diesmal allerdings nur für die besten zwei globalen Auswertungen 3D-3D inkl. Stereo und 2D-3D-OI sowie dem Median, dafür aber zusätzlich eine Auswertung mit SIFT und MSER kombiniert (orange und gelb). Man erkennt bei der kombinierten Auswertung der Detektoren aufgrund der größeren Anzahl zur Verfügung stehender Korrespondenzen eine Erhöhung der Genauigkeit um ca. 10% bzgl. der besseren der beiden einzelnen Auswertungen, bei der Leiterplatte 2D-3D-OI sogar eine Verbesserung des mittleren Orientierungsfehlers von 5° auf 3°. Nur der Median kann von der Kombination nicht profitieren.

6.3.1 Detektoren

In allen Experimenten wurden die Regionendetektoren SIFT und MSER bis jetzt separat eingesetzt. In diesem Abschnitt werden sie dagegen gemeinsam verwendet und die lokalen Lagen von SIFT- bzw. MSER-Regionenkorrespondenzen integriert bei der Gruppierung und der Lageauswertung berücksichtigt. Dies wird u. a. durch die Überführung der Korrespondenzen in die lokalen Lagen ermöglicht, da diese Repräsentation unabhängig von der Art der zugrundeliegenden Regionen ist. Dies ist insofern von Vorteil, da SIFT (Blobs und Ecken) und MSER (homogene Flächen) komplementäre Strukturen zur Konstruktion der Regionen verwenden (vgl. auch Abschnitt. 3.2) und daher bei gemeinsamer Verwendung ein größeres Spektrum an Objektstrukturen ausgewertet werden kann. Abb. 6.14 zeigt dieselben Experimente wie Abb. 6.13 zur Evaluierung der globalen Lageauswertung, allerdings werden nur noch ausgewählte Verfahren präsentiert und für diese zusätzlich die Ergebnisse mit einer kombinierten Auswertung von SIFT und MSER (orange und gelb) dargestellt.

Eine kombinierte Auswertung führt zu mehr gefundenen Regionen und daher auch zu mehr Korrespondenzen bzw. mehr abgeleiteten lokalen Lagen. Dies erhöht aufgrund der größeren Anzahl lokaler Lagen die Robustheit beim Gruppieren. Dies zeigt sich z. B. bei der (nicht in der Abbildung dargestellten) Detektionsrate des Handys bei Auswertung der Ground-Truth-Sequenz. Diese ist bei SIFT 0.99, bei MSER 0.94 und erst bei der kombinierten Auswertung bei 1.00, d. h. nur dort wurde das Objekt in allen 324 Bildern gefunden. Ähnlich verhält es sich durch SIFT (0.99), MSER (0.86) und kombiniert (1.00) bei der Detektionsrate des Steckers.

Ein robusteres Gruppierungsergebnis lässt auch genauere Cluster und damit ein genaueres Clusterzentrum, d. h. eine genauere globale Auswertung durch den Median erwarten. Wie Abb. 6.14 allerdings zeigt, ist dies nicht der Fall. Die höhere Anzahl nützt in diesem Fall nichts, die lokalen Lagen sind einfach zu ungenau für ein robuste, finale Lokalisation.

Im Gegensatz dazu profitieren die Algorithmen auf Basis von Punktkorrespondenzen von den zusätzlichen Korrespondenzen. Die Genauigkeit erhöht sich bei der kombinierten Auswertung bzgl. der besseren der beiden Einzelauswertungen um ca. 10%, bei der Leiterplatte 2D-3D-OI von einem

mittleren Rotationsfehler von $\approx 5^\circ$ auf $\approx 3^\circ$ sogar deutlich um 40%. Es soll allerdings nicht unerwähnt bleiben, dass dieser Effekt bei Bauteilen mit ungünstigen Strukturen nicht zutreffen muss. So liegt die mittlere Genauigkeit von dem (nicht dargestellten) Stecker mit SIFT bei 1.2°, mit MSER bei 69.6° (mit anderen Worten MSER funktioniert auf diesem Bauteil nicht) und bei einer kombinierten Auswertung bei 23.2°. Es ist bei diesem Worst-Case Szenario also eine Verschlechterung eingetreten.

Als Fazit lässt sich sagen, das sowohl die Robustheit als auch die Genauigkeit bei Verwendung mehrerer Detektoren im allgemeinen verbessert wird. Dies ist insbesondere bei Bauteilen von Vorteil, bei denen ein einzelner Detektor nur wenige stabile Regionen findet. Eine Kombination vieler komplementärer Detektoren ist daher zu empfehlen. Abschnitt 3.2.6 gibt dafür konkrete Vorschläge, die neben SIFT und MSER sinnvoll sind. Allerdings muss bei ungünstigen Bauteilen geprüft werden, ob einzelne Detektoren nicht robust funktionieren und das Gesamtsystem daher mehr stören als ihm Nutzen bringen.

6.3.2 Modellansichten

Beschränkt man sich beim Einlernen des Modells auf eine einzelne Ansicht, so lässt sich nur ein bestimmter Bereich an Blickwinkeländerungen bei der Lokalisation abdecken. Wie im geometrischen Modell in Abschnitt 3.1.2 hergeleitet, führen Verschiebungen des Objekts entlang der X- bzw. Y-Achse der Kamera zu einer Verschiebung der Intensitätsinformationen in der Bildebene, Verschiebungen entlang der Z-Achse zu einer Skalierung der Intensitätsinformationen und eine Rotation um die Z-Achse zu einer Rotation der Intensitätsinformationen in der Bildebene. All diesen Transformationen ist gemeinsam, dass unabhängig von der Objektgestalt keinerlei Informationen in der Bildebene verloren gehen. Sie lassen sich ohne Approximationen durch eine ST in der Bildebene modellieren und durch alle ähnlich-kovarianten Detektoren behandeln.

Verkippungen um die X- bzw. Y-Achse der Kamera führen bei allgemeinen, nicht-planaren Oberflächen durch die Projektion zu linear nicht modellierbaren Intensitätsänderungen, da bestimmte Bereiche des Objekts verdeckt bzw. andere Bereiche des Objekts sichtbar werden. Kein ohne weiteres Wissen ausgestatteter Detektor kann diese Änderungen in der Bildebene behandeln. Unter bestimmten Annahmen (vgl. Abschnitt 3.1.2), u. a. einer lokalen Ebenenannahme, können Verkippungen zumindest auf eine affine Verzerrung der Intensitätsinformationen zurückgeführt werden und dementsprechend von affin-kovarianten Detektoren behandelt werden. In der Praxis werden diese Annahmen von realen Objekten nur eingeschränkt erfüllt, so dass mit zunehmendem Verkippungswinkel α die Modellierungsfehler des geometrischen Modells größer und die Lokalisation damit unrobuster wird. Aber selbst bei einer perfekten Ebene als Oberfläche, kann, bei senkrecht zur Ebene eingelernter Modellansicht ($\alpha = 0^\circ$), theoretisch nur ein Bereich von $\alpha \in]-90^\circ, 90^\circ[$ bei der Lokalisation abgedeckt werden, da immer nur ein Bereich der Ebene proportional zum $\cos(\alpha)$ in der Bildebene beobachtbar ist. Bei $\alpha = \pm 90^\circ$ ist die Ebene daher nicht mehr sichtbar, in der Praxis bricht die Reproduzierbarkeit der Detektoren (vgl. Abb. 3.8) allerdings früher bei $\alpha \approx \pm 60^\circ$ ein. Es ist daher unvermeidbar, mehrere Ansichten mit unterschiedlichen Verkippungen einzulernen, um über einen großen Blickwinkelbereich ein robustes System zu erhalten. Bei nicht-planaren Objekten ist dieser Vorgang selbst bei kleineren Blickwinkeländerungen essentiell, um die Approximationsfehler in Grenzen zu halten.

Aufgrund der Überführung der gefundenen Regionenkorrespondenzen in die ansichtsunabhängige, universelle Darstellung mittels lokaler Lagen (siehe Abschnitt 4.3.3), können im Gruppierungsschritt die Korrespondenzen aller Modellansichten in einem Cluster fusioniert werden. Daraus lässt sich dann in einem integrierten Schritt mittels der globalen Lageermittlung eine auf allen Modellansichten basierende einzelne Lagehypothese ermitteln. Verfahren wie (Low99), die eine Gruppierung der Korrespondenzen nicht im 3D-Raum sondern in der Bildebene durchführen, müssen dagegen jede Modellansicht getrennt behandeln und dann aufgrund eines Gütekriteriums entscheiden, welche Modellansicht zur besten Lage geführt hat. Dies lässt sich bei dem hier neu entwickelten Verfahren

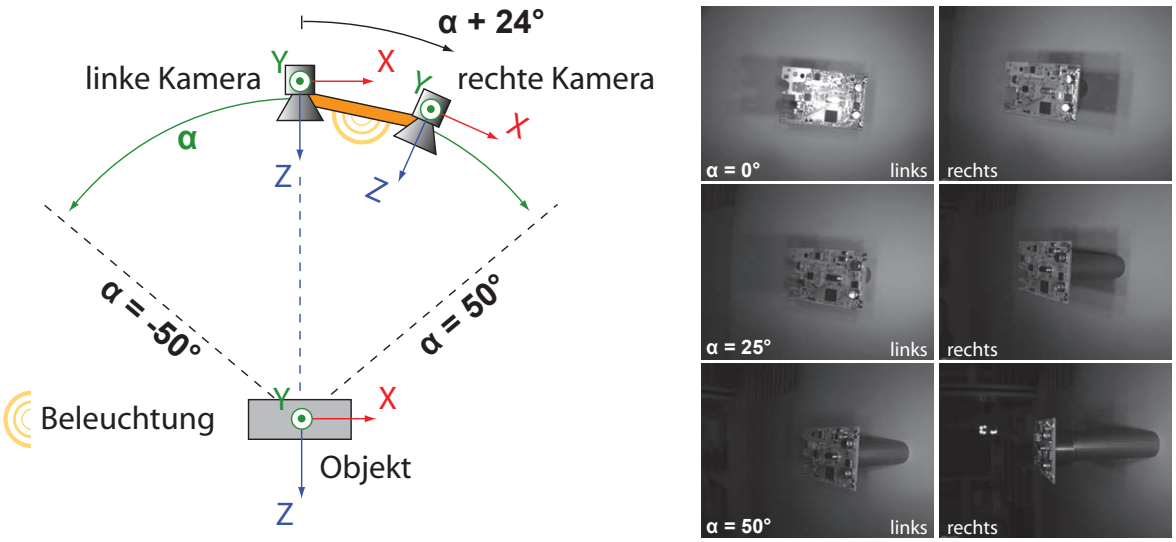

Abbildung 6.15: Sensorikaufbau für die Versuche mit mehreren Ansichten. Eine Stereokamera wird samt Beleuchtung in einem Abstand von 17 cm mit dem Verkippungswinkel α um die, zu den Y-Achsen der Kameras parallele Y-Achse des Objekts rotiert bzw. verkippt. α bezieht sich auf die Verkippung der linken Kamera, wobei für $\alpha = 0^\circ$ die Z-Achse der linken Kamera parallel zur Normalen der dominanten Ebene des Objektes steht. Die rechte Kamera ist um ca. $\alpha + 24^\circ$ versetzt. Die Verkippung wird über einen Roboter realisiert, dessen kinematisches Modell die Ground-Truth-Informationen über α bereitstellt. Der maximal abfahrbare Bereich beträgt $\alpha \in [-50^\circ, 50^\circ]$. Rechts sind für verschiedene Verkippungswinkel die Suchansichten der Leiterplatte beider Kameras dargestellt.

vermeiden und führt zu einem effizienteren und leichter handhabbaren System.

Um die Auswirkungen mehrerer Modellansichten besser nachvollziehen zu können, wird der in Abb. 6.15 dargestellte Aufbau verwendet. Die linke Kamera verkippt dabei mit dem Winkel α um die Y-Achse des Objekts (welche parallel zur Y-Achse der Kamera steht), die rechte Kamera ist um ca. $\alpha + 24^\circ$ versetzt. Einlernen und Lokalisierung erfolgt in diesem Abschnitt mit der linken Kamera, wobei das erste Modellbild immer bei $\alpha = 0^\circ$ senkrecht zur dominantesten Ebene des Objekts aufgenommen wird. Es werden fünf Modelle mit 1, 3 (alle 30°), 5 (alle 20°) und 11 (alle 10°) Modellansichten mit unterschiedlichen Verkippungswinkel α trainiert, wobei der Abstand zwischen den einzelnen Ansichten der Modelle jeweils gleich ist. Für die Evaluation werden mittels eines Roboters in 0.25° Schritten im Bereich $\alpha \in [-50^\circ, 50^\circ]$ Suchansichten aufgenommen und die Orientierungsabweichung der ermittelten 6 DoF Lagehypothese zu der Ground-Truth-Orientierung ermittelt. Die Ergebnisse sind für die Leiterplatte in 6.16 zu sehen, wobei die Kurven durch eine gleitende Mittelung über 9 Werte ($\pm 1^\circ$) geglättet wurden.

Der Abdeckungsbereich der Blickwinkeländerungen bei einer Modellansicht beläuft sich auf ca. $\alpha \in [-37^\circ, 25^\circ]$. Diese Asymmetrie lässt sich in abgemilderter Form auch bei allen anderen Kurven erkennen. Sie lässt sich auf die Güte des Stereo-Tiefenbildes zurückführen, welches aufgrund der um $\alpha + 24^\circ$ verschobenen rechten Kamera bei größeren positiven Verkippungen ein immer unterschiedlicheres Stereo-Bildpaar bearbeiten muss.

Vergleicht man die Ergebnisse von SIFT und MSER bei einer Modellansicht, fällt auf, dass der Abdeckungsbereich nahezu der gleiche ist, SIFT wie in den vorangegangenen Experimenten allerdings wieder etwas genauer und robuster. Dies ist in diesem Fall erstaunlich, da der affin-kovariante

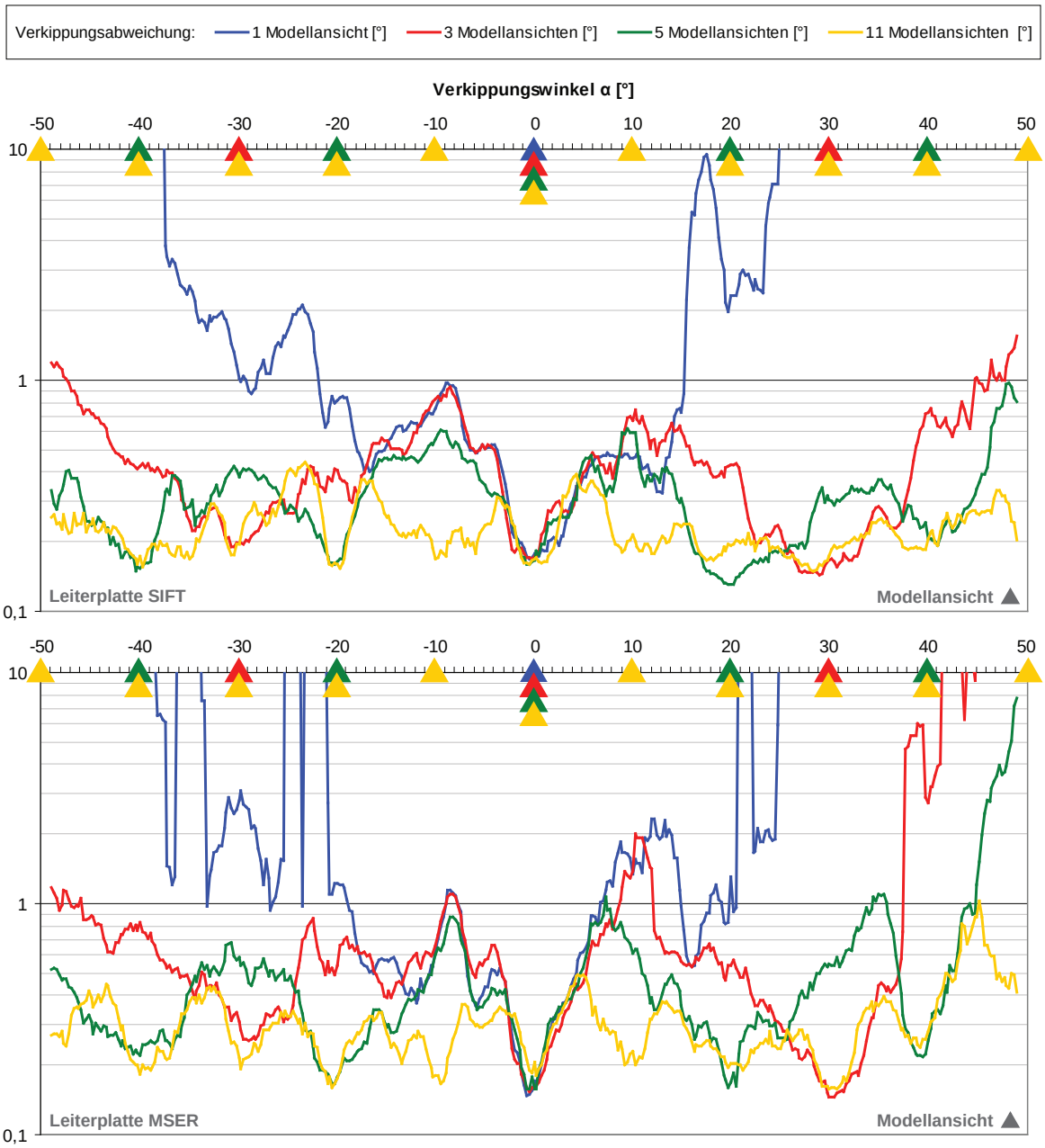

Abbildung 6.16: Auswirkung mehrerer Modellansichten auf die Lokalisationsgenauigkeit des Systems bei Verwendung der Leiterplatte und 3D-3D inkl. Stereo für SIFT (oben) und MSER (unten). Die Dreiecke geben an, bei welchen Verkippungswinkel α die Modellansichten eingelernt wurden. Dargestellt ist die über $\pm 1^\circ$ gemittelte Orientierungsabweichung des 6 DoF Lokalisationsergebnisses zu der Ground-Truth-Orientierung. Es lässt sich anhand der Kurven leicht erkennen, bei welchen Winkeln eine Modellansicht eingelernt wurde, da dort die Abweichung meist unter 0.2° fällt. Ebenfalls ist gut zu erkennen, dass mit zunehmender Anzahl Modellansichten die mittlere Abweichung immer weiter sinkt. Erstaunlich ist, dass der ähnlich-kovariante SIFT-Detektor den gleichen Bereich wie der affin-kovariante MSER-Detektor abdeckt, obwohl der SIFT-Detektor im Gegensatz zum MSER-Detektor keine Verkippung berücksichtigen kann.

MSER-Detektor Verkippungen um die Y-Achse für planare Objekte (wie in erster Näherung die hier verwendete Leiterplatte) explizit modelliert, der ähnlich-kovariante SIFT-Detektor dagegen nicht. Die gleichen Ergebnisse erhält man auch für andere, fast ideal planare Objekte wie die Box. Eine andere als die schon ausgeführte Erklärung mit störenden Glanzpunkten auf homogene Flächen kann hierfür nicht gegeben werden.

Untersucht man den Abdeckungsbereich der Blickwinkeländerungen für die Modelle mit einer unterschiedlichen Anzahl von Modellansichten wird sofort ersichtlich, dass dieser stark erweitert wird und die mittlere Abweichung mit einer zunehmenden Anzahl von Ansichten stark sinkt. Ebenfalls ist klar ersichtlich, bei welchen Verkippungswinkeln die Modellansichten eingelernt wurden, da dort die Abweichungen minimal werden. Es genügen schon 3 Modellansichten um bei der Leiterplatte mit SIFT eine Orientierungsabweichung von unter 1° fast im gesamten betrachteten Bereich zu erhalten. Weitere Ansichten reduzieren vor allem den Abweichungsfehler, bei 11 Ansichten erreicht man im gesamten Bereich einen max. Fehler von unter 0.4°. Das Gesamtsystem profitiert daher stark von weiteren Modellansichten, allerdings muss man berücksichtigen, dass die Modelldatenbank größer und die Korrespondenzsuche langsamer wird. Nach der Korrespondenzfindung erhält man aufgrund des integrierten Konzepts allerdings keine weiteren Nachteile.

Wieviele Ansichten man nun tatsächlich für ein Objekt einlernen sollte, hängt daher stark vom abzudeckenden Bereich, der geforderten Genauigkeit, der zur Verfügung stehenden Zeit und natürlich der Objektoberfläche ab. Je ausgeprägter die Textureigenschaften und je besser die Approximationen wie die Ebenheit von dem Objekt erfüllt werden, desto geringer die Anzahl der benötigen Modellansichten. Für die Leiterplatte werden bei diesem Aufbau daher ca. alle 30° eine Ansicht benötigt, bei der nahezu optimalen Box alle $50-60^\circ$ und bei dem wesentlich schwierigeren Stecker mindestens alle 10°.

6.3.3 Suchansichten

Es können nicht nur während dem Training mehrere Ansichten eingelernt werden, sondern auch während der Lokalisationsphase alternativ bzw. zusätzlich mehrere Suchansichten verwendet werden. Auch in diesem Fall können Korrespondenzen aller Ansichten über den Gruppierungsschritt fusioniert und anschließend integriert ausgewertet werden. Eine getrennte Auswertung wie in (Low99) ist in diesem Fall noch ineffizienter, da für jede mögliche Kombination von Such- und Modellansicht eine eigene Lokalisation durchgeführt werden muss.

Bei Verwendung mehrerer Suchansichten muss allerdings gewährleistet werden, dass diese bei dynamischen Szenen zeitgleich aufgenommen werden und alle Suchansichten miteinander registriert sind. Für die unterschiedlichen Suchansichten wird in diesem Abschnitt die linke und rechte Kamera herangezogen und die gleichen Experimente (allerdings nur mit SIFT) wie für die multiplen Modellansichten (siehe Abb. 6.15) durchgeführt. Abb. 6.17 zeigt diesmal nicht nur die Ergebnisse für die linke Kamera (blau), sondern ebenfalls die Ergebnisse für die rechte Kamera (rot) und die kombinierte Auswertung bei Verwendung der linken und rechten Kamera.

Dabei werden zwei Kombinationsmöglichkeiten untersucht, einmal die integrierte Behandlung durch das in dieser Arbeit entwickelte System (grün) und zum anderen eine getrennte Auswertung von linker und rechter Kamera und die Auswahl der besseren der zwei zurückgegebenen Lagen (gelber Punkt) über die in Abschnitt 4.3.4 vorgestellten Qualitätsmaße $q_a(\cdot)$. Für die Auswertung mit 3D-3D inkl. Stereo ist dies in Gl. 4.22 beschrieben, für die globale Lageermittlung mit 2D-3D-OI in Gl. 4.24. Die Auswahl mittels Qualitätspräferenz simuliert dabei ein Vorgehen, wie es bei einer getrennten Auswertung aller Ansichten erfolgen müsste.

Gut zu erkennen sind die unterschiedlichen Abdeckungsbereiche der Blickwinkeländerungen aufgrund des Versatzes beider Kameras, da sich beide Auswertungen auf *dasselbe* Modell mit eingelernten Ansichten der *linken* Kamera beziehen. Allerdings sollte man erwarten, dass die Abweichung

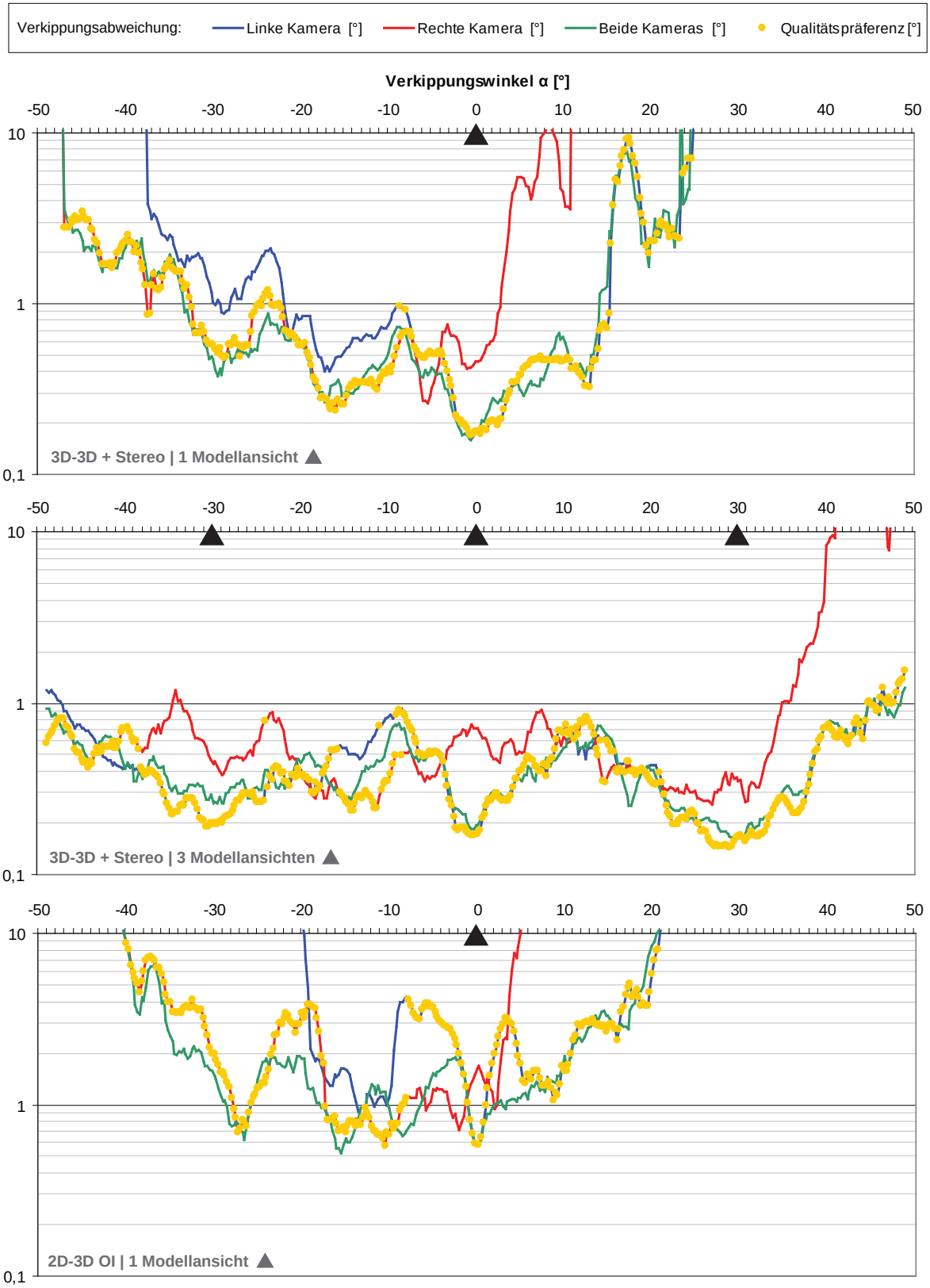

Abbildung 6.17: Auswirkung mehrerer Suchansichten auf die Lokalisationsgenauigkeit des Systems bei Verwendung der Leiterplatte und SIFT für die globale Auswertung mittels 3D-3D inkl. Stereo (oben u. Mitte) und 2D-3D-OI (unten). Blau und rot sind die getrennten Auswertungen für die linke und rechte Kamera angegeben, grün die fusioniere Auswertung mit dem in dieser Arbeit vorgestellten neuen System und mittels gelber Punkte markiert die Auswahl der vermeintl. besseren der beiden, getrennt ermittelten Lagen über das Qualitätsmaß $q_a(\cdot)$.

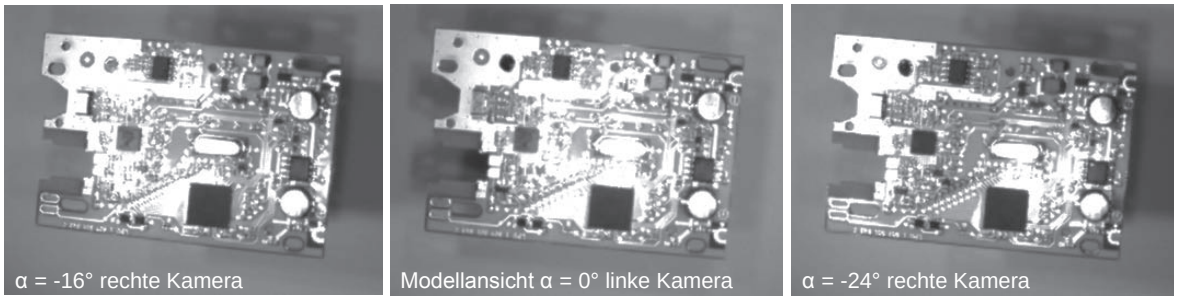

Abbildung 6.18: Geometrische und photometrische Unterschiede bei Suchansichten mit unterschiedlichen Verkippungen. Die Suchansicht der rechten Kamera ist bei $\alpha = -24^\circ$ zwar geometrisch identisch mit der Modellansicht der linken Kamera bei $\alpha = 0^\circ$, die rechte Ansicht bei $\alpha = -16^\circ$ ist aufgrund der Beleuchtung dagegen vom Erscheinungsbild identischer.

bei der Auswertung der rechten Kamera bei $\alpha = -24^\circ$ am kleinsten ist, da dort die rechte Kamera genau im selben Winkel zum Objekt steht wie die linke Kamera während des Trainings. Die kleinste Abweichung ist allerdings bei $\alpha = -16^\circ$ zu beobachten. Dies ist auf die Beleuchtung des Objekts zurückzuführen. Abb. 6.18 macht ersichtlich, dass zwar die rechte Suchansicht bei $\alpha = -24^\circ$ geometrisch identisch mit der eingelernten Modellansicht der linken Kamera bei $\alpha = 0^\circ$ ist, die rechte Ansicht bei $\alpha = -16^\circ$ bis auf eine kleine geometrische Stauchung aufgrund der Beleuchtung allerdings vom Erscheinungsbild identischer ist. Offensichtlich dominieren hier die photometrischen die geometrischen Unterschiede.

Beide Arten der Kombination, die integrierte Behandlung (grün) und die getrennte Auswertung mit Auswahl über das Qualitätsmaß (gelb) sind von der Abweichung in den Randbereichen ähnlich. Dort dominiert praktisch eine der beiden Suchansichten und die Auswahl über $q_a(\cdot)$ funktioniert zuverlässig. Überschneiden sich dagegen die Einzugsbereiche der Suchansichten, wie es im Bereich $\alpha \in [-10^\circ, 0^\circ]$ der Fall ist, kommt es insbesondere bei 2D-3D-OI zu Sprüngen der Abweichung aufgrund der $q_a(\cdot)$-Auswahl. In diesem Bereich ist die integrierte Auswertung robuster und zeigt den Vorteil der neu entwickelten Gruppierung über lokale Lagen.

Vergleich man den mittleren Fehler im Einzugsbereich der kombinierten Auswertungen bei 1 eingelernten Modellansicht, dies ist $\alpha \in [-47^\circ, 25^\circ]$ für 3D-3D inkl. Stereo bzw. $\alpha \in [-40^\circ, 21^\circ]$ für 2D-3D-OI, so liegt die integrierte Auswertung mit 3D-3D inkl. Stereo bei 0.91° und die Qualitätspräferenz bei 0.94°. Bei drei Modellansichten ist die mittlere Abweichung ebenfalls nahezu identisch bei 0.41° und 0.40°. Die Lageermittlung mit 2D-3D-OI profitiert dagegen sehr von der neuen Methode, der mittlere Fehler liegt bei der integrierten Auswertung bei 1.67° und bei der Qualitätspräferenz um ca. die Hälfte mehr bei 2.38°. Dies lässt sich u. a. damit begründen, dass bei der Lageermittlung mittels 3D-3D inkl. Stereo die Informationen der rechten Suchansicht schon über die Stereotriangulation verwertet werden und der Mehrgewinn durch die zusätzlichen Korrespondenzen sich in Grenzen hält. Bei der Lageermittlung mit 2D-3D-OI fließen dagegen keine Disparitätsinformationen in die Auswertung mit ein, die Hinzunahme der Korrespondenzen der rechten Ansicht bringt daher einen deutlichen Mehrwert.

Dies wird besonders ersichtlich bei der senkrechten Betrachtung der Objektebene aus der linken Suchansicht, da bei 2D-3D-OI der Verkippungswinkel α durch die Stauchung des Objekts geschätzt werden muss. Es lässt sich daher nur der $\cos\alpha$ in der Suchansicht beobachten, der im Bereich $\alpha \in [-8^\circ, 8^\circ]$ nahezu identisch ist ($\cos 0^\circ = 1$, $\cos 8^\circ \approx 0.99$) und die Genauigkeit der Orientierung in diesem Bereich daher sehr leidet. Ausgenommen davon ist der Bereich unmittelbar in der Nähe der eintrainierten Modellansicht, der aufgrund der nahezu identischen Detektion der Regionen in

den Suchansichten und der Modellansicht hervorragend funktioniert. Erst bei größeren Verkippungen verändert sich der Cosinus stärker und die Genauigkeit wird deutlich besser, wie in Abb. 6.17 im untersten Diagramm gut ersichtlich. Bei Verwendung mehrerer Suchansichten ist dagegen immer eine Ansicht dabei, die die Ebene unter einem größeren Verkippungswinkel beobachtet und eine genaue Auswertung ermöglicht.

Als Fazit lässt sich sagen, sowohl die integrierte Auswertung mit 3D-3D inkl. Stereo als auch die integrierte Auswertung mittels 2D-3D-OI profitiert von mehreren Suchansichten ohne sich wie bei der Qualitätspräferenz explizit für eine Ansicht entscheiden zu müssen. Allerdings ist dieser Effekt ohne Verwendung einer Stereotriangulation bei 2D-3D-OI aufgrund des größeren Informationsmehrwerts deutlich ausgeprägter.

6.4 Evaluierung der optionalen Erweiterungen

Die in Abschnitt 4.5 vorgestellten Erweiterungen des in dieser Arbeit entwickelten Grundkonzepts sind für das System nicht essentiell, behandeln aber Problematiken die bis dorthin noch nicht explizit berücksichtigt wurden. Dies sind der Umgang mit Mehrdeutigkeiten, die bei Objekten (wie dem Stecker) mit ähnlichen Strukturen auftreten und die Optimierung der Modelldatenbank. Sie werden in den folgenden Abschnitten behandelt.

6.4.1 Eliminierung ähnlicher Suchregionen

Eine Problematik der lokalen Betrachtungsweise vor der Gruppierung und damit einhergehenden globalen Betrachtungsweise ist die Verwechslungsproblematik bei der Korrespondenzfindung. Dies passiert, wie z. B. in Abb. 4.6 ersichtlich, falls sich auf dem betrachteten Objekt zu viele ähnliche Objektstrukturen befinden. Dadurch sind die Merkmalsvektoren $\mathbf{o}$ der extrahierten Regionen nicht mehr eindeutig und führen zu Fehlkorrespondenzen während der Zuordnung zu der Modelldatenbank. Wie in Abschnitt 4.5.1 beschrieben, ist eine Möglichkeit dieses Problem zu adressieren, die Eliminierung von mehrdeutigen Suchregionen. Zwei Regionen werden dabei als mehrdeutig definiert, falls für ihre Merkmalsvektoren $d_{\mathbf{des}}(\mathbf{o}_a, \mathbf{o}_b) < \delta_{\mathbf{des}}$ gilt. In der Praxis ist diese Vorgehensweise allerdings unbrauchbar, sobald sich mehr wie ein Objekt desselben Typs innerhalb des Bildes befindet, da sonst ein Großteil der gefundenen Regionen eliminiert wird. Die folgenden Versuche sollen aber ein Gefühl für den Schwellwert $\delta_{\mathbf{des}}$ und seine Auswirkungen auf das System vermitteln.

Dazu wurden wieder die Standardsequenzen mit Ground-Truth-Informationen der Box und des Steckers verwendet und die Genauigkeit des Systems bei Verwendung unterschiedlicher Schwellwerte $\delta_{\mathbf{des}}$ untersucht. Die Ergebnisse sind in Abb. 6.19 dargestellt. Ebenfalls wird in den unteren zwei Bildern der Effekt der Eliminierung der verwechselbaren Regionen bei der Box gezeigt. Die gleichen Bilder für den Stecker finden sich in Abb. 4.6.

Der Effekt bei der Box ist erwartungsgemäß kaum erkennbar, da sich nur wenige verwechselbare Strukturen auf dem Objekt befinden. Die Mehrzahl der eliminierten Regionen sind unstabile Regionen entlang der Linien, die keinerlei robusten Informationen für die Objektlokalisation besitzen. Die einzig weiteren eliminierten Regionen sind die dunklen Quadrate. Sie haben aber offensichtlich weder einen positiven noch negativen Effekt auf die Genauigkeit der Lokalisation. Erst wenn der Schwellwert $\delta_{\mathbf{des}}$ deutlich ansteigt und zu viele Regionen eliminiert werden, bricht die Genauigkeit des Systems ein. Offensichtlich werden bei $\delta_{\mathbf{des}} > 0.3$ nicht nur mehrdeutige Informationen ausgesondert, sondern auch ein nicht unwesentlicher Teil der eindeutigen Nutzinformation.

Der gleiche Effekt ist auch bei dem Stecker zu beobachten. Allerdings ist bei diesem, aufgrund seiner extremen Anzahl mehrdeutiger Strukturen in dem Bereich $0.0 \leq \delta_{\mathbf{des}} \leq 0.3$, im Gegensatz zur Box eine Badewannenkurve mit einem Minimum bei $\delta_{\mathbf{des}} = 0.2$ zu beobachten. Offensichtlich führt

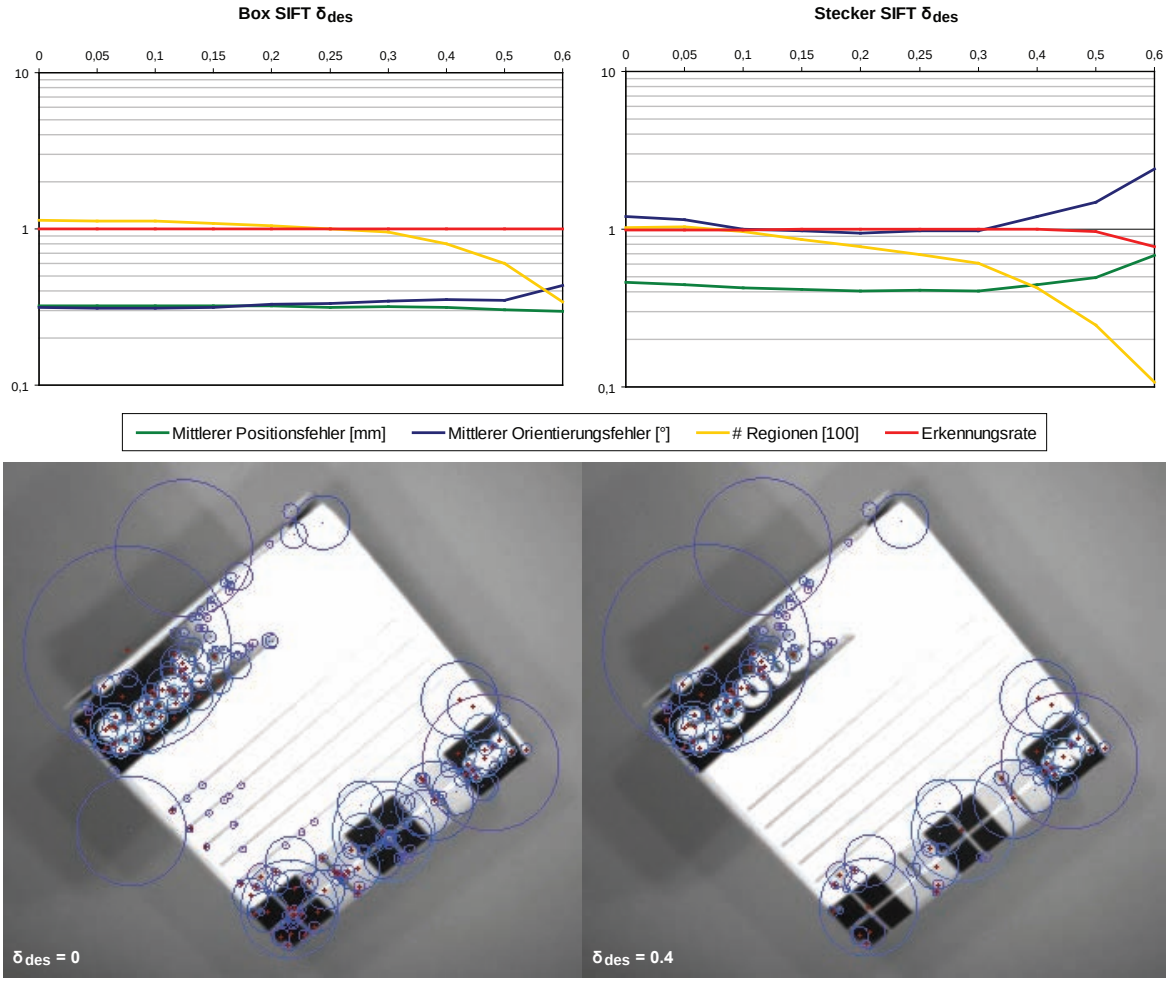

Abbildung 6.19: Auswirkung der Eliminierung von mehrdeutigen Suchregionen auf die Genauigkeit des Gesamtsystems bei Verwendung von SIFT und 3D-3D inkl. Stereo am Beispiel der Box und des Steckers. Zwei Suchregionen gelten als mehrdeutig, falls die Ähnlichkeit $d_{\mathbf{des}}(\mathbf{o}_a, \mathbf{o}_b)$ ihrer zugehörigen Merkmalsvektoren $\mathbf{o}$ unter einem Schwellwert $\delta_{\mathbf{des}}$ liegen. Darunter sind für eine Beispielansicht der Box alle Regionen ($\delta_{\mathbf{des}} = 0.0$) sowie die bis zum Wert $\delta_{\mathbf{des}} = 0.4$ verbleibenden Regionen abgebildet. Gut zu erkennen sind die fehlenden Regionen an den mehrdeutigen schwarzen Quadraten und die gleichartigen (sowie unrobusten) Regionen auf der Linie. Die gleichen Bilder für den Stecker finden sich in Abb. 4.6.

die hohe Anzahl an Fehlkorrespondenzen aufgrund der mehrdeutigen Regionen zu einer schlechteren Performance des Systems. Daher ist es in diesem Fall sinnvoll, die ähnlichsten Regionen zu entfernen. Der Wert von $\delta_{\mathbf{des}} = 0.2$ wird später zur Optimierung der Modelldatenbank als die Grenze zwischen eindeutigen und mehrdeutigen Regionen herangezogen. Die Eliminierung selbst wird in weiteren Experimenten nicht verwendet.

6.4.2 Verwertung der k ähnlichsten Modellregionen

Während bei der Eliminierung mehrdeutiger Regionen Informationen verloren gehen, kann man, wie in Abschnitt 4.5.1 als Alternative beschrieben, die Mehrdeutigkeit explizit berücksichtigen, indem man einer Suchregion $\mathcal{R}_s$ anstatt nur der nächsten Modellregion $\mathcal{R}_1$ in der Datenbank die nächsten

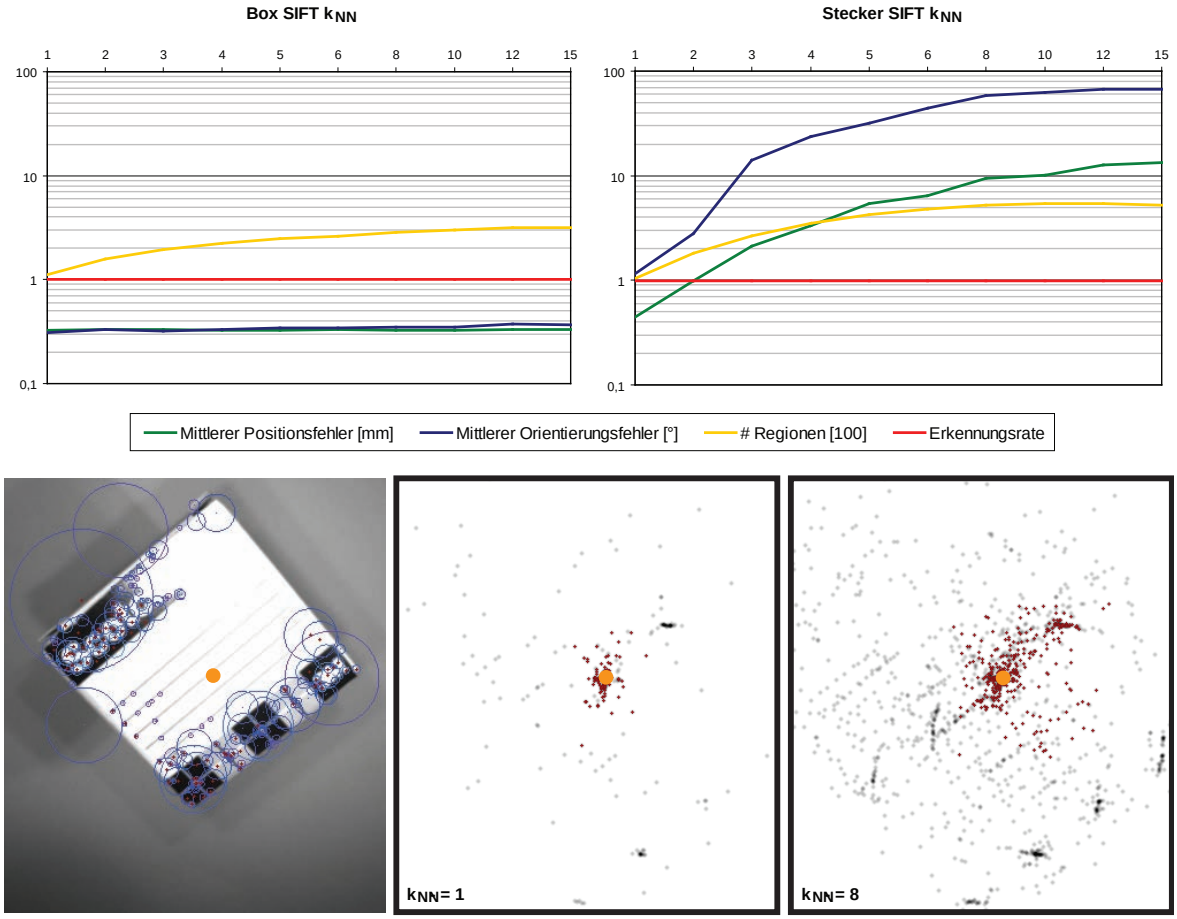

Abbildung 6.20: Verhalten des Systems bei Verwendung der k nächsten Modellregionen bzw. Nachbarn während der Korrespondenzsuche am Beispiel der Box und des Steckers und Benutzung von SIFT sowie 3D-3D inkl. Stereo. Die Auswirkungen auf die lokalen Lagen ist für die unten links abgebildeten Suchregionen für $k = 1$ und $k = 8$ in den rechts davon angeordneten Bildern zu sehen. Gut zu erkennen sind rechts die erhöhte Anzahl an Fehlkorrespondenzen und die Bildung von Nebenmaxima. Während dies bei der Box nur geringfügige Auswirkung hat, führt dies bei dem (weniger gut visualisierbaren) Stecker aufgrund der vielen ähnlichen Strukturen zu einem unrobusten System. Dies ist vor allem auf den Clusterschritt zurückzuführen, der häufiger die lokalen Lagen der Nebenmaxima gruppiert und an die globale Lageauswertung zurückliefert. Der orange Punkt befindet sich in allen Bildern an derselben Stelle und markiert die Ground-Truth-Position des Objekt-Koordinatensystems.

k Modellregionen $\mathcal{R}_i$, $i = \{1,\ldots,k\}$ zuordnet. Man erhält dadurch für jede Suchregionen k Korrespondenzen, deren Deskriptorabweichung $d_i = d_{\mathbf{des}}(\mathbf{o}_s, \mathbf{o}_i)$ als Gewicht für den Clusteralgorithmus (siehe Gl. 4.17) und das Qualitätsmaß (siehe Gl. 4.24 bzw. 4.24) jedes RANSAC-Zykluses der globalen Lageermittlung herangezogen wird. Das Gewicht jeder Korrespondenz ergibt sich dabei zu $w_i = (d_i + \varepsilon)^{-1}$ und wird anschließend noch über alle Korrespondenzen einer Suchregion normiert, so dass $\sum_{i=1}^{k} w_i = 1$ gilt. ε ist ein sehr kleiner Wert zur Vermeidung einer Division durch 0. Die Gewichtung hat zur Aufgabe, dass ähnlichere Korrespondenzen näherer Nachbarn bzw. die daraus abgeleiteten Informationen in der weiterführenden Verarbeitung stärker berücksichtigt werden als die weniger übereinstimmende Korrespondenzen entfernterer Nachbarn.

Abb. 6.20 zeigt das Verhalten des Gesamtsystems bei Verwendung der k nächsten Nachbarn wäh-

rend der Korrespondenzzuordnung bei Verwendung von SIFT und 3D-3D inkl. Stereo anhand der Ground-Truth-Sequenzen für die Box und den Stecker. Normalerweise würde man einen linearen Anstieg der Anzahl der Korrespondenzen (gelb) in Abhängigkeit von k erwarten. Es werden allerdings weiterhin alle Korrespondenzen mit einer zu großen Deskriptorabweichung $d_i > \tau_{\mathbf{des}} = 0.7$ entfernt, so dass sich mit zunehmendem k ein asymptotischer Verlauf einstellt.

Wie schon bei der Eliminierung lässt sich bei der Box kein großer Effekt erkennen, da die Verwechlungsgefahr gering ist und der nächste Nachbar bzw. Modellregion eine deutlich höhere Gewichtung bekommt als die restlichen Nachbarn. Die Auswertung des Steckers wird aufgrund der vielen mehrdeutigen Strukturen bzw. periodischer Elemente deutlich stärker beeinflusst. Allerdings ist leicht ersichtlich, dass die Auswirkungen deutlich negativ zu bewerten sind und die Genauigkeit des Systems schon bei $k = 2$ deutlich abnimmt. Bei höheren k funktioniert das System schlichtweg nicht mehr robust und es kommt zu häufigen Verwechslungen des Clusteralgorithmusses, der dann ein Nebenmaxima dem eigentlichen Clusterzentrum vorzieht. Diese Nebenmaxima sind in Abb. 6.20 unten rechts für $k = 8$ für die (besser visualisierbare) Box dargestellt.

Der Mehrgewinn an Information durch die Auswertung der k nächsten Modellregionen wird von der ebenfalls stark erhöhten Anzahl an Fehlkorrespondenzen dominiert und kann mit den in dieser Arbeit entwickelten Algorithmen ohne weitere theoretische Modellierung nicht genützt werden. Dieser Ansatz wird daher verworfen.

6.4.3 Optimierung der Modelldatenbank

Anstatt Bewertungen der Suchregionen zur Laufzeit vorzunehmen, können auch die Modellregionen während des Trainingsschritts bewertet werden. Hierbei hat man mehrere Möglichkeiten, da man einen Roboter zur Generierung von definierten Ansichten mit Ground-Truth-Informationen und genügend Zeit zur Verfügung hat. Damit lassen sich nach Abschnitt 4.5.2 die einzelnen Modellregionen in *gestörte*, *schwache* und *robuste* Beschreibungen klassifizieren und damit die Modelldatenbank optimieren. Betrachtet man weiterhin die zugrunde liegenden Objektstrukturen, lassen sich praktische Beispiele für robust detektierbare Textur- bzw. Objektmerkmale angeben und damit die theoretischen Überlegungen in Abschnitt 3.2 untermauern.

Für jede trainierte Modellansicht wird mittels des Roboters eine Evaluierungs-Sequenz von 100 Bildern mit bekanntem Blickwinkel aufgenommen und damit die zugehörigen Modellregionen evaluiert. Es werden für jede Region zwei Größen bestimmt: a gibt die Anzahl zugeordneter Suchregionen bzw. damit entstandener Korrespondenzen an, b die Anzahl, bei der die zugehörigen lokalen Lagen sich innerhalb des Ground-Truth-Clusters befanden.

Ist $a < \tau_a$ handelt es sich um eine *gestörte* Modellregion, die keinerlei Information zur Auswertung beiträgt, da sie sehr selten überhaupt einer Suchregion zugeordnet wird. Diese Regionen werden meist an zufällig entstandenen Objektstrukturen innerhalb der Modellansicht detektiert, die meist durch ein Zusammenspiel von Textur und Glanzpunkt entstehen und bei kleinen Blickwinkelvariationen sofort wieder verschwinden. Da sie während der Korrespondenzzuordnung nur Rechenzeit verschwenden, werden sie aus der Modelldatenbank entfernt.

Gilt dagegen $a \geq \tau_a$, ist es interessant, das Verhältnis $r = \frac{b}{a}$ von korrekten zu allen Korrespondenzen zu betrachten. r ist in dem Intervall $[0,1]$, wobei $r = 0$ bedeutet, dass während der Validierungssequenz nur Fehlkorrespondenzen mit der entsprechenden Modellregion entstanden sind, bei $r = 1$ dagegen nur korrekte Korrespondenzen. Ist $r < \tau_r$ wird die Region als *schwach* bezeichnet, d. h. es entstehen viel häufiger Fehl- wie korrekte Korrespondenzen. Sie hat nur einen geringen positiven, aber oftmals negativen Einfluss auf die Lokalisation. Schwache Regionen werden allerdings nicht aus der Datenbank gelöscht, sondern nur markiert und mit einem Gewicht 0 versehen. Sie können also weiterhin zur Korrespondenzfindung herangezogen werden, allerdings werden die lokalen Lagen und damit die zugehörigen Suchregionen verworfen. Dies lässt sich daher auch als eine Bewertung

der Suchregionen über die NN-Klassifikation des Matchers *nach* der Zuordnung verstehen. Schwache Regionen werden meist an mehrdeutigen bzw. nicht eindeutig kovariant konstruierbaren Objektstrukturen oder an, von der Form häufig auftretenden Glanzpunkten detektiert. Der in Abschnitt 4.5.2 beschriebene Ansatz zur Benutzung multipler Korrespondenzen bei mehrdeutigen Strukturen über eine kNN-Suche wird aufgrund der Ergebnisse im vorangegangenen Abschnitt 6.4.2 verworfen.

Die verbleibenden Modellregionen werden als *robust* eingestuft und verbleiben ohne Modifikation in der Modelldatenbank. Optional könnte man sie mittels des Verhältnisses r gewichten, allerdings ist die Korrelation zwischen r und der Nützlichkeit der Informationen während der globalen Lageauswertung gering. Dafür ist nämlich die exakte Lage der Regionenzentren der besten Korrespondenzen entscheidend und dies ändert sich je nach Blickwinkel doch erheblich.

Für die Validierung der Modellregionen wird $\tau_a = 3$ angenommen, d. h. es müssen mind. in 3% der verwendeten 100 Evaluierungsansichten eine Korrespondenz gefunden werden, damit die Regionen nicht als gestört klassifiziert werden. Für τ_r werden verschiedene Einstellungen im Bereich $[0,1]$ untersucht. Dabei wird das Modell optimiert und mit der üblichen Ground-Truth-Sequenz am Beispiel der Box (SIFT und MSER) und des Steckers (nur SIFT) unter Verwendung von 3D-3D inkl. Stereo die Auswirkungen untersucht. Wichtig dabei ist, dass es zwischen der Evaluierungs- und der üblichen Ground-Truth-Sequenz keine übereinstimmenden Blickwinkel gab, um keinen Overfitting-Effekt zu erzeugen. Die Ergebnisse sind in Abb. 6.21 zu sehen.

In den Diagrammen ist ebenfalls der Anteil schwacher Regionen zur Gesamtzahl der Regionen in der Datenbank angegeben. Mit zunehmenden Schwellwert τ_r erhöht sich deren Anteil, so dass während der Lokalisation immer weniger Regionen ausgewertet werden. Beim Stecker ist der Anteil mit ca. 90% bei $\tau_r = 0.7$ deutlich größer als bei der Box mit ca. 50%. Dies lässt sich auf die, fast nur aus Mehrdeutigkeiten bestehenden Objektstrukturen des Steckers zurückführen. Allerdings ist der Einfluss auf die Genauigkeit des Systems gering, außer es wird wie beim Stecker ab $\tau_r > 0.3$ eine zu hohe Anzahl an Korrespondenzen ausgesondert.

Interessant ist aber das Verhalten der lokalen Lagen zu beobachten. Abb. 6.21 enthält daher unter den Diagrammen deren Verteilung für die Box und den Stecker mit schwachen Regionen ($\tau_r = 0.0$) und dem Extremfall ($\tau_r = 0.7$) ohne schwache Regionen. Bei der Box werden vor allem die Fehlkorrespondenzen unterdrückt und es verbleibt ein nahezu sauber erkennbares Clusters. Dieses verliert auch fast keine korrekten Korrespondenzen, wie es die Anzahl der geclusterten Regionen in dem Diagramm zeigt. Die Anzahl bleibt nahezu konstant, obwohl bei $\tau_r = 0.7$ ca. 50% der Modellregionen als schwach markiert sind. Der Effekt ist bei dem Stecker zwar derselbe, allerdings werden nicht nur die Fehlkorrespondenzen unterdrückt, sondern leider auch eine bedeutende Anzahl an korrekten Korrespondenzen. Das Cluster wird daher immer kleiner und die Genauigkeit des Systems sinkt. In diesem Fall kann aufgrund der Validierung der Modellregionen nicht überzeugend zwischen schwachen und robusten Regionen unterschieden werden, dies ist bei dem Stecker stark von dem Blickwinkel abhängig. Da sich der negative Effekt bis $\tau_r = 0.3$ in Grenzen hält, wird diese Einstellung im Weiteren verwendet.

Sehr interessant ist die Betrachtung der Validierung von einzelnen Regionen, um ein Gefühl für robuste und ungeeignete Objektstrukturen zu bekommen. Zwei Dinge zeichnen eine robuste Struktur aus, einerseits die Eignung für eine stabile kovariante Konstruktion der Region und zweitens ihre Eindeutigkeit. Detektoren nutzen je nach Verfahren zwar aus ihrer Sicht nur robuste Intensitätsstrukturen aus, allerdings können diese nur temporär aus einem Zusammenspiel von Beleuchtung und Objektstruktur bzw. -textur entstanden sein.

Abb. 6.22 zeigt in der obersten Zeile A eine Auswahl von jeweils 2 gestörten, aus der Datenbank entfernten SIFT- (grün) und MSER-Modellregionen (rot) der Box. Insgesamt wurden auf der Box mit dem Schwellwert $\tau_a = 3$ 4 von 229 SIFT-Regionen auf der Box als gestört identifiziert und 6 von 90 MSER-Regionen. Auf dem Stecker sind es 9 von 283 SIFT-Regionen. Gut zu erkennen ist, dass die zugrundeliegende Struktur in jedem Fall durch einen Beleuchtungsspot gestört ist. Am besten lässt

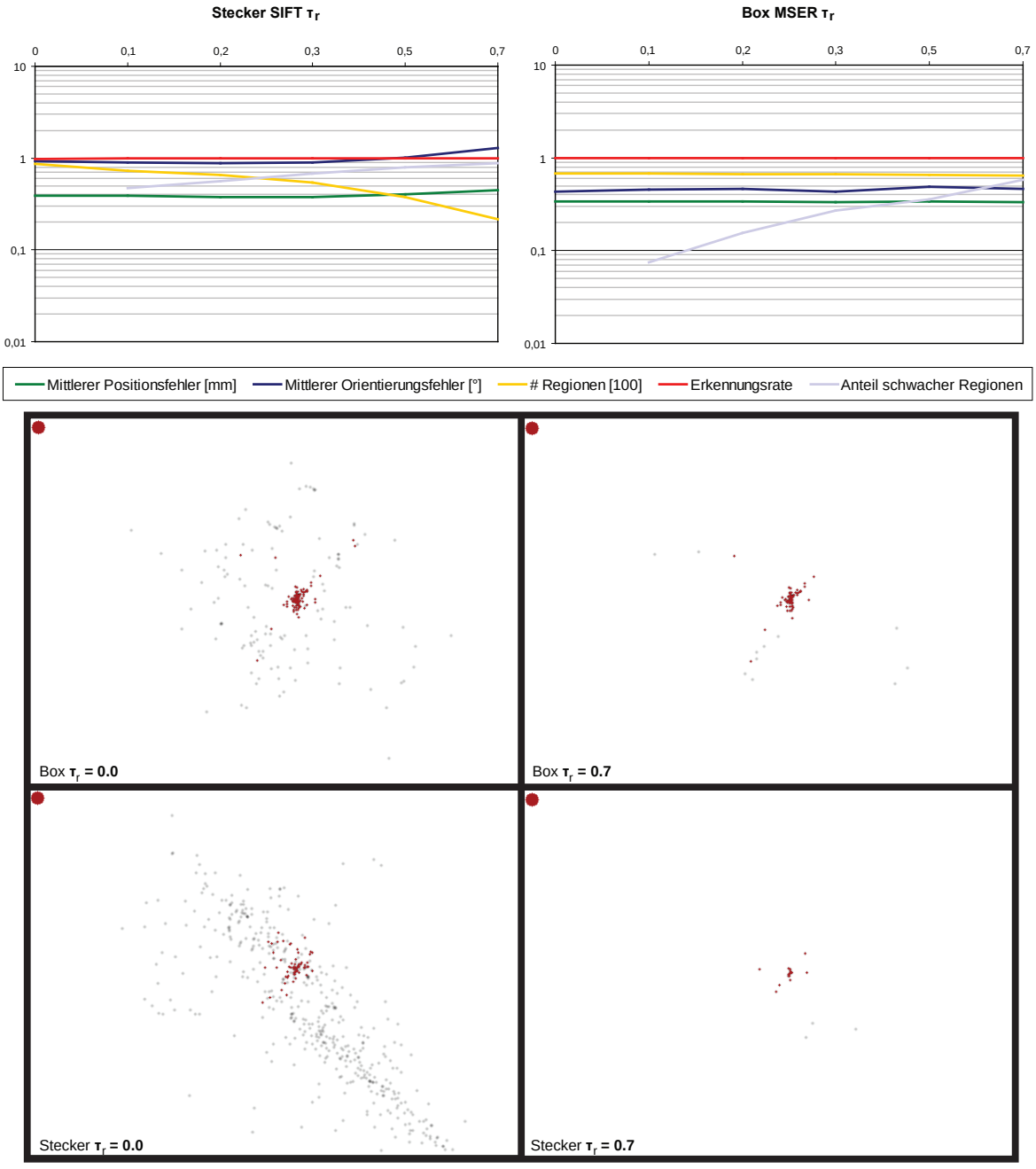

Abbildung 6.21: Verhalten des Systems bei Verwendung optimierter Modelldatenbanken am Beispiel der Box und des Steckers unter Verwendung von MSER bzw. SIFT und 3D-3D inkl. Stereo. Korrespondenzen schwacher Modellregionen mit einem Verhältnis $r < \tau_r$ werden für die Lokalisation nicht berücksichtigt. In grau ist der Anteil schwacher Regionen in der Datenbank angegeben, in gelb die mittlere Anzahl der Regionen innerhalb des gefundenen Clusters. Während der Einfluss auf die Genauigkeit gering ist, ist der Effekt bei der Verteilung der lokalen Lagen groß. Die Fehlkorrespondenzen werden insbesondere bei der Box nahezu vollständig unterdrückt, bei nur einer geringen Verkleinerung des Clusters (vgl. jeweils linkes und rechtes oberes Bild). Bei dem Stecker gelingt dies aufgrund der periodischen, leicht verwechselbaren Strukturen nicht so sauber, hier wird auch eine große Zahl der korrekten Korrespondenzen innerhalb des Clusters verworfen (vgl. unteres Bildpaar).

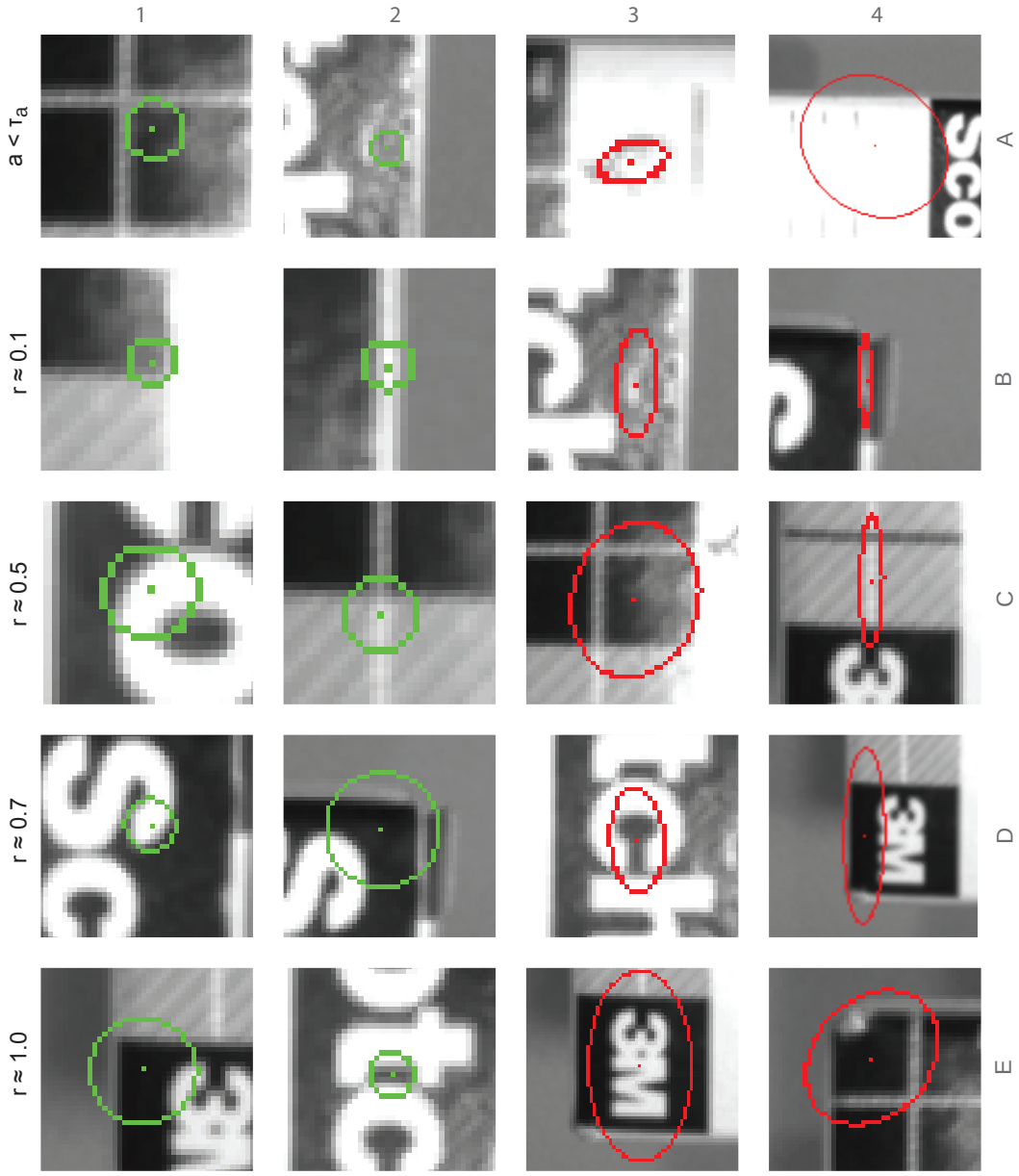

Abbildung 6.22: Beispiele der Validierung von SIFT- (grün) und MSER-Modellregionen (rot) der Box. In der obersten Zeile A sind Beispiele für, meist durch Glanzpunkte gestörte Regionen abgebildet, die aus der Modelldatenbank entfernt wurden. Darunter sind von oben nach unten immer robustere Regionen dargestellt, wobei die Robustheit durch das Verhältnis $r = \frac{b}{a}$ von korrekten zu allen Zuordnungen bestimmt wurde.

sich dies an der zweiten Region von links (als A2 bezeichnet) erkennen, wo ein lokaler, dunkler Blob aufgrund der Rauhigkeit der Boxoberfläche zwischen zwei Glanzpunkten entstanden ist. Diese Situation entsteht natürlich nur für exakt den Blickwinkel, der für das Training der Ansicht genutzt wurde. Die Region ist daher für andere Blickwinkel unbrauchbar und daher ein Verbleib in der Datenbank sinnlos.

Weiterhin enthält die Abb. 6.22 Beispiele für Regionen mit unterschiedlichem Verhältnis r, angefangen in der zweiten Zeile B mit sehr schwachen Regionen ($r \approx 0.1$) bis in der letzten Zeile E mit

den robustesten Regionen ($r \approx 1.0$) der Box. Rein visuell lässt sich schon erkennen, desto schwächer eine Region ist, desto stärker ist die Überlagerung der Objektstruktur mit der Beleuchtung, bzw. desto schwächer sind die Kontraste. Die Region lässt sich daher nicht ausreichend stabil konstruieren, wie am besten bei Region B2 ersichtlich. Ähnlich wie das Blendenproblem beim optischen Fluss lässt sich der Freiheitsgrad entlang der Kante nicht robust bestimmen, sondern seine Festlegung wird von lokalen, je nach Blickwinkel variierenden Beleuchtungsverhältnissen dominiert.

Die Bedeutung der Eindeutigkeit der Objektstruktur lässt sich am besten an der MSER-Region E4 zeigen, deren zugrundeliegende Objekttextur eine der 8 schwarzen Quadrate der Boxoberfläche ist. Diese Quadrate sind aufgrund ihres hohen Kontrastes für MSER sehr robust detektierbar, allerdings sind die Regionen weder eindeutig, noch lässt sich der rotative Freiheitsgrad im normalisierten Patch eindeutig bestimmen, da die Quadrate eine 4-Symmetrie um ihren Schwerpunkt aufweisen. Eine Ausnahme ist eben dieses Quadrat in der Ecke, welches eine Delle aufweist und sich daher alle Mehrdeutigkeiten auflösen lassen. Die dazugehörige Region wird im Gegensatz zu den Regionen der anderen Quadrate zu einer sehr robusten, für die Lokalisation bedeutenden Region.

Wie sich in den Zeilen D und E gut erkennen lässt, ist Schrift als robuste Struktur aufgrund des hohen Kontrastes sehr geeignet. Allerdings sind Einzelbuchstaben bei längeren Schriftzügen nicht sehr eindeutig, da sie öfters vorkommen. Eine Beschreibung über mehrere Buchstaben ist da deutlich besser, wie sich an der besten Region E3 der Box zeigt. Sie wird in 70 von 100 Evaluierungsansichten korrekt zugeordnet und nicht ein einziges mal falsch. Es ist daher bei mehrdeutigen Objektstrukturen und insbesondere bei Schriften wichtig, den Detektor so einzustellen, dass die zurückgelieferte Region einen größeren Bereich des Objekts abdeckt als nur die zur Konstruktion notwendige Struktur. Diese Problematik wird in Abschnitt 3 ausführlich diskutiert.

Anhand der Einzelauswertungen zeigt sich auch nochmals, wieso SIFT auf industriellen Bauteilen meist besser funktioniert als MSER. SIFT nutzt sowohl Blobs als auch Ecken zur Konstruktion und kann daher eine deutlich größere Vielfalt an Strukturen auswerten als MSER, die auf homogene Flächen beschränkt sind. Diese sind zwar auf der Box durch die bedruckte Oberfläche und insbesondere die Textur gegeben, auf industriellen Bauteilen mit komplexeren geometrischen Strukturen allerdings meist nicht.

Ein interessanter Aspekt, auf den in dieser Arbeit leider nicht mehr eingegangen werden kann, wäre aufgrund dieser Ergebnisse bzw. Bewertung von Regionen und deren zugrundeliegenden Bildinformationen einen überwachten Lernvorgang aufzubauen und damit einen Klassifikator zu trainieren, der ohne eine Validierung der Modellregionen deren Güte schätzen kann. Damit wäre man während des Trainings für eine Validierung nicht mehr auf einen Roboter angewiesen und könnte das System flexibler einsetzen.

6.5 Analyse des besten Systems

In den vorangegangenen Abschnitten wurde das beste System, SIFT mit der globalen 3D-3D-Auswertung inkl. der Verwendung von Stereo und die optimierten Parametereinstellungen ermittelt. Dieses System soll nun bzgl. Laufzeit, Genauigkeit und Robustheit abschließend an den 6 in Abb. 6.1 dargestellten Objekten untersucht werden. Während in den vorangegangenen Abschnitten immer optimal aufgenommene Sequenzen verwendet wurden, liegt nun der Fokus auf Verdeckungen, variierendem Hintergrund, Beleuchtungsänderungen, starken Blickwinkelunterschieden, multiplen gleichartige Objekten und komplexen Szenen.

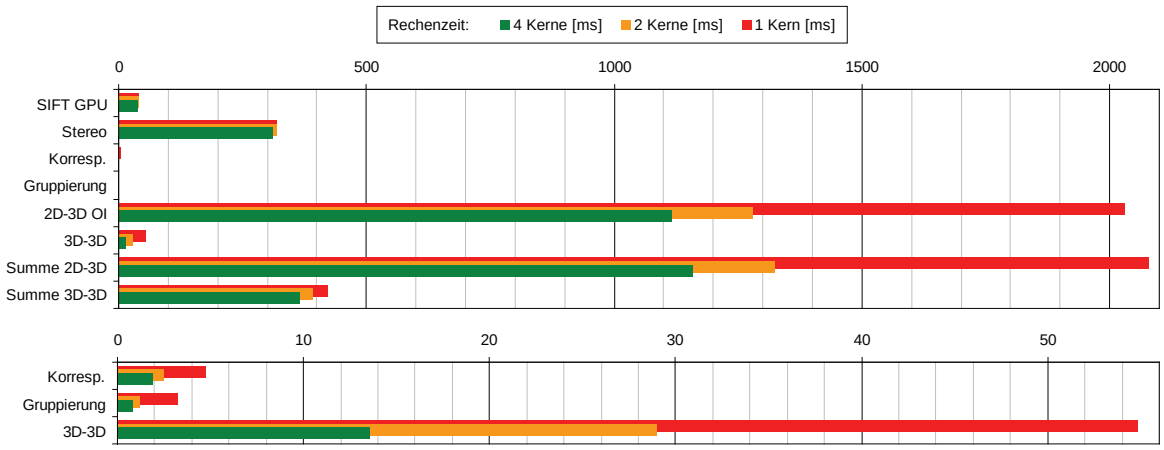

Abbildung 6.23: Laufzeit der Komponenten des Gesamtsystems, sowie die Gesamtzeit für die zwei alternativen Auswerteverfahren 2D-3D bzw. 3D-3D inkl. Stereo-Tiefenrekonstruktion auf einem Intel Core 2 Quad 3 GHz mit 4 Kernen. Zusätzlich wird der Parallelisierungseffekt durch Verwendung von 1, 2 bzw. 4 Threads (und damit auch Kernen) dargestellt. Die 3 schnellsten Komponenten sind unten nochmals auf einer feineren Skala angegeben.

6.5.1 Geschwindigkeit

Zur Messung der Laufzeit wird die übliche Ground-Truth-Sequenz der Box mit dem optimierten System verwendet. In der Datenbank befinden sich $n_k = 363$ Modellregionen aus einer eintrainierten Ansicht. Die in Abb. 6.23 angegebenen Zeiten ergeben sich aus dem Mittelwert über die 324 Suchansichten der Sequenz auf einem Intel Core 2 Quad 3 GHz mit 4 Kernen. Zur Messung der Parallelisierung wird das System mit einer unterschiedlichen Anzahl $n = \{1,2,4\}$ Threads betrieben, womit das System je nach Anzahl der Threads eine unterschiedliche Anzahl von Prozessorkernen nutzt. Zur Bewertung der Güte der Parallelisierung wird der Speedup $s(n)$, dem Verhältnis zwischen einer rein sequentiellen Ausführung mit $n = 1$ und einer parallelen Abarbeitung mit $n > 1$ betrachtet. Im optimalen Fall beträgt der Speedup $s(n) = n$, normalerweise gilt aber $s(n) < n$ mit einem asymptotischen Verlauf gegen den Zeitbedarf der sequentiellen Code-Abschnitte. Aufgrund von unterschiedlich effizienter Cache-Ausnutzung kann sogar ein superlinearer Verlauf $s(n) > n$ entstehen. Da der Cache meist eine große Bedeutung auf $s(n)$ hat, wird hier darauf hingewiesen, dass sich bei dem Intel Core 2 Quad 3 GHz jeweils 2 Kerne den L2-Cache teilen und daher selten ein linearer Verlauf von $s(n)$ zu beobachten ist.

Bei dem optimierten System wird der SIFT-Detektor und -Deskriptor mittels einer GPU Variante auf einer NVIDIA GTX 280 ausgeführt, deren Implementierung unabhängig von n (d. h. $s(n) = 1$) im Mittel 40 ms bei durchschnittlich $n_l = 264$ Regionen benötigt. Die im Projekt ebenfalls verwendete CPU-Implementierung benötigt im Vergleich ca. 1 s.

Der kommerzielle Stereo-Disparitäts-Algorithmus von Halcon benötigt inkl. der Konvertierung der Disparitäten in Tiefeninformationen ca. 320 ms im seq. Fall. Da nur die wenig rechenintensive Konvertierung parallel implementiert ist, fällt der Speedup mit $s(4) = 1.03$ sehr gering aus.

Als Korrespondenzzuordnung wird die SSE3 optimierte CPU-Variante verwendet, die in Abb. 6.4 ausführlich untersucht worden ist. Sie hat eine Komplexität von $O(n_k n_l)$ und ist inkl. der ebenfalls hier angesiedelten Berechnung der lokalen Lagen vollständig parallelisiert. Dies zeigt sich auch in dem Speedup $s(2) = 1.93$ bzw. $s(4) = 2.52$. Da hier allerdings große Datenmengen bearbeitet werden, bricht der Speedup aller Wahrscheinlichkeit nach aufgrund des geteilten Caches bei $n = 4$ ein.

Der Clusteralgorithmus im Gruppierungsschritt ist ebenfalls vollständig parallelisiert, wie sich an $s(2) = 2.64$ bzw. $s(4) = 3.97$ zeigt. Er verarbeitet kleinere Datenmengen als die Korrespondenzzuordnung und daher ist der Speedup nahezu optimal bzw. bei $n = 2$ sogar superlinear. In diesem Fall kommt es offensichtlich durch die zwei Caches zu weniger Fehlzugriffen und daher zu diesem Effekt.

Bei der globalen Lageauswertung werden zum Vergleich die beiden Alternativen 2D-3D-OI und 3D-3D betrachtet. Aufgrund der iterativen Struktur ist im sequentiellen Fall 2D-3D-OI mit durchschnittlich 2030 ms deutlich langsamer als 3D-3D mit 50.84 ms. Aufgrund des RANSAC sind beide Verfahren vollständig parallelisiert, der Speedup beträgt bei 2D-3D-OI $s(2) = 1.59$ bzw. $s(4) = 1.88$ und bei 3D-3D nahezu optimal $s(2) = 1.82$ bzw. $s(4) = 4.04$. Der Unterschied in der Parallelisierungsgüte lässt sich auf eine Lastverteilungs-Problematik zurückführen. Die RANSAC-Implementierung teilt die Zyklen im Vorfeld statisch auf die zur Verfügung stehenden Threads auf. Während die Zyklen bei 3D-3D aufgrund der analytischen Lösung immer den gleichen Rechenbedarf benötigen, kann diese bei der iterativen Auswertung des 2D-3D-OI je nach Zyklus stark variieren, so dass in ungünstigen Fällen ein Thread eine deutlich längere Rechenzeit benötigt. In diesem Fall wäre eine dynamische Verteilung sinnvoller.

Zuletzt werden noch die Zeiten der zwei möglichen Kombinationen des Gesamtsystems untersucht. Dies sind bei 2D-3D alle essentiellen Komponenten, d. h. SIFT GPU, Korrespondenzzuordnung und Gruppierung sowie die globale 2D-3D-OI-Lagebestimmung und bei 3D-3D zusätzlich die Stereoverarbeitung sowie die globale 3D-3D-Auswertung Lageauswertung. Die Laufzeit dieser Systeme liegt im sequentiellen Fall bei 2078 ms bzw. 423.4 ms und im parallelisierten Fall bei 1158 ms bzw. 365.3 ms. Das System lässt sich ohne Berücksichtigung der Bildaufnahme ohne Stereoverarbeitung mit einem knappem 1 Hz bzw. bei Stereoverarbeitung mit knappen 3 Hz betreiben. Der geringe Speedup bei 3D-3D von $s(4) = 1.16$ hängt mit dem hohen Anteil an sequentiellen Code von 85% zusammen. Dies ist die Regionen- und die Stereo-Tiefenrekonstruktion, auf die in dieser Arbeit kein Fokus gelegt wurde.

6.5.2 Genauigkeit

Zur Betrachtung der Genauigkeit des Systems werden alle 6 Objekte mit den gleichen Einstellungen eingelernt und evaluiert, ohne eine Optimierungen auf objektspezifische Eigenschaften. Dies ist wichtig, um den Anspruch an ein vollautonomes Training und eine universelle Lokalisation zu erfüllen. Die Genauigkeit des Systems hängt sowohl von inhärenten Eigenschaften als auch in entscheidendem Maße von der Eignung des Objektes ab.

Eine untere Grenze lässt sich mit der Grundgenauigkeit des Systems angegeben. Dafür wurde jedes Objekt mit einer Modellansicht eingelernt und unter demselben Blickwinkel 25 weitere Ansichten für die Lokalisation aufgenommen. Der mittlere Fehler der Position und Orientierung hängt in diesem Fall nur von den Störeinflüssen während der Bildaufnahme und den zufälligen Teilen der Algorithmen ab. Die Fehler betragen im schlechtesten Fall 0.022 mm (Handy) bzw. 0.09° (Plastikbauteil). Die Aussagekraft dieser Größen ist allerdings begrenzt, da sich höchstens ein sehr schwacher Zusammenhang zwischen der Grundgenauigkeit einzelner Objekte und der Robustheit bzw. Genauigkeit unter Blickwinkelvariationen finden lässt.

Weit interessanter ist eine Genauigkeitsanalyse mit den, auch schon in den vorangegangenen Abschnitten verwendeten Ground-Truth-Sequenzen mit 324 Ansichten und max. 15° Verkippung, 45° Rotation in der Bildebene und ± 15 mm Translation in allen Achsen. Die Genauigkeit in der Position und Orientierung sowie die Detektionsrate ist für alle Objekte in dem linken, oberen Diagramm in Abb. 6.24 zu sehen.

Die Leiterplatte, die Box, der Stecker und das Handy haben eine Detektionsrate von 1 und werden daher in allen 324 Suchansichten erkannt. Das Plastikbauteil wird dagegen nur in 90% der Fälle wiedergefunden und die Koppeleinheit sogar nur in 65% der Fälle. Allein aus der Detektionsrate lässt

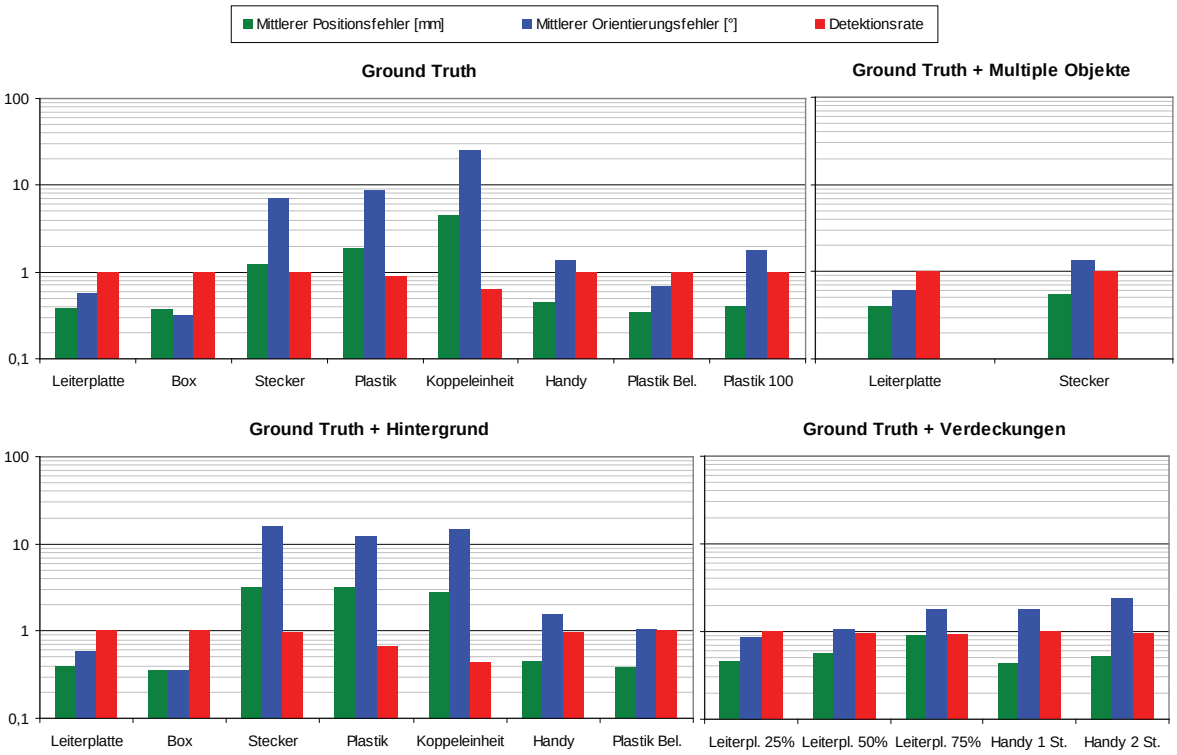

Abbildung 6.24: Positions- und Orientierungsgenauigkeit sowie Detektionsrate des besten Systems für alle in dieser Arbeit untersuchten Objekte. Das Diagramm links oben enthält die Auswertung der Ground-Truth-Sequenz, das Diagramm rechts oben die Auswertung der Ground-Truth-Sequenz bei Verwendung *mehrerer gleicher* Objekte, das Diagramm links unten die Auswertung der Ground-Truth-Sequenz bei Verwendung eines *nicht trainierten, komplexen Hintergrunds* und das Diagramm rechts unten die Auswertung der Ground-Truth-Sequenz bei unterschiedlich *verdeckten* Objekten. Alle Objekte wurden mit 1 Modellansicht eingelernt, bis auf das Plastikbauteil welches zusätzlich mit *100* eingelernten Ansichten evaluiert wurde (Bezeichnung Plastik 100). Ebenfalls wurde das Plastikbauteil zusätzlich mit einer *stationären* anstatt der beweglichen, neben der Kamera montierten Beleuchtung untersucht (Bezeichnung Plastik Bel.).

sich daher schon eindeutig erkennen, welche Objektoberflächen gut und welche schlecht geeignet sind. Die Koppeleinheit besitzt keine echte Textur, sonder nur sehr schwach ausgeprägte und daher kontrastarme Oberflächenmerkmale oder extrem spiegelnde durchsichtige Bereiche. Die Detektoren versagen daher schon bei kleinen Beleuchtungs- oder Blickwinkelvariationen und können nicht in ausreichendem Maße den Modellregionen entsprechende Suchregionen finden. Das Plastikbauteil ist farblich homogen, es lassen sich aber aufgrund der komplexen, feinen 3D-Oberflächenstruktur Regionen an Ecken und Löchern detektieren. Da die Strukturen nur bedingt einen Ebenencharakter aufweisen, ändern sie insbesondere bei Verkippungen aus der Kameraebene hinaus schnell ihr Erscheinungsbild. Als Beleg dafür wurde zum Vergleich das Plastikbauteil nicht nur mit 1 sondern 100 Modellansichten eingelernt, die in 10°-Schritten den gesamten Blickwinkelbereich bis max. 45° Verkippung abdecken. Mit dieser Maßnahme lässt sich ebenfalls eine Detektionsrate von 1 erreichen.

Die Eignung der einzelnen Objekte lässt sich noch viel exakter an der Genauigkeit ablesen. Die Leiterplatte, die Box und das Handy haben den niedrigsten mittleren Positionsfehler von ca. 0.4 mm. Der Fehler setzt sich zusammen aus der Translationsabweichung entlang der Bildebene und der Ab-

weichung entlang der optischen Achse, welche sich durch einen Skalierungsfehler in der Bildebene bemerkbar macht. Der Positionsfehler lässt sich daher auch in Abhängigkeit der in Abschnitt 1.3 im Detail beschriebenen Sensorik interpretieren. Bei dem, während des Trainings verwendeten Abstands des Objekts zur Kamera von 170 mm bedeckt ein Pixel einen Bereich der Objektoberfläche von ca. 0.23 mm Kantenlänge. Der Positionsfehler von 0.4 mm entspricht daher bei reiner Translation in der Bildebene ca. 1.7 pix und bei reiner Skalierung ca. 0.6 pix, falls man das Objekt mit $\frac{1}{3} = 256$ pix der Bildgröße annimmt.

Der Orientierungsfehler beträgt bei der Leiterplatte und der Box 0.6° bzw. 0.3°, das Handy ist mit 1.3° etwas schlechter. Da man bei einer Verkippung nur die, um den Cosinus des Verkippungswinkels gestauchte Objektoberfläche in der Bildebene beobachtet, entsprechen die 0.3° der Box ca. 0.3 pix bei 15° Verkippung und ansonsten gleichen Annahmen wie bei der Interpretation der Positionsgenauigkeit.

Die Gründe für das schlechte Abschneiden des Plastikbauteils und der Koppeleinheit bei der Genauigkeit wurden schon in der Diskussion bei der Detektionsrate genannt. Aber auch hier zeigt sich wieder, dass bei 100 eingelernten Modellansichten die Genauigkeit des Plastikbauteils sehr stark verbessert wird und dann vergleichbar mit der Genauigkeit des Handy's (allerdings mit 1 Modellansicht) ist.

Der Stecker ist ein diskussionswürdiger Sonderfall. Obwohl man anhand der Detektionsrate ein besseres Ergebnis erwarten würde, ist die Genauigkeit insbesondere der Orientierung sehr schlecht. Dies liegt aber an den periodischen Strukturen und einer fast symmetrischen Oberfläche. Es kommt daher in seltenen Fällen zu Verwechslungen während des Gruppierungsschritts, wo anstatt des Hauptclusters ein Nebencluster gefunden wird. Die Nebencluster entstehen oftmals durch symmetriebedingte Verwechslungen während der Korrespondenzzuordnung und führen daher zu, um 180° falsch orientierte Lagehypothesen. Dieser Effekt wirkt sich daher stark auf die Mittelung des Fehlers aus und führt zu dem schlechten Ergebnis. Der Effekt lässt sich gut in Abb. 6.29 bei dem mittleren Stecker beobachten. Dort wurde sowohl das Haupt- als auch das Nebencluster gefunden und für beide eine globale Lagehypothese ermittelt.

Anhand dieser Ergebnisse lässt sich mit Box, Leiterplatte, Handy, Stecker, Plastikbauteil und Koppeleinheit eine Reihenfolge der Objekte angeben, wie gut sich deren Objektoberflächen für das in dieser Arbeit vorgestellte System eignen. Wie schon in Abschnitt 6.4.3 ermittelt, funktionieren kontrastreiche, eindeutige Texturmerkmale am besten. Daher ist es nicht verwunderlich, dass die Box aufgrund der Schrift am besten funktioniert. Die Leiterplatte ist ebenfalls sehr gut, sie enthält mehr, dafür nicht ganz so kontrastreiche Merkmale. Im Gegensatz zur Leiterplatte spiegelt sie allerdings deutlich stärker. Strukturmerkmale wie Löcher und Ecken sind schwächere Merkmale und brauchen mehr Ansichten, wie anhand des Plastikbauteils gezeigt. Werden die Kontraste wie bei der Koppeleinheit zu schwach, bricht das System ein. Symmetrien und Mehrdeutigkeiten wirken sich ebenfalls negativ auf die Genauigkeit aus.

6.5.3 Robustheit

Neben der Genauigkeit ist der wichtigste Aspekt eines erfolgreichen Objektlokalisationssystems seine Robustheit. Von praktischer Bedeutung ist vor allem die Reaktion des Systems auf unterschiedliche Beleuchtungen, Hintergrundänderungen, Variationen der Blickwinkel, mehrere Objekte des gleichen Typs und teilweise Verdeckungen. Diese Herausforderungen werden in den folgenden Unterabschnitten einzeln diskutiert. Der letzte Unterabschnitt 6.5.3.6 diskutiert dann abschließend das Verhalten des Systems bei komplexen Szenen, bei denen meist alle der oben genannten Herausforderungen zusammen auftreten.

6.5.3.1 Beleuchtung

Bei einem ansichtsbasierten System spielt die Beleuchtung eine große Rolle, da immer nur das Erscheinungsbild, eine Überlagerung aus Beleuchtung und Reflexion der Oberfläche eines Objektes beobachten werden kann. Man muss allerdings zwischen lambertschen und spekularen Effekten unterscheiden. Im ersten Fall hängt die abgestrahlte Lichtmenge in eine bestimmte Richtung nur von der Absorption und der eingefallenen Lichtmenge ab, im zweiten Fall nach dem Reflexionsgesetz noch von Einfallsrichtung der Beleuchtung relativ zu der Oberflächennormale und dem Beobachtungswinkel. In der Praxis bestehen alle Oberflächen aus einer Kombination beider Effekte und werden je nach Gewichtung als *matt* oder *glänzend* bezeichnet.

Matte Oberflächen eignen sich besonders zur Beobachtung von Texturinformationen, da Änderungen des Beobachtungs- oder Beleuchtungswinkel nur zu geringfügigen Änderungen des Erscheinungsbilds führen. In erster Linie wird hier je nach Lichtmenge (bei der verwendeten Sensorik vor allen durch den Abstand zum Objekt beeinflusst) der Kontrast der beobachteten Textur beeinflusst. Bei glänzenden Oberflächen führen dagegen schon geringfügige Änderungen der Szenenanordnungen zu starken Störungen des Erscheinungsbilds. Insbesondere werden Texturinformationen von spekularen Glanzpunkten an den entsprechenden Stellen vollständig überlagert und können nicht zur Lokalisation herangezogen werden. Beide Effekte lassen sich in Abb. 6.25 beobachten. Im hellen Bereich wird das Bild durch den Glanzeffekt vollständig überstrahlt, die dort gefundenen Regionen tragen keinerlei Nutzinformationen. Die restlichen Bereiche sind vorwiegend von lamb. Abstrahlung dominiert und lassen daher je nach Abstand zur Kamera eine unterschiedlich kontrastreiche Beobachtung der Textur zu. Wird der Kontrast allerdings zu gering, können wie an der entferntesten Leiterplatte ersichtlich keine Regionen mehr detektiert werden.

Noch gravierender wirkt sich die Beleuchtung auf Regionen aus, die an 3D-Strukturmerkmalen wie Löcher und Ecken gefunden werden. Hier tritt neben der von der Oberflächenbeschaffenheit abhängige Abstrahlcharakteristik auch noch das Phänomen der Abschattung auf. Der Schattenwurf ändert sich wie die Glanzpunkte ebenfalls schon bei kleinen Änderungen am Szenenaufbau.

Matte Objekte mit Textur, wie z. B. die Box eignen sich daher am besten für das System. Sind die Texturinformationen allerdings großflächig und stark ausgeprägt, wie bei der Leiterplatte, eignen sich auch glänzende Objekte, da die zwangsläufig beobachteten Glanzpunkte nicht die gesamte Information verdecken. Schwieriger wird es bei matten Objekten ohne Textur wie dem Plastikbauteil, wo die Regionen nur noch an Strukturmerkmalen gefunden werden.

Für solche Objekte gibt es zwei Möglichkeiten. Einerseits die schon gezeigte Variante, möglichst viele Ansichten und damit Erscheinung des Objekts unter verschiedenen Blickwinkeln während des Trainings zu berücksichtigen und andererseits eine relativ zum Objekt stationäre Beleuchtung. Letztere Variante löst zumindest die Abschattungsproblematik und ist beispielhaft in Abb. 6.26 gezeigt.

Die Varianten können in Abb. 6.24 im oberen linken Diagramm verglichen werden. Unter der Bezeichnung Plastik befindet sich die übliche Ground-Truth-Sequenz mit 1 Modellansicht, unter Plastik 100 die gleiche Sequenz mit 100 Modellansichten und unter Plastik Bel. die gleiche Sequenz mit stationärer Beleuchtung und 1 Modellansicht. Mit beiden Varianten steigt die Detektionsrate von 90% auf 100%, ebenfalls erhöht sich die Genauigkeit drastisch. Die Variante mit stationärer Beleuchtung reduziert den Fehler von 2 mm auf 0.35 mm bzw. von 9° auf 0.7° am meisten. Leider ist solch ein Aufbau mit der im Projekt üblicherweise verwendeten Sensorik nicht möglich. Es bleibt deshalb nur die mit Fehlern von 0.4 mm bzw. 1.8° ebenfalls gut funktionierende Variante mit dem Training vieler unterschiedlicher Erscheinungsbilder, allerdings bremst dies das System bei der Korrespondenzfindung stärker aus. Noch deutlicher wird der Unterschied bei der im nächsten Abschnitt diskutierten Verwendung eines anderen Hintergrundes.

Es verbleibt noch die Diskussion von glänzenden Bauteilen ohne Texturinformationen, wie es z. B. bei Metallobjekten wie Schrauben oder ähnlichem der Fall ist. Hier funktioniert eine Generalisie-

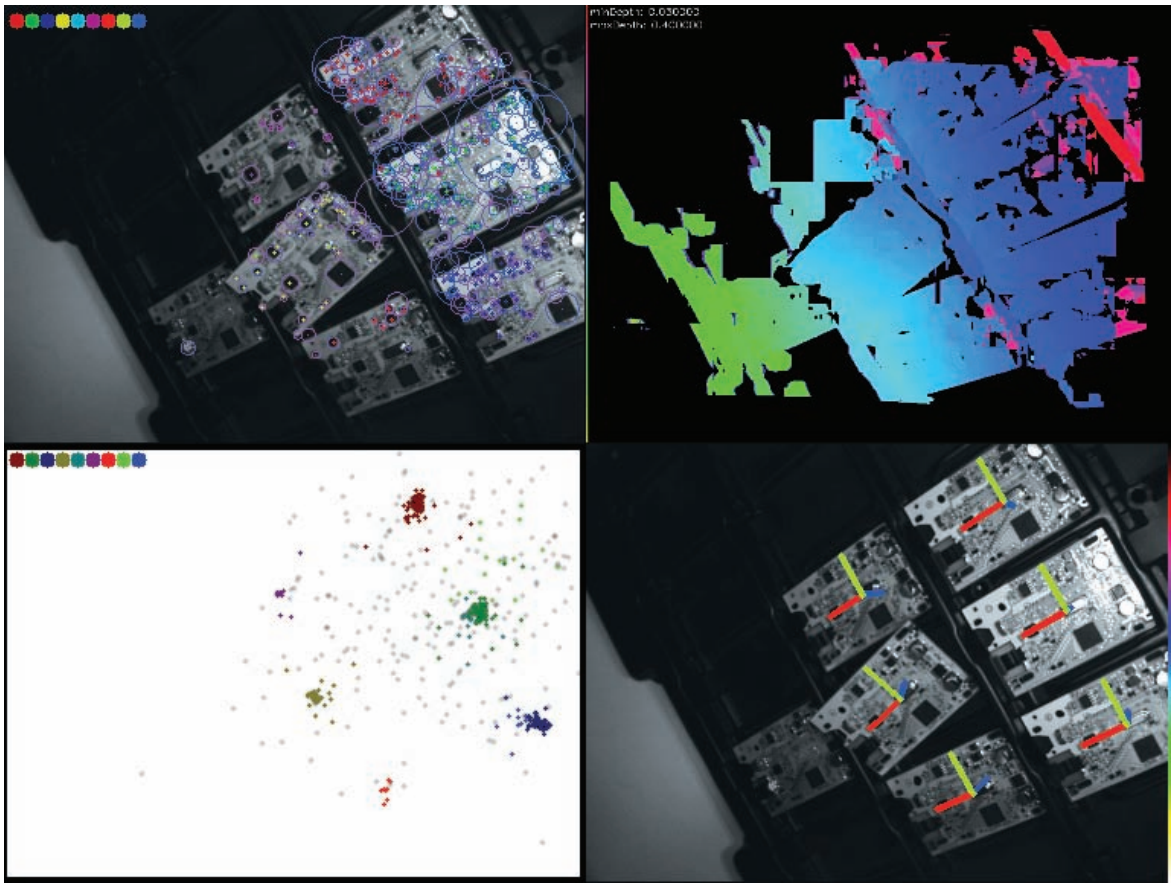

Abbildung 6.25: Visualisierung aller Schritte einer Lokalisation. Links oben ist die Suchansicht mit detektierten Regionen dargestellt, rechts daneben das zugehörige Tiefenbild in Falschfarbendarstellung (von rot mit 30 mm Abstand über blau und grün nach gelb mit 400 mm Abstand), links unten die lokalen Lagen und rechts unten die, aus den Clustern ermittelten globalen Lagehypothesen. Es wurden 9 Cluster innerhalb den lokalen Lagen gefunden, ersichtlich an den 9, in den Farben der Cluster dargestellten Kreise. In den gleichen Farben sind darüber die Zentren der zugehörigen Suchregionen markiert, um die, durch die Gruppierung realisierte Segmentierung zu verdeutlichen. Ebenfalls lassen sich bei der Suchansicht die Effekte der Beleuchtung erkennen. Oben links ein durch Glanzpunkte überstrahlter Bereich der Textur, der durch spekulare Reflexion ausgelöst wird, im restlichen Bild mit dem Abstand abnehmende Helligkeit. Letzteres führt zu immer geringeren Kontrasten, bis der Detektor keine Regionen mehr finden kann.

rung eines eintrainierten Erscheinungsbilds über einen bestimmten Blickwinkelbereich nicht mehr. Man könnte sich theoretisch zwar behelfen, indem der gesamte benötige Bereich fein diskretisiert eintrainiert wird. Da es sich dabei aber um bis zu 6 zu beachtende DoF's handelt, würde dies jede Datenbank sprengen.

6.5.3.2 Variabler Hintergrund

Eingelernt werden alle Objekte mit einem homogenen Hintergrund so wie es in Abb. 4.5 zu sehen ist. Die Evaluierung der normalen Ground-Truth-Sequenz erfolgt ebenfalls auf einem homogenen Hintergrund. In der Praxis ändert sich der Hintergrund allerdings ständig und variiert insbesondere im

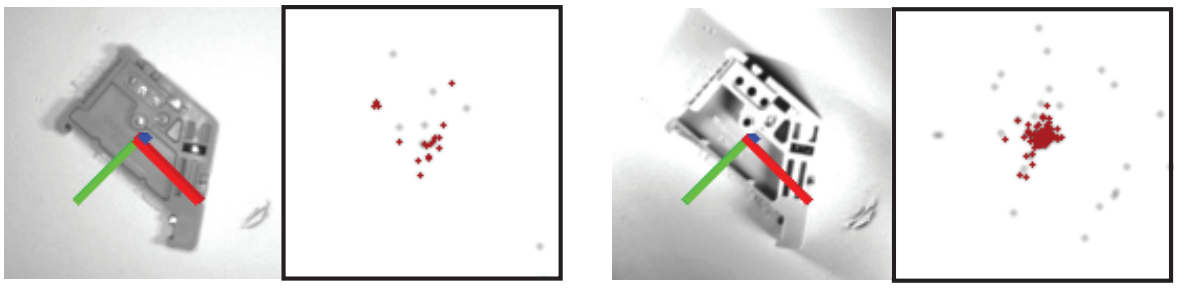

Abbildung 6.26: Verhalten des Systems bei unterschiedlichen Beleuchtungsarten, links ist das Plastikbauteil mit der am Roboter befestigten bewegten Beleuchtung abgebildet, rechts dasselbe Objekt mit einer stationären Beleuchtung. Der stationäre Fall hat den Vorteil, dass sich die Beleuchtung relativ zum Objekt und damit auch die Abschattung nicht ändert. Das Erscheinungsbild bleibt damit im Vergleich zum eingelernten Zustand ähnlicher wie bei der bewegten Beleuchtung und erlaubt damit ein wesentlich bessere Auswertung, wie sich leicht an der Verteilung der lokalen Lagen erkennen lässt.

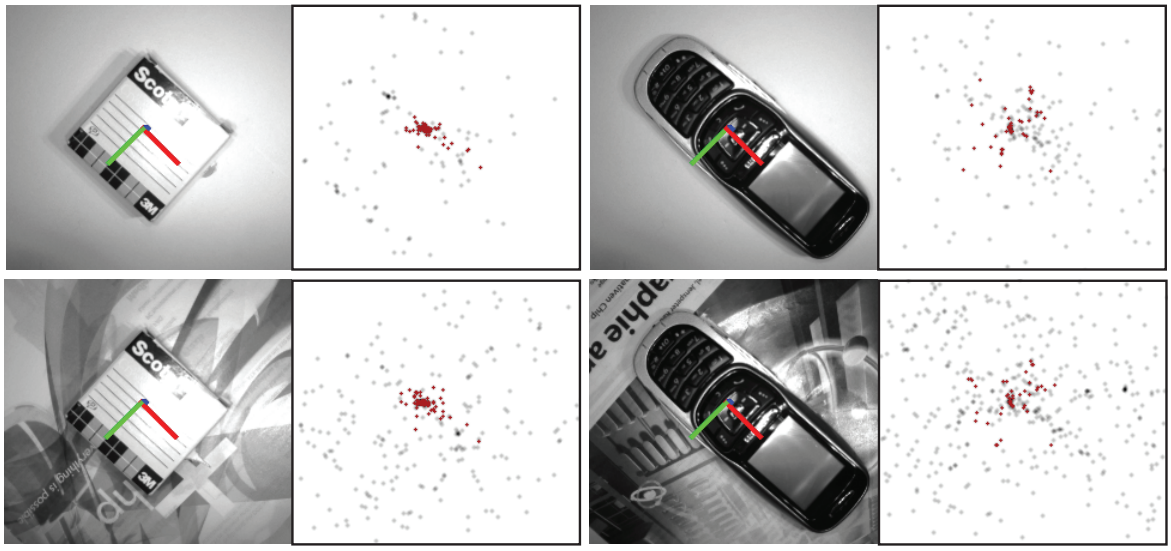

Abbildung 6.27: Vergleich des Systems bei homogenem (oben) und komplexem Hintergrund (unten) am Beispiel der Box und des Handys. Deutlich zu erkennen ist die erhöhte Anzahl an Fehlkorrespondenzen durch die Detektion von Regionen im Hintergrund. Ebenfalls werden es (geringfügig) weniger korrekte Korrespondenzen innerhalb der Cluster, da Regionen an der Objektgrenze teilweise durch den Hintergrund gestört sind und daher gar nicht oder wesentlich schwerer den entsprechenden Modellregionen zugeordnet werden können.

Vergleich zum Training sehr. Es ist daher unbedingt notwendig für ein flexibles Objektlokalisationssystems mit dieser Herausforderung umzugehen. Es wird daher für alle Objekte die Ground-Truth-Sequenz nochmals mit einem komplexen Hintergrund evaluiert. Beispielansichten der Box und des Handys finden sich in Abb. 6.27.

Zwei Dinge wirken sich auf die Leistung des Systems aus. Erstens sind deutlich mehr Fehlkorrespondenzen zu erwarten, da mehr irrelevante Suchregionen auf dem Hintergrund zugeordnet werden. Davon wird zwar nur ein Bruchteil fälschlicherweise den Modellregionen zugeordnet, bei aber bis zu

1000 im komplexen Hintergrund gefunden Regionen eine nicht zu vernachlässigende Menge. Zweitens werden Regionen an den Objektgrenzen gestört. Entweder nur die Beschreibung des Regioneninhalts, was zu einer niedrigeren Zuordnungs-Wahrscheinlichkeit führt, oder gar durch eine veränderte Form der Region, die dann keine korrekte Lagerekonstruktion zulässt. In beiden Fällen führt dies dazu, dass das Cluster im Vergleich zu dem Cluster bei einem homogenen Hintergrund eine geringere Anzahl korrekter Lagen enthält und daher schwieriger zu finden ist.

Ein komplexer Hintergrund wirkt sich daher aufgrund der schwächeren Cluster und erhöhten Anzahl an Fehlkorrespondenzen sowohl negativ auf die Detektionsrate aus, als auch auf die Genauigkeit, da ebenfalls mehr Fehlkorrespondenzen innerhalb eines gefundenen Clusters zu erwarten sind. Diese Effekte kann man bei einem Vergleich der beiden untereinander befindlichen linken Diagramme in Abb. 6.24 beobachten.

Die robusten Bauteile, wie die Box, die Leiterplatte, das Handy und das Plastikbauteil bei stationärer Beleuchtung mit ausgeprägten und genügend großen Clustern haben nur eine geringfügig schlechtere Genauigkeit wie bei der normalen Ground-Truth-Sequenz mit homogenen Hintergrund. Bei den schwierigen Objekten, wie dem Plastikbauteil (mit normaler Beleuchtung) oder der Koppeleinheit mit schwachen, schlecht ausgeprägten Clustern bricht die Detektionsrate dagegen ein. Der Stecker wird ebenfalls schlechter, da sich die eindeutigen Regionen in größeren Maße an den Objektgrenzen befinden und die mehrdeutigen Strukturen im Objektinneren. Es kommt daher tendenziell zu noch mehr Verwechslungen der Orientierung und daher zu einem noch höheren Fehler.

Zusammenfassend lässt sich sagen, dass Hintergrundänderungen nur für Bauteile problematisch sind, auf denen sich nur wenige robuste Regionen detektieren lassen und daher ein schwach ausgeprägtes Cluster besitzen oder Objekte, bei denen aufgrund einer komplexen Silhouette die Mehrzahl der Regionen an der Objektgrenze und nicht innerhalb des Objektes gefunden werden. Ersterem lässt sich z. B. das Plastikbauteil zuordnen, letzterem die Koppeleinheit.

6.5.3.3 Blickwinkelvariationen

Interessant ist das Verhalten des Systems bei Blickwinkeländerungen zu evaluieren, schließlich ist es die Essenz eines Lokalisationssystems von Ansichten mit bekanntem Blickwinkel auf andere Ansichten zu generalisieren, d. h. dort das Objekt wiederzuerkennen und den korrekten Blickwinkel einzuschätzen.

Um die Einflüsse der unterschiedlichen möglichen Blickwinkeländerungen besser bewerten zu können werden sie separat betrachtet. Es handelt sich dabei um eine Rotation und Translation bzgl. jeder Achse des Kamera-Koordinatensystems. Da sich die X- und Y-Achse jeweils parallel zur Bildebene befinden wirken sich dort die Änderungen vergleichbar aus, die optische oder Z-Achse steht dagegen senkrecht zur Bildebene und muss separat betrachtet werden. Es werden daher 4 Sequenzen aufgenommen und jeweils die Fehler zum eingelernten Zustand betrachtet. Die Ergebnisse sind in Abb. 6.28 zu sehen.

In allen Auswertungen lassen sich drei Dinge beobachten. Erstens lässt sich tendenziell eine Zunahme der Fehler mit größerem Abstand von der eingelernten Modellansicht, aufgrund einer zunehmenden Änderung des Erscheinungsbilds durch die Beleuchtung, die Linsenverzerrung und nichtplanare Oberflächen, feststellen. Dem überlagert sind grobe Abweichungen der Tendenz, insbesondere bei Verkippungen, die auf wandernde Glanzpunkte zurückzuführen sind. Als letztes lässt sich ein Rauschen der Kurve beobachten, welches sich durch die zufällige Auswertung der globalen Lageermittlung mittels des RANSAC-Algorithmus begründen lässt.

Als erstes wird die Rotation in der Bildebene, d. h. um die Z-Achse der Kamera diskutiert. Eingelernt wurde eine Modellansicht bei 0° und anschließend alle 5° eine Suchansicht aufgenommen. Da alle Detektoren kovariant gegenüber solchen Rotationen sind, sollten diese Blickwinkeländerungen für das System kein Problem darstellen. Die Leiterplatte und das Handy zeigen sich auch sehr robust

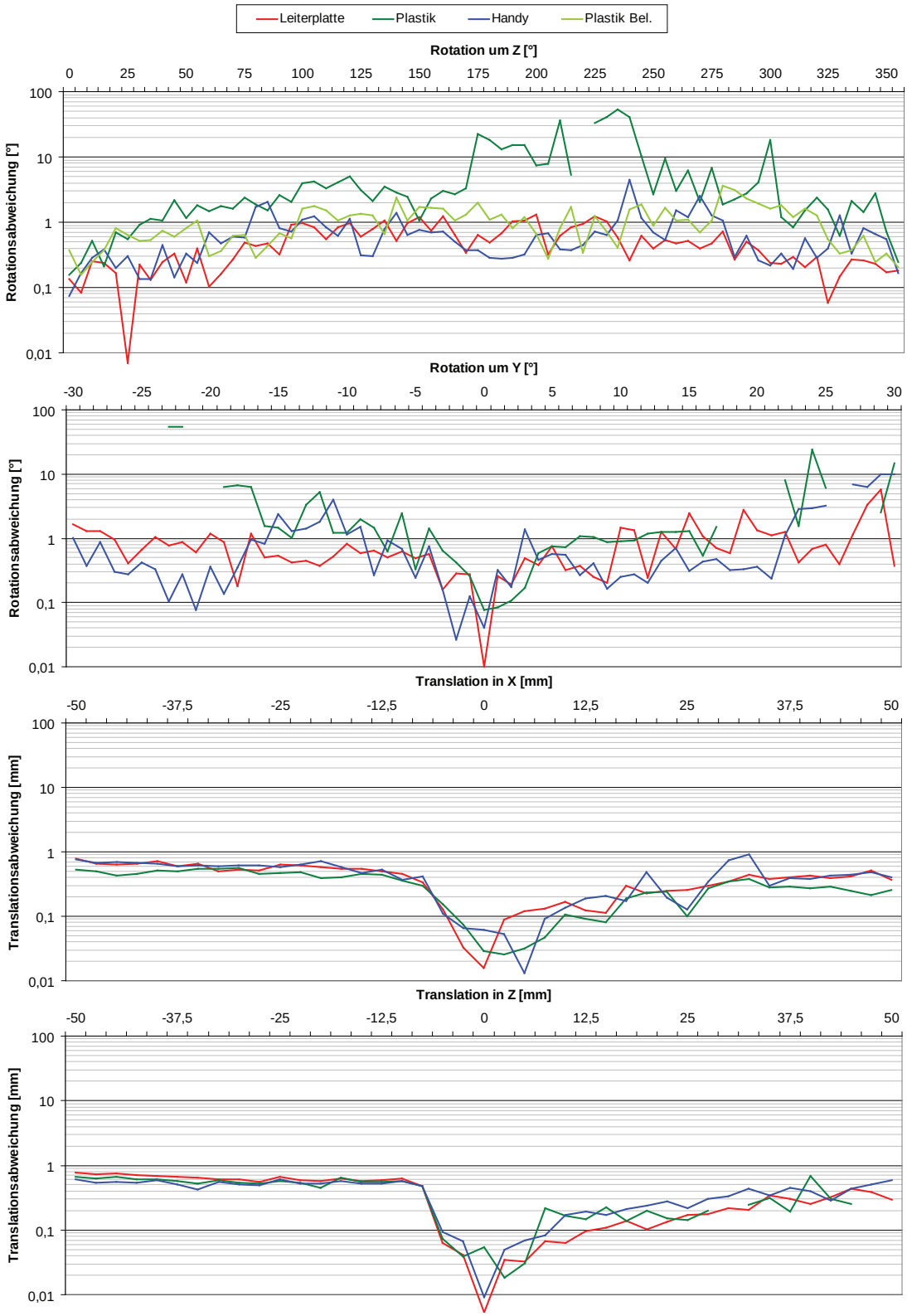

Abbildung 6.28: Von oben nach unten ist das Verhalten des Systems bei einer Blickwinkeländerung durch eine Rotation in der Bildebene um die Z-Achse der Kamera, eine Verkippung aus der Bildebene um die Y-Achse, eine Translation in der Bildebene entlang der X-Achse und eine Skalierungsänderung durch eine Translation entlang der Z-Achse abgebildet. Lücken bedeuten, dass kein Objekt gefunden wurde. Die Modellansicht ist jeweils bei 0 ° bzw. 0 mm aufgenommen.

und werden immer gefunden. In der Nähe der Modellansicht ist der Fehler aufgrund der geringen Beleuchtungsänderung am geringsten und steigt bei der Leiterplatte bei der größten Rotation um 180° nur bis auf maximal ca. 1° an. Die Abweichung beim Plastikbauteil mit seinen Strukturmerkmalen erhöht sich dagegen viel schneller bis auf weit über 10°. Das Bauteil wird an manchen Stellen sogar gar nicht mehr detektiert. Dies liegt an der in Abschnitt 6.5.3.1 ausführlich diskutierten Beleuchtungsabhängigkeit des Erscheinungsbilds bei Strukturmerkmalen. Zum Beweis wird die gleiche Sequenz für das Plastikbauteil auch mit einer stationären Beleuchtung (hellgrün Kurve) aufgenommen und verhält sich dann bzgl. Robustheit und Genauigkeit ähnlich wie das Handy.

Als zweites werden Verkippungen aus der Bildebene, d. h. eine Rotation um die X-Achse der Kamera betrachtet. Diese Blickwinkeländerungen wurden schon ausführlich in Abschnitt 6.3.2 diskutiert und hier nur der vollständigkeitshalber nochmals aufgezählt. Ebene Bauteile haben hier deutliche Vorteile, da sie weniger Probleme mit Eigenverdeckungen haben als komplexe Oberflächenstrukturen. Die Leiterplatte ist daher auch bis mind. 25° Verkippung robust erkennbar, das Plastikbauteil aufgrund seiner Strukturmerkmale nur bis. max. 15°.

Bei den beiden im Folgenden diskutierten Blickwinkeländerungen durch Translation fällt in den zwei untersten Diagrammen von Abb. 6.28 zuerst einmal die nicht ohne weiteres erklärbare Asymmetrie zwischen negativer und positiver Verschiebung auf. Der Grund liegt hierbei nicht an einer schlechteren Auswertung des Systems, sondern an unpräziseren Ground-Truth-Daten. Bei den Aufnahmen der Sequenzen hat sich, bei dem auffälligen Anstieg der Abweichungen nahe der eingelernten Modellansicht durch die Verschiebung, der Ellenbogen des Roboters umkonfiguriert. Zu erkennen ist hier die Auswirkung des absoluten Fehlers des kinematischen Modells auf die Ground-Truth-Lage, der offensichtlich bei dieser Konfiguration bei ca. 0.3 mm liegt. Das Rauschen bzw. der tendenzielle Anstieg im negativen Bereich der Translation ist auf einem anderen Niveau wie im positiven Bereich weiterhin vorhanden, durch die logarithmische Darstellung allerdings nicht mehr offensichtlich.

Die Verschiebung innerhalb der Bildebene, d. h. eine Translation in Richtung der X-Achse stellt das System vor keine Schwierigkeit, da einerseits alle Detektoren verschiebungs-kovariant sind und andererseits die Beleuchtungsänderungen eine geringere Rolle spielen. Für die Sequenz wurde eine Modellansicht mit einem mittig platzierten Objekt eintrainiert und dann im Abstand von 2.5 mm bis zu einem Maximum von ±50 mm in beide Richtungen Suchansichten so aufgenommen, dass das Objekt noch vollständig sichtbar war. Die Abweichungen bleiben selbst bei der Umkonfiguration des Roboters und dem Plastikbauteil immer unter 1 mm.

Verschiebungen entlang der Z-Achse der Kamera entsprechen Skalierungsänderungen des Erscheinungsbilds eines Objekts in der Suchansicht. Die Beleuchtungsvariationen sind dabei bis auf eine Kontraständerung aufgrund des Abstandes vernachlässigbar. Zur Untersuchung dieser Blickwinkeländerung wird bei 170 mm eine Modellansicht eingelernt und dann alle 2.5 mm bis 120 mm bzw. 230 mm eine Suchansicht aufgenommen. Dies entspricht einer Änderung der Skalierung im Bereich $[0.71, 1.35]$. Alle Detektoren sind ebenfalls skalierungs-kovariant, allerdings können kleine Modellregionen in einem noch kleineren Erscheinungsbild als während des Trainings oftmals nicht mehr wiedergefunden werden bzw. große in noch größeren Erscheinungsbildern. Um ein Objekt daher robust in unterschiedlichen Entfernungsbereichen erkennen zu können, ist es daher günstig, wenn im Training eine ausbalancierte Mischung von Regionen unterschiedlicher Größe detektiert wird. Während dies bei dem Handy und der Leiterplatte in ausreichendem Maße erfüllt ist und die Objekte über den gesamten Skalierungsbereich mit einem Fehler unter 1 mm erkannt werden, ist dies bei dem Plastikbauteil nicht der Fall. Dessen Modell enthält fast nur kleine Regionen mit einem Durchmesser von max. 5-10 pix, die bei zunehmender Entfernung nicht mehr zuverlässig detektiert werden können. Dies führt dazu, dass das Cluster nicht mehr robust erkannt werden kann und es dann zu den abgebildeten Aussetzern kommt.

Zusammenfassend lässt sich sagen, dass Verkippungen aus der Bildebene die schwierigsten Blickwinkeländerungen darstellen und daher während des Modelltrainings berücksichtigt werden sollten.

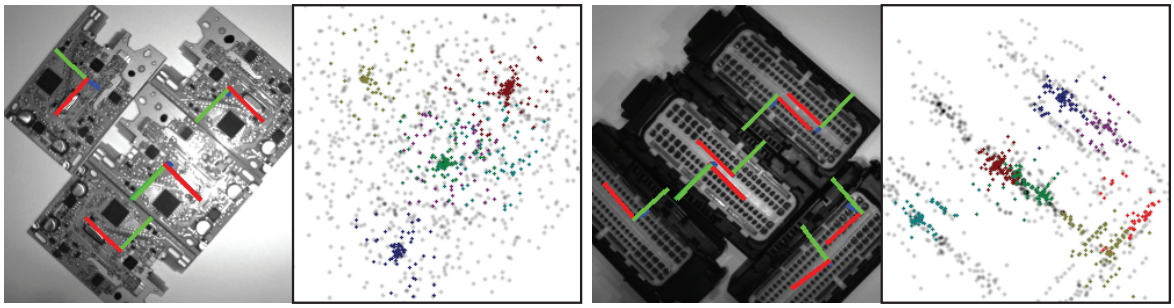

Abbildung 6.29: Verhalten des Systems bei multiplen Objekten desselben Typs. Es wurde nach max. 7 Clustern gesucht und dafür eine Lagehypothese ermittelt. Bei der Leiterplatte werden überzählige Hypothesen zuverlässig verworfen, bei dem Stecker aufgrund der Mehrdeutigkeiten nicht immer, wie in der Beispielansicht an den 7 abgebildeten Lagehypothesen ersichtlich. Gut zu erkennen ist bei dem Stecker, dass aufgrund der Teilsymmetrie oftmals eine zweite, um 180° falsch orientiere Hypothese gefunden wird. Wird nur ein einzelner Cluster und damit eine einzelne Hypothese ermittelt, kann es in seltenen Fällen durchaus vorkommen, das die falsche Hypothese zurückgegeben wird.

Mit den anderen Fällen hat das System keine Schwierigkeiten, auch weil alle Detektoren sich bei diesen Transformationen kovariant verhalten. Allerdings gibt es bei der Rotation in der Bildebene eine Ausnahme von dieser Aussage falls die Objektstruktur sehr instabil gegenüber Beleuchtungsschwankungen ist, sowie bei Skalierungsänderungen falls die Verteilung der Regionengrößen in der Modelldatenbank unausgewogen ist.

6.5.3.4 Multiple Objekte

Eine Herausforderung für viele Detektions- bzw. Lokalisationsalgorithmen sind dicht beieinanderliegende Objekte desselben Typs. Die Schwierigkeit liegt hierbei in der Segmentierung, d. h. in der Zuordnung der Bildbereiche zu den einzelnen Objektinstanzen. Die Bereiche weisen alle dieselben charakteristischen Eigenschaften des Objektes auf und machen daher eine Trennung schwer.

Wiederum wird die Ground-Truth-Sequenz verwendet, allerdings werden diesmal für die Leiterplatte und dem Stecker neben dem zu evaluierenden Objekt mit bekannter Lage, drei weitere Objekte desselben Typs platziert. (vgl. dazu Abb. 6.29). Für die Auswertung erfolgt allerdings im Vergleich zu der normalen Ground-Truth-Sequenz mit einem Objekt eine kleine Modifikation, da der Gruppierungsschritt immer das beste Cluster zurückgibt und dieses nicht unbedingt dem zu evaluierenden Objekt entspricht. Es werden deshalb die besten 7 Cluster ermittelt und dafür die globale Lage bestimmt. Für die Evaluierung wird dann immer die Lagehypothese herangezogen, deren Position sich am nächsten an der Ground-Truth-Position befindet. Da sich max. 4 Objekte in den Suchansichten befinden, werden die Cluster, die keinem Objekt entsprechen, während der globalen Auswertung verworfen. Dies klappt bei der Leiterplatte sehr gut, bei dem Stecker aufgrund der vielen Mehrdeutigkeiten nicht, wie in Abb. 6.29 ersichtlich.

Der Gruppierungsschritt entspricht der Segmentierung der Regionen (siehe hierzu auch Abb. 6.25) und ist offensichtlich auch bei multiplen Objekten sehr robust. Da die Objekte allerdings dicht an dicht liegen, sind vergleichbar wie bei dem variierenden Hintergrund Regionen an den Objektgrenzen gestört. Ebenfalls kommt es aufgrund der erhöhten Anzahl an Objekten zu mehr Fehlkorrespondenzen.

Wie sich im Vergleich der Diagramme oben links und rechts der Abb. 6.24 zwischen der Ground-Truth-Sequenz mit einem und multiplen Objekten zeigt, hat dies auf das Clusterergebnis und daraus

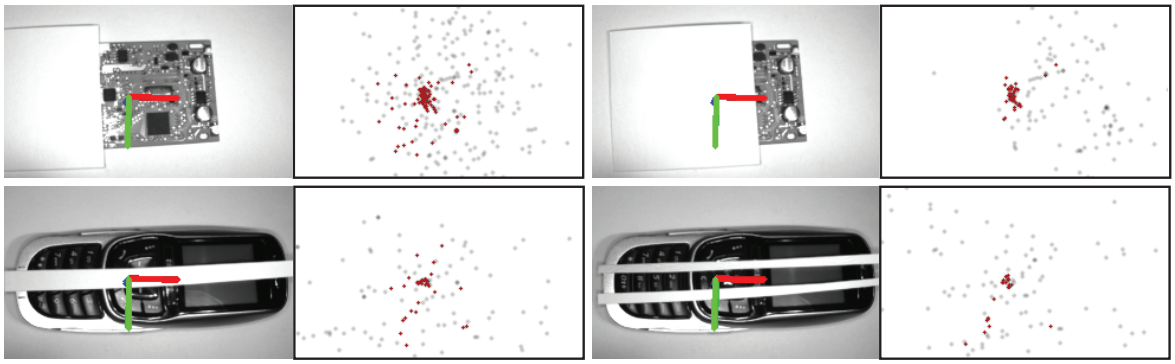

Abbildung 6.30: Verhalten des Systems bei unterschiedlichen Verdeckungen, oben die Leiterplatte mit 25% und 75% kompakter, zusammenhängender Verdeckung, unten das Handy mit jeweils 25% Verdeckung. Links jedoch zusammenhängend mit einem Streifen, rechts aufgeteilt in zwei Streifen. Je größer oder zergliederter die Verdeckung, desto schwächer ausgeprägt ist das Cluster.

resultierend die Genauigkeit der Leiterplatte keine Auswirkung. Erst wenn der Abstand zwischen den Clustern zweier Objekte zu klein wird, weil z. B. die Objekte bei ähnlicher Orientierung untereinander geschoben werden, kann es vorkommen, dass zwei Cluster zusammen gruppiert werden. In diesem Fall würde sich der RANSAC der globalen Lageermittlung für eines der beiden Zentren entscheiden. Dieser Effekt ist in Abb. 6.33 zu sehen, trat aber bei dieser Evaluierung nicht auf.

Beim Vergleich des Steckers könnte man mit Verwunderung feststellen, dass auf einmal seine Genauigkeit, insbesondere der mittlere Fehler der Orientierung sehr viel besser geworden ist. Dies lässt sich aber mit der schon diskutierten Verwechslung der Orientierung aufgrund der Teilsymmetrie begründen. Da nun nach mehreren Clustern gesucht wird, ist das richtig orientiere Cluster auf jeden Fall dabei und der Stecker wird immer (auch) richtig erkannt.

Aufgrund dieser Ergebnisse sind multiple Objekte gleichen Typs für das System kein Problem. Auch die Suche und Auswertung nach vielen Objekten bzw. Clustern funktioniert erfolgreich. Überflüssige Cluster werden zuverlässig verworfen, außer es handelt sich, wie bei dem Stecker um Objekte mit vielen Mehrdeutigkeiten und daher ausgeprägten Nebenmaxima. In diesem Fall muss eine (in dieser Arbeit nicht realisierte) zusätzliche Überprüfung der Lagehypothesen stattfinden.

6.5.3.5 Verdeckungen

Die letzte Herausforderung sind Verdeckungen, d. h. wenn bestimmte Bereiche der Objektoberfläche in der Suchansicht nicht sichtbar sind. Dies kann z. B. bei einer Überdeckung durch andere Objekte entstehen, durch ein Abschneiden des Objekts an den Bildgrenzen oder durch Glanzpunkte, die die Oberflächeninformationen überlagern.

Zum Vergleich mit den normalen Ground-Truth-Sequenzen, werden dieselben Sequenzen mit unterschiedlichen Verdeckungen der Objekte aufgenommen. Die Leiterplatte wurde dafür mit einem Karton verdeckt, so dass 25%, 50% oder 75% der Oberfläche nicht sichtbar war (siehe Abb. 6.30). Das Handy wurde ebenfalls zu 25% verdeckt, allerdings einmal mit einem kompakten, zusammenhängenden Streifen des Kartons und das andere mal mit zwei halb so breiten Streifen. Auch wenn in beiden Fällen der gleiche Anteil des Handys verdeckt ist, ist eine zergliederte Verdeckung schwieriger, da Regionen schon gestört sind, falls nur ein Teil von ihnen von der Verdeckung betroffen ist.

Abb. 6.24 zeigt in dem Diagramm unten rechts die Ergebnisse der Sequenzen mit Verdeckungen. In allen Fällen verhält sich das System sehr robust, bei der Leiterplatte ist die Detektionsrate selbst bei 75% Verdeckungen noch bei beachtlichen 0.92. Solange auf dem verbleibenden Bereich der Objekto-

berfläche noch genügend Regionen detektiert werden, kann durch die lokale Auswertung das Objekt auch weiterhin erkannt werden. Für die Leiterplatte ist dies der Fall, da die Regionen einigermaßen gleichverteilt über die gesamte Oberfläche verstreut sind. In den Fällen wo sie nicht mehr erkannt wird, wird der verbleibende sichtbare Bereich meist durch die unvermeidbaren Glanzpunkte gestört. Mit zunehmender Verdeckung sinkt aufgrund zweierlei Effekte ebenfalls die Genauigkeit, bei der Leiterplatte z. B. von 0.6° ohne Verdeckung auf 1.8° bei 75% Verdeckung. Erstens wird das Cluster kleiner und damit stehen dem RANSAC weniger Korrespondenzen zur Verfügung und zweitens sind die zugehörigen Regionen über einen kleineren Bereich des Objektes verteilt. Letzteres führt insbesondere bei der Orientierungsschätzung zu größeren Unsicherheiten, da sie aus den relativen Positionsunterschieden mehrerer Regionenzentren berechnet werden muss. Das Handy verhält sich vergleichbar, allerdings erkennt man, dass wie erwartet bei der zergliederten Verdeckung mit zwei Streifen die Genauigkeit nochmals sinkt.

Zusammenfassend lässt sich sagen, dass Verdeckungen für das System kein Problem darstellen, solange noch genügend Regionen in den sichtbaren Bereichen der Objektoberfläche gefunden werden können. Die Genauigkeit sinkt allerdings mit zunehmender Verdeckung ab. Verstärkt wirkt sich dies auf die Orientierung aus, wenn nur noch ein kleiner zusammenhängender Objektbereich sichtbar ist. Allerdings nicht nur der Verdeckungsanteil ist ausschlaggebend, sondern auch die Kompaktheit des Verdeckungsbereichs, da mit zunehmender Zergliederung mehr Regionen gestört werden.

6.5.3.6 Komplexe Szenen

Abschließend wird die Leistung des System bei komplexen Szenen mit vielen, auch gleichartigen Objekten, variierenden Hintergründen, Verdeckungen, großen Blickwinkelbereichen und unterschiedlichen Beleuchtungsverhältnissen getestet. Für diese Szenen sind allerdings keine Ground-Truth-Informationen möglich, so dass die Auswertung rein visuell erfolgen muss. Das Verhalten des Systems kann bei allen Szenen in den Abb. 6.31, 6.32 und 6.33 überprüft werden. Dort sind sowohl die Suchansichten mit den eingezeichneten globalen Lagehypothesen wie auch die Verteilung der lokalen Lagen abgebildet. Die Auswertung erfolgt nicht an dieser Stelle sondern direkt in den Bildbeschreibungen, um die beobachtbaren Effekte besser nachvollziehen zu können.

In allen Szenen wurden die Objekte bis zu einem Verkippungswinkel von 45° eingelernt. Für die Box und die Leiterplatte wurden alle 15° eine Modellansicht eintrainiert, für das Handy, den Stecker und das Plastikbauteil alle 10°. Zusätzlich wurden für das Plastikbauteil und den Stecker noch alle 90° in der Bildebene rotierte Ansichten eintrainiert. Bei der Leiterplatte und Box waren es insgesamt jeweils 16 Modellansichten, bei dem Handy 25 und dem Plastikbauteil und Stecker jeweils 100. Die größte Modelldatenbank hat der Stecker mit 22000 Regionen, die kleinste das Plastikbauteil mit 4700.

Eine Ausnahme stellt das Training der Koppeleinheit da. Für dieses Objekt wurden keine unterschiedlichen Verkippungen eintrainiert, sondern insgesamt 72 Ansichten mit 9 unterschiedlichen Verschiebungen im gesamten Bildbereich und jeweils 8 Rotation in 45° Schritten in der Bildebene. Damit kann dann auch für dieses wenig robuste Objekt die einzelnen Instanzen einschließlich deren Orientierung in einer parallel zur Bildebene ausgerichteten Palette erfolgreich erkannt werden. Dies lässt sich in der ersten Szene in Abb. 6.31 erkennen.

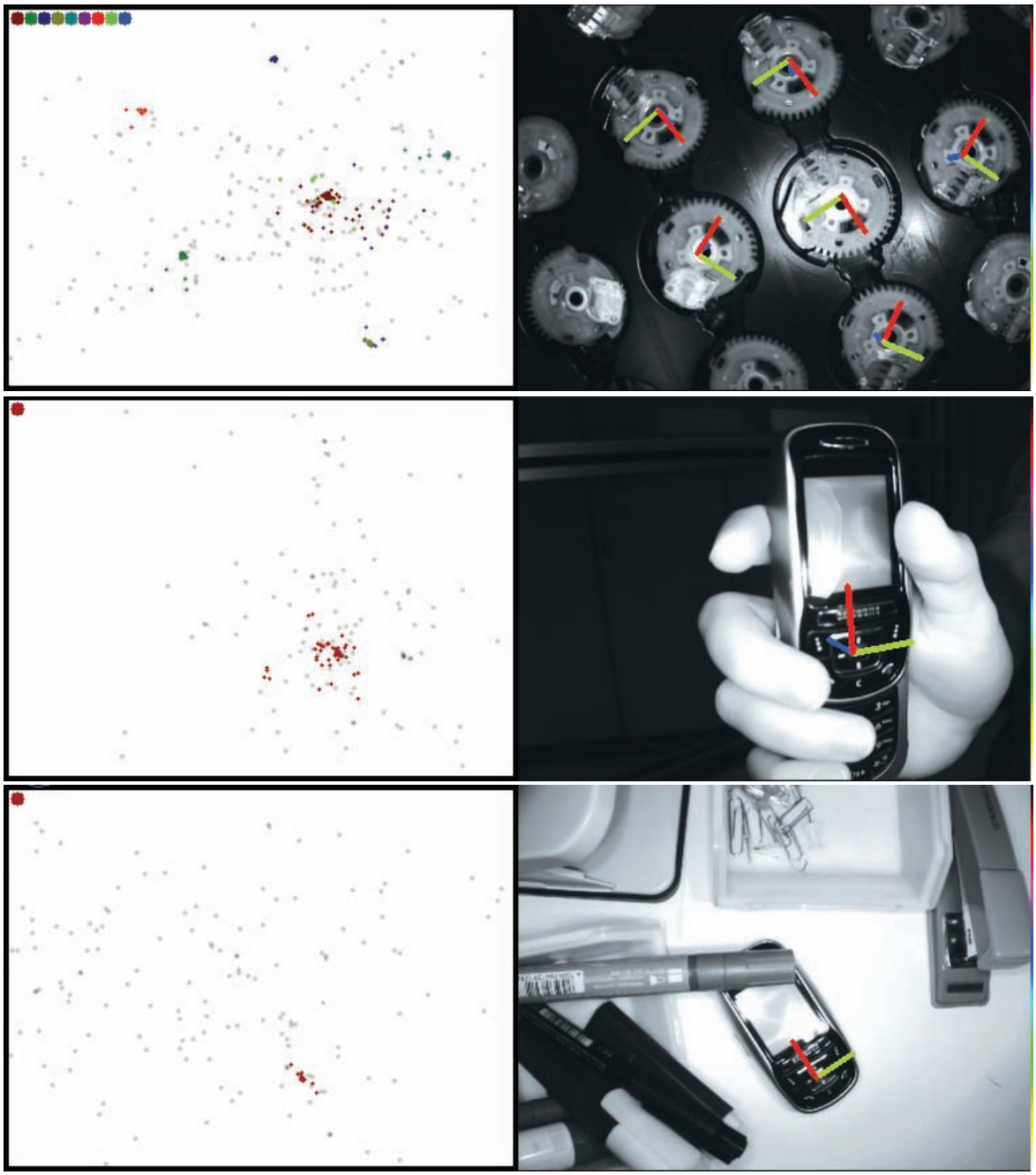

Abbildung 6.31: Die *oberste* Szene zeigt eine parallel zur Bildebene ausgerichtete Palette mit vielen Koppeleinheiten. Diese werden aufgrund eines optimierten Trainings robust inkl. korrekter Orientierung erkannt. Wie die Verteilung der lokalen Lagen zeigt, sind die Cluster relativ schwach. Daher werden die am Rand befindlichen Objekte auch nicht mehr erkannt. Das hellgrüne und -blaue Cluster werden falsch gefunden, von der globalen Lageermittlung allerdings korrekt verworfen. Die *mittlere* Szene zeigt eine robuste Lokalisation des in der Hand gehaltenen Handys, die *untere* Szene dasselbe Handy in zusammengeschobenem Zustand. Auch dort wird es korrekt erkannt, allerdings aufgrund der deutlich kleineren Objektoberfläche und damit einhergehendem deutlich schwächeren Cluster weniger robust.

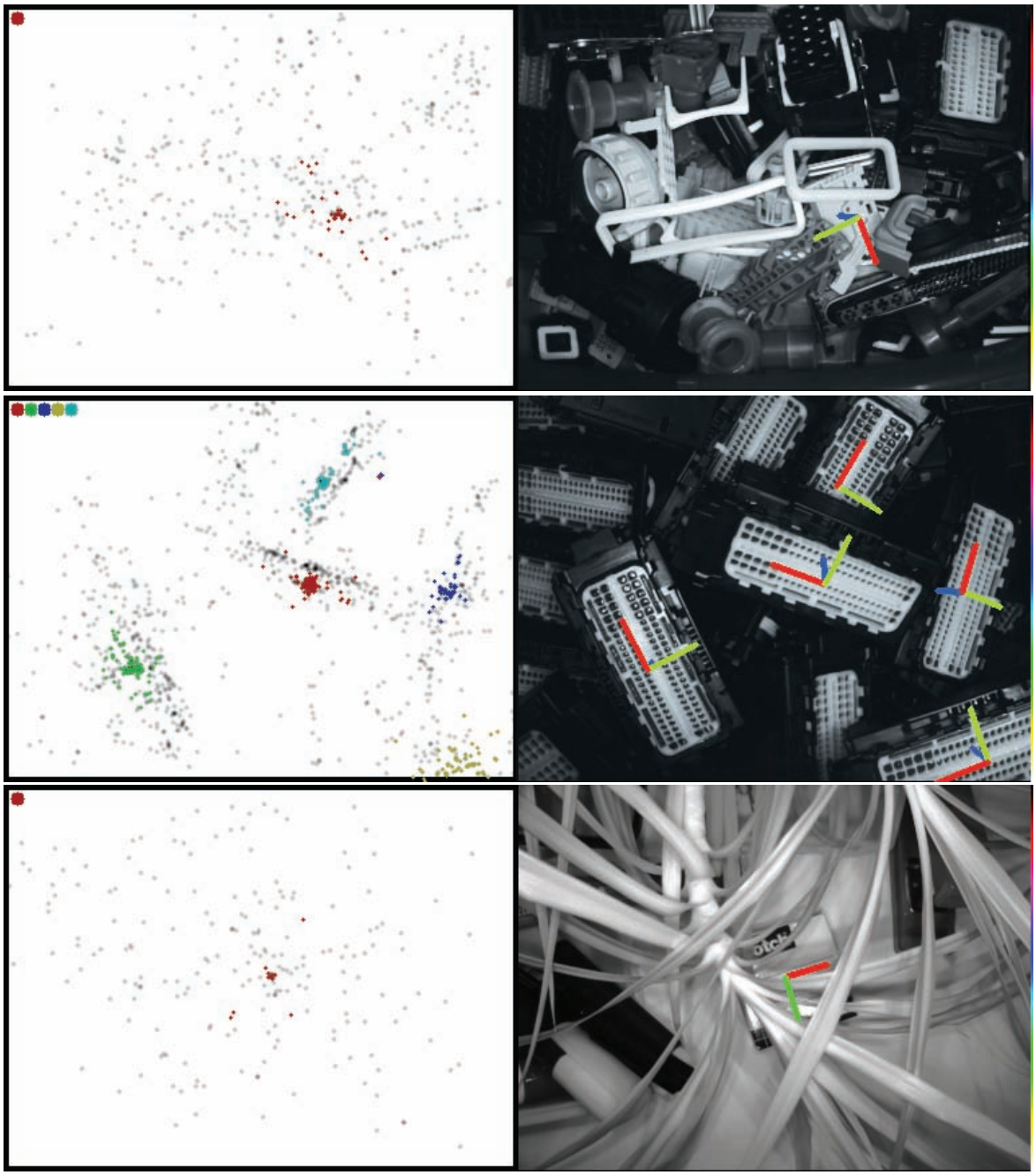

Abbildung 6.32: Die *obere* und *mittlere* Szene zeigen für das Plastikbauteil und den Stecker einen erfolgreichen Blick in die Kiste. Insbesondere die Lokalisation des Steckers ist zuverlässig, da durch die Vielzahl an eintrainierten Ansichten weniger Verwechslungen beim Clustern und der Lageermittlung auftreten. Die *untere* Szene zeigt eine extrem verdeckte Box, die allerdings immer noch korrekt erkannt wird. Allerdings zeigt die Größe des Clusters, dass das System an seine Grenzen gelangt und eine weitere Verdeckung nicht mehr kompensieren könnte.

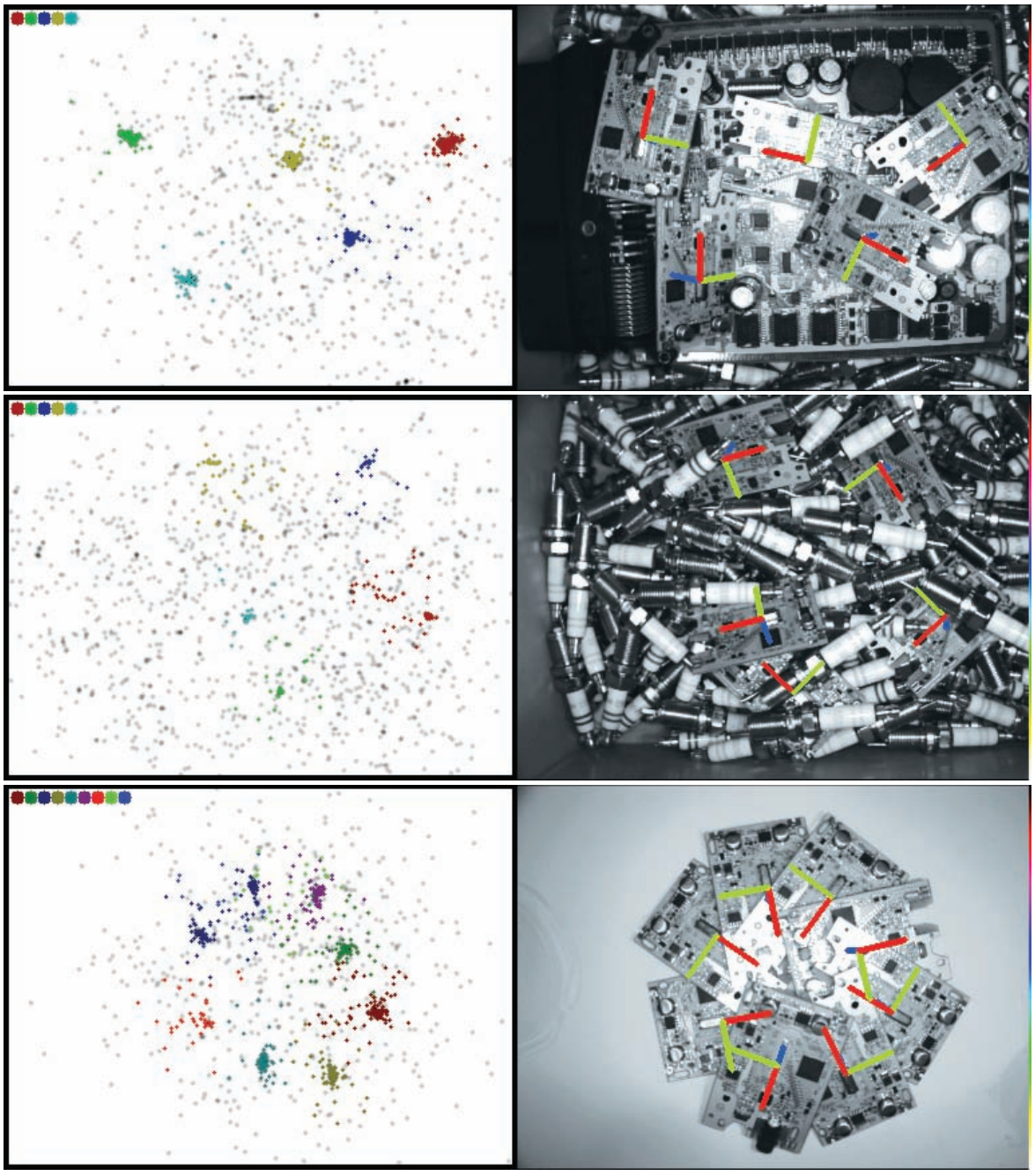

Abbildung 6.33: Die *obere* und *mittlere* Szene zeigen für die Leiterplatte ebenfalls einen erfolgreichen Blick in die Kiste. Oben sind die einzelnen Objekte weniger verdeckt, dafür befindet sich im Hintergrund ein weiteres Elektronikbauteil mit ähnlichen Strukturen. Alle Instanzen werden in beiden Bildern mit korrekter Lage erkannt, in der mittleren Szene aufgrund der teilweise großen Verdeckungen allerdings weniger robust als in der oberen Szene. Der ähnliche Hintergrund in der obersten Szene stört das System nicht. In der *unteren* Szene werden 8 ineinander geschobene Leiterplatten korrekt identifiziert und deren Lager ermittelt. Allerdings gruppiert das dunkelblaue Cluster aufgrund der Nähe die lokalen Lagen zweier Leiterplatten. Dieser Fall zeigt die Bedeutung und Robustheit des RANSAC-Ansatzes der nachfolgenden globalen Lageermittlung, der dennoch erfolgreich die korrekte Lage der linken der beiden zusammen segmentierten Leiterplatten ermitteln kann. Die rechte Platte wird in diesem Fall (allerdings wenig robust) von dem hellgrünen Nebencluster richtig erkannt.

ALLES WISSEN UND ALLES VERMEHREN UNSERES
WISSENS ENDET NICHT MIT EINEM SCHLUSSPUNKT,
SONDERN MIT EINEM FRAGEZEICHEN.

Hermann Hesse

Kapitel 7

Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war es ein System zu entwickeln, welches autonom ohne Zusatzwissen texturierte Objekte mit einer kalibrierten Stereokamera einlernen und sie anschließend online mit derselben Sensorik detektieren und ihre Lage in allen 6 Freiheitsgraden bestimmen kann.

Im Folgenden wird die in dieser Arbeit entwickelte Lösung zuerst zusammengefasst, daraufhin weiterführende Arbeiten diskutiert und abschließend ein genereller Ausblick für die Problemstellung gegeben.

Zusammenfassung Um die Aufgabenstellung lösen zu können, wurde ein ansichtsbasiertes Lokalisationssystem entwickelt, welches lokale Korrespondenzen zwischen affin- oder ähnlich-kovarianten Regionen einer Modelldatenbank und ein oder mehreren Suchansichten bzw. Kamerabildern nutzt, um daraus eine robuste 6 DoF Lagehypothese ableiten zu können.

Dazu wurde in dieser Arbeit ein neu entwickeltes geometrisches Modell vorgestellt, welches die Beziehungen zwischen einer 6 DoF Starrkörpertransformation einer lokalen Objektstruktur im 3D-Raum mit der daraus induzierten 6 DoF affinen Transformation der projizierten Bildstruktur im 2D-Raum beschreibt. Dafür wurden die *lokalen* Annahmen getroffen, dass man die Objektstruktur als Ebene modellieren kann, sich die Verzerrung der Linse linear verhält und man die perspektivische Projektion in eine skalierte Parallel-Projektion überführen kann.

Darauf aufbauend wurde eine neue, analytisch geschlossene Lösung für die inverse Problemstellung des geometrischen Modells entwickelt, mit der man aus beobachteten affinen oder ähnlichen Transformationen einer Bildstruktur die zugrundeliegende Starrkörpertransformation der Objektstruktur im 3D-Raum rekonstruieren kann. Dabei gibt es allerdings zwei Einschränkungen, bei ähnlichen Transformationen können keine Verkippungen der Objektstruktur aus der Bildebene hinaus berücksichtigt werden und bei affinen Transformationen erhält man zwei Lösungen. Weiterhin muss in beiden Fällen einmalig die Tiefe des Zentrums der Objektstruktur sowie bei der affinen Transformation die Parameter der approximierten Ebene bekannt sein.

In einem autonomen Trainingsschritt können ohne Vorwissen verschiedene Ansichten eines Objekts mit bekanntem Blickwinkel durch den Roboter aufgenommen, mit Hilfe von kovarianten Regionendetektoren wie dem SIFT-DoG oder MSER saliente Objektstrukturen in der Ansicht identifiziert und mittels lokaler Regionen beschrieben werden. Diese werden dann zusammen mit dem Blickwinkel der zugehörigen Ansicht, den Tiefeninformationen und den geschätzten Ebenenparametern in einer Modelldatenbank abgelegt. Die letzten beiden Größen lassen sich im Training mittels eines triangulationsbasierten Tiefenbilds der Stereokamera schätzen.

Während der online Lokalisation können mit Hilfe der kovarianten Regionendetektoren dieselben salienten Objektstrukturen in neuen Suchansichten gefunden und anhand einer robusten Beschreibung der Regioneninhalte, z. B. mit dem SIFT-Deskriptor, den Modellregionen zugeordnet werden. Durch die kovariante Eigenschaft der Detektoren kann aus jeder einzelnen gebildeten Korrespondenz

die lineare Bilddeformation und darüber mittels den neu entwickelten Algorithmen der Lageunterschied des Objekts im Vergleich zum Training, d. h. eine (lokale) Lagehypothese bestimmt werden.

Dieses Vorgehen ist ein entscheidender Punkt des neu entwickelten Systems, denn es erlaubt die Überführung des Unterschieds korrespondierender lokaler Regionen aus der Domäne der Bildebene in die Domäne des 3D-Szenenraums. Über das Auffinden von Clustern im 6D-Raum der gefundenen lokalen Lagehypothesen wurde eine neuartige Gruppierungsmethode der Korrespondenzen angegeben, welche eine zuverlässige Erkennung von Fehlzuordnungen, eine robuste Segmentierung von Korrespondenzen unterschiedlicher Objekte und Instanzen desselben Typs sowie eine integrierte Behandlung aller Modell- und Suchansichten in einem Schritt ermöglicht. Insbesondere Letzteres kann allein in der Bilddomäne nicht realisiert werden.

Jede einzelne Lage, die aus einer lokalen Regionenkorrespondenz abgeleitet wurde, ist eine Lösung für das in dieser Arbeit behandelte Lokalisationsproblem. Allerdings sind diese Lagen aufgrund von Fehlzuordnungen bei der Korrespondenzfindung nicht sehr vertrauenswürdig und besitzen durch ihren lokalen Informationsgehalt eine begrenzte Genauigkeit. Daher werden in dieser Arbeit die Cluster der lokalen Lagen genutzt, um eine robuste, exakte globale Lagehypothese aus den, über das gesamte Objekt verteilten zugehörigen Regionenkorrespondenzen zu erhalten. Es wurden mehrere Algorithmen für diese Aufgabenstellung vorgestellt, wobei sich der 3D-3D- und 2D-3D-OI-Algorithmus am besten bewährt haben.

Der 3D-3D-Algorithmus verwendet die 3D-Zentren der korrespondierenden Regionen in Verbindung mit RANSAC, um eine globale Tranformation zwischen den Modell- und Suchansichten zu berechnen. Dafür sind allerdings Tiefeninformationen für die Regionenzentren der Suchansichten nötig, die von dem Tiefenbild der Stereokamera bereitgestellt werden. Alternativ kann die Tiefe auch über den Skalierungsunterschied der Regionenkorrespondenzen ermittelt werden. Es hat sich in dieser Arbeit allerdings gezeigt, dass die Genauigkeit dieser Tiefeninformationen aufgrund der lokalen Auswertung nicht ausreicht. Mit dem 3D-3D-Algorithmus kann das komplette System mit ca. 3 Hz betrieben werden und erreicht bei geeigneten Objekten eine Genauigkeit von unter 1 mm und 1°.

Steht für die online Lokalisation kein Stereosystem zur Verfügung, können anstatt der 3D-Zentren der Suchregionen deren 2D-Bildpunkte genutzt werden. Der dafür vorgestellte 2D-3D-OI-Algorithmus nutzt ebenfalls RANSAC, ist aber aufgrund seiner zusätzlichen iterativen Struktur langsamer und durch inhärente Einflüsse aufgrund der Projektion insbesondere bei der Schätzung der Verkippung aus der Bildebene hinaus ungenauer. Er benötigt keinen Stereoaufbau und kann mit ca. 1 Hz betrieben werden. Der Algorithmus erreicht bei günstigen Objekten eine Genauigkeit von bis zu 1 mm und 1°, realistischer ist bei dieser Auswertung allerdings höchstens 2 mm und 5° zu erwarten.

Alle in dieser Arbeit entwickelten Algorithmen lassen sich hervorragend parallelisieren und erlauben erst durch die vollständige parallele Implementation die Ausnutzung moderner Multicore-Hardware und damit die genannten Auswertezeiten.

Vorausgesetzt es lassen sich auf einem Objekt genügend saliente Strukturen finden, verhält sich das System aufgrund der hintereinander geschalteten lokalen und globalen Auswertung sowie der integrierten Berücksichtigung aller Modell- und Suchansichten äußerst robust gegenüber Einflussgrößen, wie variierende Beleuchtung, Verdeckungen, Hintergrundänderungen, starken Blickwinkeländerungen oder vielen, selbst gleichen Objekten. Dies wurde in dieser Arbeit abschließend anhand zahlreicher komplexer Szenen belegt.

Weiterführende Arbeiten Das in dieser Arbeit vorgestellte System bildet eine komplette, funktionsfähige Verarbeitungskette eines Lokalisationssystems - von der Eingabe der Bilder bis zu der Rückgabe der 6 DoF Lagehypothesen - ab und wurde vollständig im Rahmen dieser Arbeit entworfen und implementiert. Aufgrund des begrenzten Zeitrahmens verbleiben allerdings einige interessante Themen für weitere Verbesserungen des Systems:

- Eine Hinzunahme weiterer komplementärer kovarianter Regionendetektoren würde das System gleich in mehreren Punkten positiv beeinflussen. Es würde einerseits die Genauigkeit des Systems erhöhen und andererseits die Robustheit, insbesondere bei Bauteilen mit wenigen salienten Strukturen deutlich verbessern. Geeignete Kombinationen wären die affin-kovarianten Harris- und Hessian-Regionendetektoren, die zusammen mit dem MSER-Detektor saliente Strukturen auf Basis der „nullten“ , ersten und zweiten Ableitung des Erscheinungsbilds eines Objekts detektieren. Man beschränkt sich damit allerdings immer noch auf Oberflächen mit günstiger Textur oder Struktur. Ein deutlich größeres Spektrum an erkennbaren Objekten würde die Hinzunahme von kovarianten Regionen auf Basis von 3D-Informationen bringen. Dieser Schritt wurde in Abschnitt 4.3.2 auch schon konzeptionell ausgearbeitet und ermöglicht durch die neu entwickelte, universelle Repräsentation von Korrespondenzen über die lokale Lagehypothesen eine einfache Integration beider Detektorklassen. Für die 3D-Regionen sind deutlich besser aufgelöste Tiefeninformationen nötig, als das verwendete Stereosystem zur Verfügung stellen kann. Deshalb wäre ebenfalls eine verbesserte Sensorik nötig.

- Der verwendete SIFT-Deskriptor überzeugt durch eine gute Balance zwischen Einfachheit, geringer Berechnungskomplexität und Güte bei der Zuordnung. Eine interessante Alternative wären die beiden ordnungsbasierten Deskriptoren OSID- oder R-SIFT. Sie sind zwar geringfügig langsamer bei der Berechnung, zeigen aber eine erhöhte Robustheit bei der Korrespondenzfindung. Der R-SIFT-Deskriptor hätte noch einen zweiten Vorteil, da dessen Merkmalsvektor nicht mit Fließkomma- sondern mit Ganzzahlen codiert werden kann. Damit ließe sich der gesamte Matcher auf Ganzzahl-Arithmetik umstellten und ein deutlicher Performance-Gewinn erzielen, der vermutlich die komplexere Berechnung des Deskriptors bei weitem kompensiert.

- Das Training neuer Objekte funktioniert vollständig autonom. Im Vorfeld muss von einem Benutzer lediglich der abzudeckende Blickwinkelbereich und dessen Diskretisierung beim Einlernen der Modellansichten gewählt werden. Dies ist nicht optimal, da der Benutzer anhand der Oberflächenmerkmale die maximal erlaubten Blickwinkelunterschiede zwischen zwei Modellansichten schätzen muss und diese innerhalb des gesamten einzulernenden Bereichs fix sind. Aus Sicherheitsgründen wird daher meist ein sehr kleiner Unterschied gewählt und die Modelldatenbank mit vielen Ansichten unnötig aufgebläht. Ein flexiblerer Ansatz sollte explorativ den gesamten Blickwinkelbereich untersuchen und eine optimale Verteilung der Modellansichten festlegen.

- Durch die Überführung der Korrespondenzen in die universelle Repräsentation mittels lokaler Lagen, lassen sich sowohl unterschiedliche Detektorklassen, Modell- als auch Suchansichten fusionieren. Dies bezieht sich nur auf eine räumliche Fusion, bei der die verschiedenen Ansichten die gleiche Szene aus unterschiedlichen Blickwinkeln abbilden. Die Ansichten müssen daher entweder zeitgleich aufgenommen werden oder es muss sich um eine statische Szene handeln. Um zeitliche Informationen fusionieren zu können, muss das Verhalten der lokalen Lagen über die Zeit modelliert werden. Dazu kann das gesamte System in ein probabilistisches Partikelsystem überführt werden, worin jede räumlich oder zeitlich unterschiedliche Suchansicht eine neue Beobachtung repräsentiert. Dies würde ebenfalls völlig neue Anwendungsgebiete ermöglichen, wie die robuste Schätzung von Objekttrajektorien oder der Eigenbewegung der Sensorik innerhalb der Szene.

- Der Clusteralgorithmus ist sehr robust und hat sich selbst bei komplexen Szenen mit vielen Fehlkorrespondenzen bewährt. Er verwendet eine fixe Clustergröße, die unter Umständen zur Verschmelzung benachbarter Cluster führen kann, falls zwei Objekte mit ähnlicher Orientierung sehr dicht beieinander liegen (siehe Abb. 6.33). In diesem Fall wäre eine adaptive Ermitt-

lung der Clustergröße sinnvoller, die aufgrund eines Kompaktheitsmaßes den optimalen Wert bestimmt.

- Bei der Optimierung der Modelldatenbanken werden aufgrund einer Evaluierungssequenz ungeeignete Modellregionen identifiziert und in der Datenbank markiert bzw. sofort gelöscht. Dafür ist jedoch ein Roboter notwendig, der die Ground-Truth-Informationen generiert. Es wäre daher interessant, mittels des schon vorhandenen Datensatzes von ungeeigneten Modellregionen einen Klassifikator zu trainieren, der ohne Evaluierungssequenz ungeeignete Regionen bzw. Oberflächenmerkmale identifizieren und zurückweisen kann.

- Zurzeit erfolgt eine Überprüfung der gefundenen Lagehypothesen über das Qualitätsmaß der globalen Lageermittlung. Dies ermöglicht eine erfolgreiche Zurückweisung der Hypothesen bei fälschlicherweise gefundenen Clustern zufälliger Fehlkorrespondenzen. Handelt es sich bei dem fehlerhaften Cluster um einen Häufungspunkt, der aufgrund von Verwechslungen durch symmetrische oder periodische Elemente auf ungeeigneten Objektoberflächen entstanden ist, kann es sein, dass das Qualitätsmaß aufgrund übereinstimmender relativer Lagebeziehungen zu groß wird um sie für die Zurückweisung noch verlässlich nutzen zu können. In diesem Fall müssen alternative Plausibilitätskriterien entworfen werden, die eine erfolgreiche Unterdrückung von Lagehypothesen auf Basis von Fehlclustern ermöglichen. Ein Ansatz wäre die Überprüfung der Konfiguration aller gefundener Objektlagen auf Überlappungen um notfalls die am schlechtesten bewerteten Lagehypothesen zu verwerfen.

- Eine weitere Reduzierung der Laufzeiten lässt sich bei den in dieser Arbeit entwickelten Komponenten aufgrund deren Parallelisierung auf einfache Weise durch Hinzunahme weitere Rechenkerne erreichen. Diese Komponenten besitzen aber bei dem besten und schnellsten System, der 3D-3D-Auswertung inkl. Stereo nur einen geringen Anteil der gesamten Rechenzeit. Den größten Benefit würde dort eine Portierung des Stereo-Algorithmusses auf die GPU bringen, welches eine Auswertezeit der gesamten Verarbeitungskette von bis zu 10 Hz ermöglichen sollte. Ebenfalls ließe sich die 2D-3D-Auswertung mit dem OI-Verfahren durch eine dynamische Lastverteilung der parallelen RANSAC-Zyklen bei der globalen Lageauswertung schneller als 2 Hz betreiben.

Ausblick Das vorgestellte System zeigt auf geeigneten Objekten eine Performance und Robustheit die zumindest ansatzweise mit bestimmten Fähigkeiten der menschlichen visuellen Wahrnehmung, wie dem Umgang bei Verdeckungen oder multiplen Objekten zu vergleichen ist. Allerdings gelingt dies nur, falls sich eine genügend große Anzahl starker Merkmale auf der Objektoberfläche finden lassen, die eine kovariante Regionenkonstruktion und eine eindeutige Beschreibung zulassen. Wenn diese Merkmale allerdings wie z. B. bei einer unbedruckten Tasse oder einer Schraubenmutter fehlen, kann die Auswertung keine Gruppen bzw. eindeutigen Kompositionen lokaler Strukturen mehr finden und die Erkennung schlägt fehl.

Der Mensch zeigt auch in diesen Fällen eine hervorragende Erkennungsleistung. Offensichtlich schafft er es ebenfalls schwächere Merkmale wie z. B. Kanten auszuwerten und deren Informationen beim Finden von globalen Kompositionen zu nutzen. Für das vorgestellte System bedeutet dies, dass bei der lokalen Auswertung nicht mehr nur vollständig definierte Lagehypothesen verwendet werden dürfen, sondern die universelle Darstellung so aufgeweitet werden muss, dass ebenfalls unvollständige bzw. unsichere Informationen berücksichtigt werden können.

Ein weiterer Punkt betrifft die Erkennung der informationsreichen und eindeutigen Strukturen auf der Objektoberfläche. In der Einleitung dieser Arbeit wurde im Rahmen der menschlichen Objekterkennung die Theorie der komponentialen Erkennung (Bie87) vorgestellt, bei der Objekte anhand der Komposition von lokalen Komponenten, den Geons erkannt werden. Diese Geons wurden in der

Einleitung vereinfacht als saliente Strukturen bezeichnet, die trotz ihrer lokalen Definition genügend Information besitzen, um sie für die lokale Auswertung heranzuziehen. Die Geons werden ihrerseits aus primitiven kantenbasierten Merkmalen, wie Krümmung, Parallelität, Schnittpunkte oder Symmetrie gebildet. Die Geons sind also selbst starke Merkmale, die aus einer Komposition von schwachen Merkmalen aufgebaut sind und sich daher auch auf schwierigeren Bauteilen wie der angesprochenen Tasse oder Schraubenmutter finden lassen. Die Erkennung besteht daher eigentlich aus drei Schritten: Erkennung von primitiven Merkmalen, Gruppieren dieser schwachen Merkmale zu starken Merkmalen sowie Gruppieren der starken Merkmale zu robusten Hypothesen. In dem vorliegenden System fehlt der erste Schritt und es werden nur Oberflächenstrukturen verwendet, die sofort starke Merkmale bilden. Für schwierige Oberflächen sollte diese Vorverarbeitung ergänzt werden.

Selbst wenn diese beiden Konzepte in einem System umgesetzt werden, ist es fraglich, ob eine mit dem Menschen annähernd vergleichbare visuelle kognitive Leistung erreicht werden kann. Denn dieser kann auf viele weitere Fähigkeiten wie kontextabhängige Interpretation, Szenenverständnis oder ganz abstrakt der Intuition zurückgreifen, die das Erkennen von Objekten stark unterstützen. Diese weiterführenden Fähigkeiten sind jedoch noch schwieriger nachzubilden als die Erkennung an sich und werden die Forschung weit über die Grenzen des maschinellen Sehens noch für die nächsten Jahrzehnte beschäftigen.

Anhang A

Notation und Namenskonventionen

Im Folgenden wird die in dieser Arbeit verwendete Notation vorgestellt:

- Skalare Größen s werden klein und kursiv, Vektoren $\mathbf{v} = (v_1, \cdots, v_n)^{\mathrm{T}}$ in einem n-dimensionalen Vektorraum klein und fett dargestellt. $|\mathbf{v}|$ bezeichnet den Betrag, d.h. die Länge eines Vektors.

- Matrizen werden groß und fett geschrieben. Zu einer Matrix $\mathbf{A}$ ist $\mathbf{A}^{\mathrm{T}}$ die transponierte Matrix. Für $(\mathbf{A}^{\mathrm{T}})^{-1} = (\mathbf{A}^{-1})^{\mathrm{T}}$ wird die kompakte Darstellung $\mathbf{A}^{-\mathrm{T}}$ benutzt. Die Elemente einer $n \times m$-dimensionalen Matrix $\mathbf{A}$ werden mit $a_{ij} = (\mathbf{A})_{ij}, 1 \leq i \leq n, 1 \leq j \leq m$ bezeichnet. Für eine eindeutige Kennzeichnung der Dimension wird auch $\mathbf{A}_{n\times m}$ geschrieben. $|\mathbf{A}|$ bezeichnet die Determinante, $\mathrm{tr}\,\mathbf{A}$ die Spur von $\mathbf{A}$.

- Mengen werden kalligrafisch geschrieben. Für eine Menge $\mathcal{M}$ bezeichnet $|\mathcal{M}|$ im kontinuierlichen Fall das Volumen im diskreten Fall die Anzahl der Elemente.

- Funktionen sind nach ihren Rückgabewerten Skalar f, Vektor $\mathbf{f}$, Matrix $\mathbf{F}$ oder Menge $\mathcal{F}$ formatiert und an ihrer zusätzlichen Klammerung zu erkennen. Skalare Funktionen f werden zur besseren Übersicht manchmal auch nicht-kursiv dargestellt. Eingabegrößen in runden Klammern, z.B. $f(x)$ entsprechen einem tatsächlichen Funktionscharakter über x in einem zusammenhängenden Bereich, eckige Klammern dagegen einer Parametrierung der Funktion, z.B. $\mathcal{F}[\mathbf{p}]$ einer Menge in Abhängigkeit eines Parametervektors $\mathbf{p}$. Ist die Benennung der Eingabegrößen überflüssig, wird auch kurz $f(\cdot)$ geschrieben. Ableitungen werden kompakt mit tiefstehenden eckigen Klammern gekennzeichnet, z.B. ist $f_{[x]}(\cdot) = \frac{\partial f}{\partial x}(\cdot)$ bzw. $f_{[xy]}(\cdot) = \frac{\partial^2 f}{\partial x \partial y}(\cdot)$.

- Für einen Vektor $\mathbf{v} = (v_1, \cdots, v_n)^{\mathrm{T}}$ bezeichnet $\check{\mathbf{v}} = (rv_1, \cdots, rv_n, r)^{\mathrm{T}}$ die Darstellung in homogenen Koordinaten, ebenso für eine Matrix $\check{\mathbf{A}}_{n\times m}$ mit $a_{nm} = r$, bei der alle übrigen Elemente mit r multipliziert zu verstehen sind. Die skalare Größe r ist, soweit nicht anders angegeben auf 1 normiert.

- Quaternionen $\dot{\mathbf{q}} = w + \mathrm{i}x + \mathrm{j}y + \mathrm{k}z$ mit den skalaren Komponenten $w, x, y, z \in \mathbb{R}$ und der Länge $|\dot{\mathbf{q}}| = \sqrt{w^2 + x^2 + y^2 + z^2}$ werden mit einem Punkt gekennzeichnet. Die imaginären Größen i, j und k werden klein und nicht kursiv dargestellt. Es gilt $\mathrm{ii} = \mathrm{jj} = \mathrm{kk} = \mathrm{ijk} = -1$ und $\dot{\mathbf{q}}^* = w - \mathrm{i}x - \mathrm{j}y - \mathrm{k}z$ als das konjugiert komplexe Quaternion. Für einen Vektor bzw. Punkt $\mathbf{v} = (v_x, v_y, v_z)^{\mathrm{T}} \in \mathbb{R}^3$ im euklidischen Raum ist die Quaternionendarstellung $\dot{\mathbf{v}} = \mathrm{i}v_x + \mathrm{j}v_y + \mathrm{k}v_z$.

- Elemente $a^{[k]}$ einer Folge bzw. von Iterationen werden mit hochstehenden eckigen Klammern gekennzeichnet, wobei der Folgenparameter bzw. Iterationschritt k innerhalb der Klammern steht. $a^{[\cdot]}$ bezeichnet die Folge als ganzes.

- Zufallsgrößen $\tilde{x}$ werden wie Funktionen in Abhängigkeit ihres Rückgabewertes formatiert und mit einer Schlange gekennzeichnet.
- Schätzungen oder Approximationen einer Funktion oder wahren Größe x werden mit einem Dach $\hat{x}$ markiert. Insbesondere wird $\hat{x}$ als diskrete Schreibweise verwendet, falls expliziet von der kontinuierlichen Größe x unterschieden werden soll.
- Koordinatensysteme werden in großer, nicht-kursiver Schrift bezeichnet. Eine Größe $\mathbf{x}$ bezogen auf das Koordinatensystem A wird mit ${}^{\mathrm{A}}\mathbf{x}$ gekennzeichnet, eine allg. Transformation $\mathbf{T}$ von B nach A mit ${}^{\mathrm{A}}\mathbf{T}_{\mathrm{B}}$. Dies ermöglicht eine intuitive Darstellung von Koordinatensystemtransformationen, z.B. ${}^{\mathrm{A}}\mathbf{x} = {}^{\mathrm{A}}\mathbf{T}_{\mathrm{B}} \circ {}^{\mathrm{B}}\mathbf{x}$ mit einer allg. Verknüpfungsoperation $\circ$ (siehe auch Abschnitt B.1).
- In der Beschreibung von Verfahren bedeuten so weit nicht anders angegeben Ausdrücke wie $\varepsilon = 0.5$, dass ε ein einstellbarer Parameter des Verfahrens ist und der konkrete Wert 0.5 die übliche Einstellung in der Literatur.
- Namen oder Ausdrücke die als Bezeichner im Quellcode fungieren werden in `Nichtproportionalschrift` angegeben.

Weiterhin existieren einige Namenskonventionen, die in dieser Arbeit - soweit nicht anders angegeben - durchgängig verwendet werden.

Koordinaten Die Vektoren $\mathbf{p}$, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ beschreiben Punkte im dreidimensionalen Raum $\mathbb{R}^3$ mit den Koordinaten x, y, z. Die Vektoren $\mathbf{u}$, $\mathbf{v}$ bezeichnen Punkte in der Bildebene $\mathbb{R}^2$ mit der horizontalen Koordinate u und der vertikalen Koordinate v. Unverzerrte normalisierte Bildkoordinaten $\bar{\mathbf{u}}$ werden mit einem Überstrich gekennzeichnet.

Eigenwerte Zu einer geeigneten Matrix $\mathbf{A}$ beschreiben λ_i bzw. $\lambda_i(\mathbf{A})$ die Eigenwerte der Matrix und υ_i bzw. $\upsilon_i(\mathbf{A})$ die zugehörigen Eigenvektoren. Der größte und kleinste Eigenwert wird auch als $\lambda_{\max}$ und $\lambda_{\min}$ bezeichnet.

Matrixklassen Für eine kompaktere Darstellung werden für übliche Klassen von Matrizen Bezeichner eingeführt. Hochgestellte Symbole beziehen sich dabei auf die Determinate, $^-$ beinhaltet alle Matrizen mit negativer Determinate, ^{0} mit Determinate 0, $^+$ mit positiver Determinate. Ohne angegebene Hochstellung gilt $^{-+}$, d.h. alle Matrizen mit vollem Rang. Der tiefgestellte Parameter n bezieht sich auf die Dimensionalität der betrachteten Matrizen. Soweit nicht in eckigen Klammern anders angegeben, werden alle Matrizen über den Körper $\mathbb{R}$ der reellen Zahlen definiert. $O_n = \{\mathbf{A} \,|\, \mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{I}_{n\times n}\}$ beinhaltet die Menge aller orthogonalen Matrizen, $O_n^+ = \mathcal{S}O_n = \{\mathbf{R} \,|\, \mathbf{R} \in O_n \text{ und } |\mathbf{R}| = 1\}$ die spezielle orthogonale Gruppe, d. h. die Menge aller Rotationsmatrizen im $\mathbb{R}^n$, $\mathcal{T}_{S,n}^+ = \{\mathbf{S} \,|\, \mathbf{S} = s\mathbf{R},\, s \in \mathbb{R}^+,\, \mathbf{R} \in O_n^+\}$ die Matrizen der ähnlichen Transformationsgruppe ohne Spiegelung, $\mathcal{T}_{A,n}^+ = \{\mathbf{A}_{n\times n} \,|\, |\mathbf{A}| > 0\}$ die Matrizen der affinen Transformationsgruppe ohne Spiegelung und $\mathcal{S}ym_n = \{\mathbf{M}_{n\times n} \,|\, \mathbf{M} = \mathbf{M}^{\mathrm{T}}\}$ die Menge aller symmetrischen Matrizen.

Elementare Rotationsmatrizen $\mathbf{R}_x[\alpha]$, $\mathbf{R}_y[\alpha]$ und $\mathbf{R}_z[\alpha]$ beschreiben elementare Rotationen mit Winkel α um die Achsen X, Y und Z respektive. Sie sind in $\mathcal{S}O_3$ enthalten und über $r_{22,x} = r_{33,x} = r_{11,y} = r_{33,y} = r_{11,z} = r_{22,z} = \cos\alpha$ sowie $r_{32,x} = -r_{23,x} = -r_{31,y} = r_{13,y} = r_{21,z} = -r_{12,z} = \sin\alpha$ vollständig definiert.

Vektorprodukte $\mathbf{x}^{\mathrm{T}}\mathbf{y}$ bezeichnet das Skalar- bzw. innere Produkt und $\mathbf{x} \times \mathbf{y}$ das Kreuz- bzw. äußere Produkt der Vektoren $\mathbf{x}$ und $\mathbf{y}$ soweit die Produkte definiert sind.

Arcus Tangens 2 Die Darstellung $\arctan2(y,x)$ ist eine Erweiterung des $\arctan\frac{y}{x}$, um einen Winkel über das komplette Intervall $[0,2\pi[$ zu rekonstruieren. Sie ist definiert durch

$$\arctan2(y,x) = \begin{cases} \arctan\frac{y}{x} & x > 0 \\ \arctan\frac{y}{x} + \frac{\pi}{2} & x < 0 \wedge y \geq 0 \\ \arctan\frac{y}{x} - \frac{\pi}{2} & x < 0 \wedge y < 0 \\ +\frac{\pi}{2} & x = 0 \wedge y > 0 \\ -\frac{\pi}{2} & x = 0 \wedge y < 0 \end{cases}$$

Abkürzungen Alle Abkürzungen sind im Text bei der ersten Verwendung erklärt. Die wichtigsten sind hier noch einmal aufgeführt:

DoF Freiheitsgrad (engl.: degree of freedom)

ST Ähnlichkeitstransformation (engl.: similarity transformation)

AT Affine Transformation

RT Starrkörpertransformation (engl.: rigid transformation)

Anhang B

Beweise und Formeln

Dieser Abschnitt beinhaltet Beweise und Formeln, die in der vorliegenden Arbeit benutzt aber nicht erklärt wurden. Im Text sind die entsprechenden Verweise angegeben.

B.1 Lagerepräsentationen

Ein Koordinatensystem im $\mathbb{R}^3$ lässt sich durch drei im Raum befindliche Basisvektoren $\mathbf{e_x}$, $\mathbf{e_y}$ und $\mathbf{e_z}$ samt einem Ursprungspunkt $\mathbf{o}$ darstellen. In dieser Arbeit werden im $\mathbb{R}^3$ ausschließlich rechtshändige, orthonormale Koordinatensysteme verwendet, d. h. für die $\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i = 1$, $\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_j = 0$, $\mathbf{e}_i^{\mathrm{T}} \times \mathbf{e}_j = 1$ für $i, j \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$, $i \neq j$ gilt.

Ein 3 DoF *Punkt* $\mathbf{p} = (x, y, z)^{\mathrm{T}} \in \mathbb{R}^3$ bezieht sich immer auf ein Koordinatensystem und repräsentiert den Ort $\mathbf{o} + x\mathbf{e_x} + y\mathbf{e_y} + z\mathbf{e_z}$. Wird zwischen zwei Koordinatensystemen A und B mit unterschiedlichen Basisvektoren unterschieden, wird in dieser Arbeit durch einen links hochgestellten Bezeichner ${}^{\mathrm{A}}\mathbf{p}$ bzw. ${}^{\mathrm{B}}\mathbf{p}$ das Bezugskoordinatensystem spezifiziert. Eine formale Unterscheidung zwischen einen Punkt und einem Richtungsvektor gibt es in dieser Arbeit nicht, sondern muss vom Kontext erschlossen werden.

Eine 3 DoF *Orientierung* im $\mathbb{R}^3$ kann durch rechtshändige, orthonormale Richtungsvektoren $\mathbf{v_x}$, $\mathbf{v_y}$, $\mathbf{v_z}$ repräsentiert werden und ist immer bzgl. eines Koordinatensystems zu verstehen.

Eine 6 DoF *Lage* im $\mathbb{R}^3$ besteht aus einem 3 DoF Punkt $\mathbf{p}$ samt zugehöriger 3 DoF Orientierung $\mathbf{v_x}$, $\mathbf{v_y}$, $\mathbf{v_z}$ und bezieht sich ebenfalls auf ein Koordinatensystem. Eine Lage X kann als neues Koordinatensystem B relativ zu seinen Bezugssystem A aufgefasst werden, da es über $\mathbf{o} = \mathbf{p}$, $\mathbf{e}_i = \mathbf{v}_i$, $i \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ dessen obige Definition erfüllt. Notiert wird das Bezugssystem links oben, das definierte Koordinatensystem rechts unten in der Form ${}^{\mathrm{A}}\mathbf{X}_{\mathrm{B}}$.

Vier Operatoren $^{-1}$, $\bullet$, $\circ$, $\odot$ sind im Zusammenhang mit der Überführung von Informationen zwischen verschiedenen Lagen bzw. Koordinatensystemen A und B wichtig:

$^{-1}$ Die Vertauschung von Bezugssystem und definiertem Koordinatensystem: ${}^{\mathrm{A}}\mathbf{X}_{\mathrm{B}}^{-1} = {}^{\mathrm{B}}\mathbf{X}_{\mathrm{A}}$.

$\bullet$ Die Transformation eines Punktes $\mathbf{p}$: ${}^{\mathrm{A}}\mathbf{p} = {}^{\mathrm{A}}\mathbf{X}_{\mathrm{B}} \bullet {}^{\mathrm{B}}\mathbf{p}$.

$\circ$ Die Transformation eines Richtungsvektors $\mathbf{v}$: ${}^{\mathrm{A}}\mathbf{v} = {}^{\mathrm{A}}\mathbf{X}_{\mathrm{B}} \circ {}^{\mathrm{B}}\mathbf{v}$.

$\odot$ Die Änderung des Bezugssystems einer Lage Y bzw. des korrespondierendem Koordinatensystems C: ${}^{\mathrm{A}}\mathbf{Y}_{\mathrm{C}} = {}^{\mathrm{A}}\mathbf{X}_{\mathrm{B}} \odot {}^{\mathrm{B}}\mathbf{Y}_{\mathrm{C}}$.

Je nach Aufgabe gibt es unterschiedliche mathematische Repräsentationen für Orientierungen und Punkte, z. B. benötigen Robotersteuerungen oft Eulerwinkel als Eingabe, Algorithmen verwenden

gut handhabbare Rotationsmatrizen, Quaternionen werden zur Interpolation zwischen Orientierungen benötigt und homogene Matrizen eignen sich zur formalen Darstellung oft am besten. Alle diese Repräsentationen werden in dieser Arbeit bzw. der entstandenen Software verwendet und sollen daher im Folgenden samt Verknüpfungen und Umrechnungen kurz vorgestellt werden. Viele Formeln stützen sich dabei auf (SE03).

B.1.1 Rotationsmatrizen

Die bekannteste mathematische Repräsentation von Orientierungen ist die Darstellung mittels Rotationsmatrizen $\mathbf{R} \in \mathcal{S}O_3$. Die Matrix $\mathbf{R}$ ist orthogonal mit $|\mathbf{R}| = 1$ und hat in ihren Spalten die orthonormalen Basisvektoren ${}^{\mathrm{A}}\mathbf{e_x}$, ${}^{\mathrm{A}}\mathbf{e_y}$, ${}^{\mathrm{A}}\mathbf{e_z}$ des definierten Koordinatensystems B, dargestellt im Bezugssystems A. Ein Richtungsvektor $\mathbf{v}$ lässt sich über ${}^{\mathrm{A}}\mathbf{v} = \mathbf{R}\,{}^{\mathrm{B}}\mathbf{v}$ von B nach A transformieren. Die Matrix wird daher auch mit ${}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}$ bezeichnet. Weiterhin enthält die Matrix ${}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}$ über ${}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}^{-1} = {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}^{\mathrm{T}} = {}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}$ in ihren Zeilen die Basisvektoren von A bzgl. B.

Eine Lage B kann mittels $({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{o})$ in A repräsentiert werden. Der Ursprung ${}^{\mathrm{A}}\mathbf{o}$ wird allerdings oft auch mit ${}^{\mathrm{A}}\mathbf{t}$ bezeichnet, da er als Richtungsvektor aufgefasst werden kann und die Verschiebung beider Koordinatensysteme zueinander enthält. Die Operatoren definieren sich wie folgt:

$^{-1}$ $({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{o})^{-1} = ({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}^{\mathrm{T}}, -{}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}^{\mathrm{T}}\,{}^{\mathrm{A}}\mathbf{o}) = ({}^{\mathrm{B}}\mathbf{R}_{\mathrm{A}}, {}^{\mathrm{B}}\mathbf{o})$

$\bullet$ ${}^{\mathrm{A}}\mathbf{p} = ({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{o}) \bullet {}^{\mathrm{B}}\mathbf{p} = {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{p} + {}^{\mathrm{A}}\mathbf{o}$

$\circ$ ${}^{\mathrm{A}}\mathbf{v} = ({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{o}) \circ {}^{\mathrm{B}}\mathbf{v} = {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{v}$

$\odot$ $({}^{\mathrm{A}}\mathbf{R}_{\mathrm{C}}, {}^{\mathrm{A}}\mathbf{o}) = ({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}, {}^{\mathrm{A}}\mathbf{o}_{\mathrm{B}}) \odot ({}^{\mathrm{B}}\mathbf{R}_{\mathrm{C}}, {}^{\mathrm{B}}\mathbf{o}) = ({}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{R}_{\mathrm{C}}, {}^{\mathrm{A}}\mathbf{o}_{\mathrm{B}} + {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{o})$

Lagen mittels Rotationsmatrizen darzustellen hat den Vorteil, dass man die übliche Repräsentation von Punkten und Vektoren sowie die Matrixmultiplikation verwenden kann. Weiterhin ist die Darstellung intuitiv und numerisch gut handhabbar. Allerdings werden zur Kodierung der 6 DoF's 12 Einträge benötigt.

B.1.2 Homogene Matrizen

Homogene Matrizen fassen die Rotationsmatrix ${}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}}$ und den Verschiebungsvektor bzw. Ursprung ${}^{\mathrm{A}}\mathbf{o}$ in einer Matrix

$$ {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}} = \begin{pmatrix} {}^{\mathrm{A}}\mathbf{R}_{\mathrm{B}} & {}^{\mathrm{A}}\mathbf{o} \\ \mathbf{0} & 1 \end{pmatrix} $$

zusammen. Eine Lage lässt sich daher vollständig über ihre homogene Matrix ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}$ darstellen. Überführt man Punkte $\mathbf{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ bzw. Richtungsvektoren $\mathbf{v} = (v_x, v_y, v_z)^{\mathrm{T}}$ in ihre homoge Darstellung $\check{\mathbf{p}} = (p_x, p_y, p_z, 1)^{\mathrm{T}}$ bzw. $\check{\mathbf{v}} = (v_x, v_y, v_z, 0)^{\mathrm{T}}$ lassen sich alle Operatoren mittels den üblichen Matrixoperatoren ausdrücken:

$^{-1}$ $({}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}})^{-1} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}^{-1} = {}^{\mathrm{B}}\check{\mathbf{H}}_{\mathrm{A}}$

$\bullet$ ${}^{\mathrm{A}}\mathbf{p} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}} \bullet {}^{\mathrm{B}}\mathbf{p} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{p}$

$\circ$ ${}^{\mathrm{A}}\mathbf{v} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}} \circ {}^{\mathrm{B}}\mathbf{v} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}\,{}^{\mathrm{B}}\mathbf{v}$

$\odot$ ${}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{C}} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}} \odot {}^{\mathrm{B}}\check{\mathbf{H}}_{\mathrm{C}} = {}^{\mathrm{A}}\check{\mathbf{H}}_{\mathrm{B}}\,{}^{\mathrm{B}}\check{\mathbf{H}}_{\mathrm{C}}$

Aus diesem Grund eignen sich homogene Matrizen gut für eine kompakte formale Repräsentation in dieser Arbeit. Sie benötigen allerdings 16 Einträge und werden aus Effizienzgründen für die Implementierung meist in die Darstellung mittels Rotationsmatrizen überführt.

B.1.3 Eulerwinkel

Anstatt eine Orientierung als Matrix zu beschreiben, kann auch ein als Eulerwinkel bezeichnetes 3-Tupel (α,β,γ) verwendet werden. Jeder Winkel bezeichnet eine Rotation um eine der drei Koordinatenachsen $\mathbf{e_x}$, $\mathbf{e_y}$ oder $\mathbf{e_z}$. Dabei bestimmen die Achsen und die Reihenfolge der Ausführung die resultierende Orientierungsänderung. Nur mittels dieser Kodierrichtlinie sind Eulerwinkel vollständig definiert. In der entstandenen Software wird die Reihenfolge XYZ der Rotationen um *ursprüngliche* Achsen verwendet. Die Umwandlung von (α,β,γ) in eine Rotationsmatrix erfolgt damit durch:

$$\mathbf{R} = \mathbf{R}_z[\gamma]\mathbf{R}_y[\beta]\mathbf{R}_x[\alpha] = \begin{pmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta\sin\alpha - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{pmatrix}$$

Bei der Extraktion der Eulerwinkel aus einer Rotationsmatrix muss man auf Singularitäten achten. Für die hier beschriebene Kodierrichtlinie ergibt sich:

$$\begin{aligned} \cos\beta &= \sqrt{r_{11}^2 + r_{21}^2} \\ \alpha &= \begin{cases} \operatorname{arctan2}(r_{32}, r_{33}) & \cos\beta \neq 0; \\ \operatorname{arctan2}(-r_{23}, r_{22}) & \cos\beta = 0. \end{cases} \\ \beta &= \operatorname{arctan2}(-r_{31}, \cos\beta) \\ \gamma &= \begin{cases} \operatorname{arctan2}(r_{21}, r_{11}) & \cos\beta \neq 0; \\ 0 & \cos\beta = 0. \end{cases} \end{aligned}$$

Eine Lage lässt sich in Eulerwinkel kompakt als 6-Tupel $(\alpha,\beta,\gamma,o_x,o_y,o_z)$ darstellen. Diese Repräsentation ist allerdings nicht injektiv, d.h. es existieren mehrere Tupel für dieselbe Lage. Ebenfalls lässt sich mit Eulerwinkeln nur umständlich rechnen, so dass diese Lagerepräsentation meist nur als Übergabeparameter einer Schnittstelle oder als Ausgabe für den Menschen verwendet wird.

B.1.4 Quaternionen

Quaternionen $\mathbb{H}$ sind eine Erweiterung der reellen Zahlen um die imaginären Einheiten i, j, k mit $\mathrm{ii} = \mathrm{jj} = \mathrm{kk} = \mathrm{ijk} = -1$. Im Gegensatz zu den komplexen Zahlen $\mathbb{C}$ ist die Multiplikation allerdings nicht kommutativ, die Quaternionen bilden daher keinen Körper. Ein Quaternion $\dot{\mathbf{q}} = w + x\mathrm{i} + y\mathrm{j} + z\mathrm{k} \in \mathbb{H}$ wird häufig als Vektor $(w,x,y,z)^{\mathrm{T}}$ geschrieben. $\dot{\mathbf{q}}^* = w - x\mathrm{i} - y\mathrm{j} - z\mathrm{k}$ wird als das konjugierte Quaternion bezeichnet.

Quaternionen mit Betrag $|\dot{\mathbf{q}}| = \sqrt{\dot{\mathbf{q}}\dot{\mathbf{q}}^*} = 1$ werden als Einheitsquaternionen $\mathbb{H}_1$ bezeichnet und eignen sich für die Darstellung von Rotationen bzw. Orientierungen im $\mathbb{R}^3$. Eine Rotation mit Winkel 2α um die Achse ${}^{\mathrm{A}}\mathbf{e} = (e_x,e_y,e_z)^{\mathrm{T}}$ mit $|{}^{\mathrm{A}}\mathbf{e}| = 1$ lässt sich durch das Quaternion

$${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}} = (\cos\alpha, \sin\alpha\, e_x, \sin\alpha\, e_y, \sin\alpha\, e_z)^{\mathrm{T}}$$

repräsentieren und definiert eine neue Orientierung B bzgl. des Koordinatensystems A. Diese Darstellung ist allerdings nicht injektiv, da mit $-{}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}$ ein zweites Quaternion derselben Rotation existiert. Da die Achse fix bleibt, lässt sich die umgekehrte Rotation ${}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{A}}$ um den Winkel -2α durch ${}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{A}} = {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}^*$ beschreiben. Die Verknüpfung zweier Rotationen ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}$, ${}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{C}}$ erfolgt über die Multiplikation ${}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{C}} = {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{C}}$ und ergibt wiederum eine gültige Rotation, da die Einheitsquaternionen mit der Multiplikation und der Konjugation zur Bildung des inversen Elements eine Gruppe bilden. Ein Richtungsvektor $\mathbf{v} = (v_x,v_y,v_z)^{\mathrm{T}}$ entspricht dem Quaternion $\dot{\mathbf{v}} = (0,v_x,v_y,v_z)^{\mathrm{T}}$ und kann mittels dem Ausdruck ${}^{\mathrm{A}}\dot{\mathbf{v}} = {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{q}}\,{}^{\mathrm{A}}\mathbf{q}_{\mathrm{B}}^*$ von der Orientierungsdarstellung B in das Bezugssystem A transformiert werden.

Eine Lage B lässt sich daher ähnlich zu der Darstellung mittels Rotationsmatrizen durch ein Tupel $({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{A}}\dot{\mathbf{o}})$ in A repräsentieren. Der Ursprung **o** und alle Punkte bzw. Richtungsvektoren müssen für die folgenden Transformationen allerdings in Quaternionen überführt werden:

$^{-1}\ \ ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{A}}\dot{\mathbf{o}})^{-1} = ({}^{\mathrm{A}}\dot{\mathbf{q}}^{*}_{\mathrm{B}}, -{}^{\mathrm{A}}\dot{\mathbf{q}}^{*}_{\mathrm{B}}\,{}^{\mathrm{A}}\dot{\mathbf{o}}\,{}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}) = ({}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{A}}, {}^{\mathrm{B}}\dot{\mathbf{o}})$

$\bullet\ \ {}^{\mathrm{A}}\dot{\mathbf{p}} = ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{A}}\dot{\mathbf{o}}) \bullet {}^{\mathrm{B}}\dot{\mathbf{p}} = {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{p}}\,{}^{\mathrm{A}}\dot{\mathbf{q}}^{*}_{\mathrm{B}} + {}^{\mathrm{A}}\dot{\mathbf{o}}$

$\circ\ \ {}^{\mathrm{A}}\dot{\mathbf{v}} = ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{A}}\dot{\mathbf{o}}) \circ {}^{\mathrm{B}}\dot{\mathbf{v}} = {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{v}}\,{}^{\mathrm{A}}\dot{\mathbf{q}}^{*}_{\mathrm{B}}$

$\odot\ \ ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{C}}, {}^{\mathrm{A}}\dot{\mathbf{o}}) = ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}, {}^{\mathrm{A}}\dot{\mathbf{o}}_{\mathrm{B}}) \odot ({}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{C}}, {}^{\mathrm{B}}\dot{\mathbf{o}}) = ({}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{q}}_{\mathrm{C}}, {}^{\mathrm{A}}\dot{\mathbf{o}}_{\mathrm{B}} + {}^{\mathrm{A}}\dot{\mathbf{q}}_{\mathrm{B}}\,{}^{\mathrm{B}}\dot{\mathbf{o}}\,{}^{\mathrm{A}}\dot{\mathbf{q}}^{*}_{\mathrm{B}})$

Vorteile der Quaternionen-Repräsentation sind die kompakte Darstellung von 6 DoF Lagen mit 7 Einträgen und die numerische Stabilität der Transformationen. Weiterhin kann man im Gegensatz zu den anderen Darstellungen im Raum der Orientierungen bzw. Einheitsquaternionen korrekt zwischen zwei Orientierungen interpolieren bzw. mitteln [1]. Seien $\dot{\mathbf{q}}_0$ und $\dot{\mathbf{q}}_1$ zwei Orientierungen und $\dot{\mathbf{q}}_t$, $t \in [0,1]$ die dazwischenliegende interpolierte Orientierung, wobei $\dot{\mathbf{q}}_{t=0} = \dot{\mathbf{q}}_0$ und $\dot{\mathbf{q}}_{t=1} = \dot{\mathbf{q}}_1$ entspricht. Der Innenwinkel α zwischen $\dot{\mathbf{q}}_0$ und $\dot{\mathbf{q}}_1$ auf der Einheitsspähre im 4D-Raum beträgt $\cos_\alpha = \mathrm{re}(\dot{\mathbf{q}}_0\dot{\mathbf{q}}_1^*) = \mathrm{re}(\dot{\mathbf{q}}_0^*\dot{\mathbf{q}}_1) = w_0w_1 + x_0x_1 + y_0y_1 + z_0z_1$. Da immer der kleinste Innenwinkel der Rotationen gesucht ist, muss im Falle $\cos_\alpha < 0$ die alternative Darstellung $-\dot{\mathbf{q}}_0$ verwendet werden. Für $\cos_\alpha = 1$ sind beide Orientierungen identisch und es muss nicht interpoliert werden. Ansonsten ergibt sich

$$\dot{\mathbf{q}}_t(\dot{\mathbf{q}}_0, \dot{\mathbf{q}}_1) = \frac{\sin_{(1-t)\alpha}\,\dot{\mathbf{q}}_0 + \sin_{t\alpha}\,\dot{\mathbf{q}}_1}{\sin_\alpha} \tag{B.1}$$

bzw. speziell für $\dot{\mathbf{q}}_{\frac{1}{2}} = \frac{1}{2}(\dot{\mathbf{q}}_0 + \dot{\mathbf{q}}_1)(\cos_\alpha)^{-1}$. Die Umwandlung eines Einheitsquaternions $\dot{\mathbf{q}}$ in die korrespondierende Rotationsmatrix **R** erfolgt durch:

$$\begin{pmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{pmatrix}$$

Die umgekehrte Konvertierung benötigt einige Fallunterscheidungen. Für den Fall $\mathrm{tr}\,\mathbf{R} > 0$ ergibt sich $\dot{\mathbf{q}} = \frac{1}{2}(1+\mathrm{tr}\,\mathbf{R})^{-\frac{1}{2}}(1+\mathrm{tr}\,\mathbf{R}, r_{32}-r_{r23}, r_{13}-r_{31}, r_{13}-r_{12})^{\mathrm{T}}$. Ansonsten muss das maximale Element der Hauptdiagonalen von **R** bestimmt werden. Ist dies r_{11}, ergibt sich $\dot{\mathbf{q}} = \frac{1}{2}(1+r_{11}-r_{22}-r_{33})^{-\frac{1}{2}}(r_{32}-r_{23}, 1+r_{11}-r_{22}-r_{33}, r_{12}+r_{21}, r_{13}+r_{31})^{\mathrm{T}}$, bei r_{22} ergibt sich $\dot{\mathbf{q}} = \frac{1}{2}(1+r_{22}-r_{11}-r_{33})^{-\frac{1}{2}}(r_{13}-r_{31}, r_{12}+r_{21}, 1+r_{22}-r_{11}-r_{33}, r_{32}+r_{23})^{\mathrm{T}}$ und ansonsten $\dot{\mathbf{q}} = \frac{1}{2}(1+r_{33}-r_{11}-r_{22})^{-\frac{1}{2}}(r_{21}-r_{12}, r_{13}+r_{31}, r_{32}+r_{23}, 1+r_{33}-r_{11}-r_{22})^{\mathrm{T}}$.

B.2 Rotationen und deren korrespondierende affine Verzerrungen

In Abschnitt 3.1.2 wird gezeigt, dass sich das Abbild oder Grundmuster $\mathcal{R}$ einer Teilmenge $\mathcal{S}$ einer Szene unter bestimmten Voraussetzung bzw. Näherungen eine affine Transformation **A** unterzieht, falls $\mathcal{S}$ mit einer Matrix **R** rotiert wird. Dies ist u. a. nur gültig falls sich $\mathcal{S}$ durch eine Ebene $m_x x + m_y y + z = 0$ beschreiben lässt. Nach Gleichung 3.7 lässt sich dann **A** wie folgt angeben:

$$\begin{pmatrix} r_{11}-m_x r_{13} & r_{12}-m_y r_{13} \\ r_{21}-m_x r_{23} & r_{22}-m_y r_{23} \end{pmatrix} \tag{B.2}$$

[1] wird auch als *Slerp* (engl.: spherical linear interpolation) bezeichnet.

Formal kann man dies als Funktion

$$\mathbf{G} : \mathcal{SO}_3 \times \mathbb{R} \times \mathbb{R} \rightarrow \mathcal{T}^+_{A,2} \tag{B.3}$$

auffassen, die einer Rotationsmatrix $\mathbf{R} \in \mathcal{SO}_3$ und zweier Ebenenparameter $m_x, my \in \mathbb{R}$ eine affine Matrix $\mathbf{A} = \mathbf{G}(\mathbf{R}, m_x, m_y) \in \mathcal{T}^+_{A,2}$ nach Gleichung B.2 zuordnet. In Abschnitt 4.3.1.1 wird konstruktiv gezeigt, das $\mathbf{G}(\cdot)$ nicht injektiv ist. Daher existiert auch keine (eindeutige) Umkehrung von $\mathbf{G}(\cdot)$. Im Folgenden werden nun einige Lemmata bewiesen, um die in dieser Arbeit vorgestellten Algorithmen kompakter darstellen zu können.

Lemma B.2.1 *Sei $\mathbf{R} \in \mathcal{SO}_3$ und $\mathcal{S}, \mathcal{S}'$ zwei durch die Ebenen $m_x x + m_y y + z = 0$ bzw. $m'_x x + m'_y y + z = 0$ beschreibbare Teilmengen der Szene, die durch $\mathcal{S}' = \mathbf{R}\mathcal{S}$ miteinander in Beziehung stehen. Dann gelten für die Ebenenparameter die Beziehungen*

$$\begin{aligned}
m_x &= (m'_x r_{11} + m'_y r_{21} + r_{31})(m'_x r_{13} + m'_y r_{23} + r_{33})^{-1} \\
m_y &= (m'_x r_{12} + m'_y r_{22} + r_{32})(m'_x r_{13} + m'_y r_{23} + r_{33})^{-1} \\
m'_x &= (m_x r_{11} + m_y r_{12} + r_{13})(m_x r_{31} + m_y r_{32} + r_{33})^{-1} \\
m'_y &= (m_x r_{21} + m_y r_{22} + r_{23})(m_x r_{31} + m_y r_{32} + r_{33})^{-1}
\end{aligned}$$

Beweis Bildet man die Koordinatenachsen über $(x', y', z')^{\mathrm{T}} = \mathbf{R}(x, y, z)^{\mathrm{T}}$ bzw. $(x, y, z)^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}(x', y', z')^{\mathrm{T}}$ auf das jeweilig andere System ab, setzt es dann in die Ebenengleichungen $m'_x x + m'_y y + z = 0$ bzw. $m_x x + m_y y + z = 0$ ein und bringt es dann auf die Form der jeweils anderen Ebenengleichung, erhält man durch Koeffizientenvergleich obiges Ergebnis. ■

Lemma B.2.2 *Sei $\mathbf{R}^\star, \mathbf{R}', \mathbf{R} \in \mathcal{SO}_3$ und $\mathbf{A}^\star, \mathbf{A}', \mathbf{A} \in \mathcal{T}^+_{A,2}$ mit $\mathbf{R}^\star = \mathbf{R}'\mathbf{R}$ und $\mathbf{A}^\star = \mathbf{G}(\mathbf{R}^\star, m_x, m_y)$, $\mathbf{A}' = \mathbf{G}(\mathbf{R}', m'_x, m'_y)$, $\mathbf{A} = \mathbf{G}(\mathbf{R}, m_x, m_y)$ sowie m'_x, m'_y über Lemma B.2.1 mit $m_x, m_y \in \mathbb{R}$ verknüpft, dann folgt daraus $\mathbf{A}^\star = \mathbf{A}'\mathbf{A}$.*

Beweis Sei s_{ij} das Skalarprodukt der Reihe i von Matrix $\mathbf{R}'$ mit der Spalte j von Matrix $\mathbf{R}$, dann erhält man für die Matrix $\mathbf{A}^\star = \mathbf{G}(\mathbf{R}'\mathbf{R}, m_x, m_y)$ die Form

$$\begin{pmatrix} s_{11} - m_x s_{13} & s_{12} - m_y s_{13} \\ s_{21} - m_x s_{23} & s_{22} - m_y s_{23} \end{pmatrix} \tag{B.4}$$

Die Matrix $\mathbf{A}'\mathbf{A} = \mathbf{G}(\mathbf{R}', m'_x, m'_y)\mathbf{G}(\mathbf{R}, m_x, m_y)$ dagegen hat die Form

$$\begin{pmatrix} s_{11} - m_x s_{13} + r'_{13} k & s_{12} - m_y s_{13} + r'_{13} l \\ s_{21} - m_x s_{23} + r'_{23} k & s_{22} - m_y s_{23} + r'_{23} l \end{pmatrix} \tag{B.5}$$

mit

$$\begin{aligned}
k &= m_x r_{33} - r_{31} + m'_x(m_x r_{13} - r_{11}) + m'_y(m_x r_{23} - r_{21}) \\
l &= m_x r_{33} - r_{32} + m'_x(m_x r_{13} - r_{12}) + m'_y(m_x r_{23} - r_{22})
\end{aligned}$$

Damit $\mathbf{A}^\star = \mathbf{A}'\mathbf{A}$ gilt, muss $k = l = 0$ allgemein gelten. Sei u_{ij} das Skalarprodukt von Spalte j mit Spalte i von Matrix $\mathbf{R}$, dann erhält man nach Ersetzen von m'_x bzw. m'_y mittels Lemma B.2.1 durch m_x bzw. m_y und Umsortieren:

$$k = m_x(u_{33} - u_{11}) - m_y u_{12} + m_x^2 u_{13} + m_x m_y u_{23} - u_{13} \tag{B.6}$$

$$l = m_y(u_{33} - u_{22}) - m_x u_{12} + m_y^2 u_{23} + m_x m_y u_{13} - u_{23} \tag{B.7}$$

Da $\mathbf{R} \in \mathcal{SO}_3$ folgt über deren Orthogonalität $u_{ij} = 1$ für $i = j$ bzw. $u_{ij} = 0$ sonst und damit $k = l = 0$.
■

Lemma B.2.3 *Sei* $\mathbf{R} \in \mathcal{SO}_3$ *und* $\mathbf{A} = \mathbf{G}(\mathbf{R}, n, m) \in \mathcal{T}^+_{A,2}$ *mit beliebigen* $n, m \in \mathbb{R}$. *Aus* $\mathbf{R} = \mathbf{R}_z$ *folgt* $\mathbf{A} \in \mathcal{SO}_2$, *die Umkehrung gilt dagegen nur für* $n = m = 0$.

Beweis Da $\mathbf{R}$ eine elementare Rotationsmatrix um die Z-Achse ist, folgt $r_{13} = r_{23} = 0$ und daraus $a_{ij} = r_{ij}$ für $i, j = 1, 2$. Weiterhin gilt $|\mathbf{R}| = r_{11}r_{22} - r_{12}r_{21} = 1 = a_{11}a_{22} - a_{12}a_{21} = |\mathbf{A}|$ und $r_{1i}^2 + r_{2i}^2 = r_{i1}^2 + r_{i2}^2 = 1 = a_{1i}^2 + a_{2i}^2 = a_{i1}^2 + a_{i2}^2$ für $i = 1, 2$, folglich $\mathbf{A} \in \mathcal{SO}_2$. Ist im umgekehrten Fall $n = m = 0$, gilt ebenfalls $a_{ij} = r_{ij}$ für $i, j = 1, 2$ und damit $r_{13} = r_{23} = 0$ bzw. $\mathbf{R} = \mathbf{R}_z$. Im allgemeinen Fall $n, m \neq 0$ gilt dies nicht, wie man sich an einem Beispiel klar machen kann. Selbst bei $\mathbf{A} = \mathbf{I}_{2\times 2}$ existiert nach Abschnitt 4.3.1.1 eine Rotation $\mathbf{R} \neq \mathbf{R}_z$, nämlich die Spiegelung der n, m-Ebene an der Null-Ebene $n = m = 0$. ∎

Lemma B.2.4 *Sei* $\mathbf{R} = \mathbf{R}_x\mathbf{R}_y \in \mathcal{SO}_3$ *und* $\mathbf{A}' = s\mathbf{A}$ *mit* $\mathbf{A} = \mathbf{G}(\mathbf{R}, 0, 0)$ *und* $s \in \mathbb{R}^+$, *dann lässt sich* $\mathbf{A}'$ *eindeutig in* s *und* $\mathbf{A}$ *zerlegen.*

Beweis Die Rotationsmatrix $\mathbf{R} = \mathbf{R}_x[\alpha]\mathbf{R}_y[\beta]$ hat die Form

$$\begin{pmatrix} c_y & 0 & s_y \\ s_x s_y & c_x & -s_x c_y \\ -c_x s_y & s_x & c_x c_y \end{pmatrix} \tag{B.8}$$

mit $c_x = \cos\alpha$, $c_y = \cos\beta$, $s_x = \sin\alpha$ und $s_y = \sin\beta$. Daraus leitet sich über $\mathbf{G}(\cdot)$ mit den Ebenenparametern $m_x = m_y = 0$ und dem Skalierungsparameter s die Matrix $\mathbf{A}'$ ab:

$$\begin{pmatrix} sc_y & 0 \\ ss_x s_y & sc_x \end{pmatrix} = \begin{pmatrix} e & 0 \\ f & g \end{pmatrix} \tag{B.9}$$

Quadriert man e, f und g, nutzt $\cos^2 + \sin^2 = 1$ und sortiert geeignet, erhält man eine in s doppelt quadratische Gleichung der Form

$$s^4 - s^2(e^2 + f^2 + g^2) + e^2 g^2 = 0 \tag{B.10}$$

Da per Definition $s > 0$ sind nur die zwei positiven Lösungen

$$s_\pm = \sqrt{\frac{e^2 + f^2 + g^2 \pm \sqrt{(e^2 + f^2 + g^2)^2 - 4e^2g^2}}{2}} \tag{B.11}$$

von Interesse. Im Folgenden wird nun gezeigt, dass nur die Lösung s_+ zu einer mit $\mathbf{R}$ konsistenten Matrix $\mathbf{A}$ führt. Dafür muss z.B. $e^2 s_\pm^{-2} = \cos^2\beta \leq 1$ bzw. $s_\pm^2 e^{-2} \geq 1$ gelten. Setzt man nun für $s_\pm$ die Lösungen, sowie für e, f und g die Cosinus- bzw. Sinus-Repräsentation aus Gl. B.9 ein und stellt geeignet um, ergibt sich

$$s_y^2 s_x^2 + c_x^2 - c_y^2 \pm \sqrt{(c_y^2 + s_y^2 s_x^2 + c_x^2)^2 - 4c_x^2 c_y^2} \geq 0 \tag{B.12}$$

Ersetzt man nun die Sinus durch Cosinus und nutzt die Binomischen Formeln auf die Terme unterhalb der Wurzel an, lässt sich diese eliminieren und man erhält

$$1 - 2c_y^2 + c_y^2 c_x^2 \pm 1 \mp c_y^2 c_x^2 \geq 0 \tag{B.13}$$

Dies führt bei der Lösung s_- durch $c_x^2 \geq 1$ im allgemeinen Fall auf einen Widerspruch, nur für $c_y^2 c_x^2 = 1$ ist $s_- = s_+$ und ebenfalls gültig. Sollte $e = 0$ sein, kann der Beweis auch analog mit g geführt werden. Sollte ebenfalls $g = 0$ sein, ergibt sich immer $s_- = 0$, so dass auch dort nur die Lösung s_+ zum Ziel führt. Durch $\mathbf{A} = s_+^{-1}\mathbf{A}'$ ist die Zerlegung eindeutig definiert. ∎

Anhang C

Kameramodellierung

Dieser Anhang beschäftigt sich mit der Modellierung von Kameras, um geometrische Beziehungen zwischen Punkten einer Szene und deren Projektionen in der Bildebene herstellen zu können. Zuerst wird im folgenden Abschnitt das Modell einer einzelnen Kamera vorgestellt. Der zweite Abschnitt beschäftigt sich dann mit der Modellierung eines Stereoaufbaus von zwei Kameras und dessen erweitere Möglichkeiten, insbesondere bei der Rekonstruktion von Szenenpunkten aus deren Abbildern.

C.1 Einzelkamera

Um Aussagen über eine Szene machen zu können die von einer Kamera beobachtet wird, benötigt man Modelle, die die Beziehungen zwischen wahrgenommenem Bild und realer Gestalt der Szene beschreiben. Das für diese Arbeit relevante Modell ist das geometrische Kameramodell, welches einen Punkt ${}^{\mathrm{W}}\mathbf{p} = (x_w, y_w, z_w)^{\mathrm{T}}$ der realen Welt, bezogen auf ein beliebiges Welt-Koordinatensystem W in Kamera-Koordinaten ${}^{\mathrm{C}}\mathbf{p} = \mathbf{p} = (x, y, z)^{\mathrm{T}}$ und schließlich in Bildkoordinaten $\mathbf{u} = (u, v)^{\mathrm{T}}$ überführt. Diese Transformation wird auch als Vorwärtsmodell bezeichnet, die inverse Problemstellung von Bildkoordinaten in Kamera- bzw. Welt-Koordinaten dagegen als Rückwärtsmodell. Das zweite Modell ist im allgemeinen schwieriger zu lösen, da es aufgrund des Informationsverlusts bei der Projektion ohne Kenntnis eines zusätzlichen Parameters, meist der Tiefenkoordinate z, unterbestimmt ist.

C.1.1 Idealisiertes Kameramodell

Das einfachste, idealisierte geometrische Modell ist das *Lochkameramodell*. Durch einen unendlich kleinen Punkt, dem Projektionszentrum **o** fallen Lichtstrahlen auf eine Bildebene I mit Abstand f, der Brennweite. Die optische Achse des Systems geht durch das Projektionszentrum und steht senkrecht auf I. Der Fußpunkt dieses Lots auf I wird als Durchstoßpunkt **c** bezeichnet und ist der Bezugspunkt des Bildkoordinatensystems. Das Projektionszentrum **o** ist der Ursprung des Kamera-Koordinatensystems. Dessen z-Achse zeigt im Allgemeinen entlang der optischen Achse von der Bildebene weg. Die x- und y-Achse stehen senkrecht auf der z-Achse, spannen die Fokalebene $\mathcal{F}$ auf und bilden ein Rechtssystem. Das Lochkameramodell ist zur besseren Anschauung in Abb. C.1 dargestellt. Für einen Punkt $\mathbf{p} = (x, y, z)^{\mathrm{T}}$ in Kamera-Koordinaten ergibt sich der zugehörige Bildpunkt $\mathbf{u} = (u, v)^{\mathrm{T}}$ mittels Strahlensatz wie folgt:

$$u = -f\frac{x}{z} \tag{C.1}$$

$$v = -f\frac{y}{z} \tag{C.2}$$

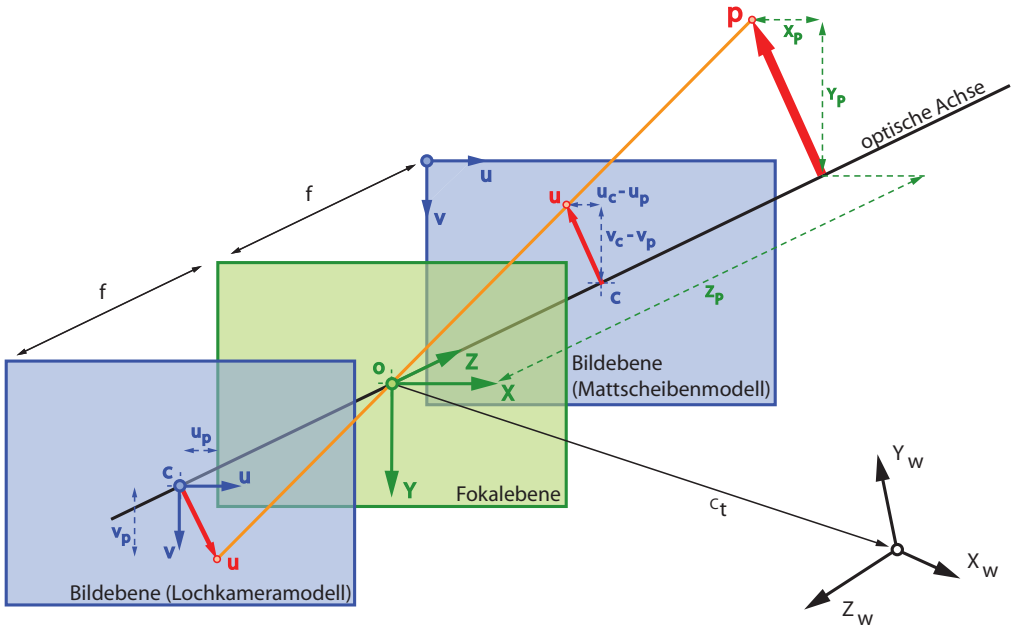

Abbildung C.1: Lochkameramodell mit Koordinatenursprung am Durchstoßpunkt **o** und Mattscheibenmodell mit verschobenem Koordinatenursprung in der linken oberen Bildecke. Dargestellt ist ein Punkt $\mathbf{p} = (x_\mathbf{p}, y_\mathbf{p}, z_\mathbf{p})^\mathrm{T}$ in Kamera-Koordinaten und sein zugehöriger Bildpunkt $\mathbf{u_p} = (u_\mathbf{p}, v_\mathbf{p})^\mathrm{T}$ in der Bildebene der Lochkamera und des Mattscheibenmodells, definiert durch den Schnittpunkt des Lichtstrahls $\overline{\mathbf{po}}$ (orange) mit der jeweiligen Bildebene.

Verwendet man das physikalisch motivierte Modell der Lochkamera, steht das erhaltene Bild auf dem Kopf und man muss es anschließend spiegeln. Alternativ dazu kann man das mathematisch äquivalente *Mattscheibenmodell* verwenden, mit an der z-Achse positiv angetragener Brennweite f und erhält sofort das korrekt dargestellte Bild. Es ist ebenfalls in Abb. C.1 eingezeichnet und wird im weiteren Verlauf verwendet.

Um zu einem flexibleren Modell zu gelangen, werden die grundlegenden Gl. C.1 und C.2 erweitert. Zum einen wird der Koordinatenursprung der Bildebene beliebig modelliert und in dieser Arbeit, so wie allgemein üblich, in die linke obere Ecke gelegt. Der Durchstoßpunkt hat dann die Bildkoordinaten $\mathbf{c} = (u_\mathbf{c}, v_\mathbf{c})^\mathrm{T}$. Weiterhin kann die u- bzw. v-Achse beliebig mit den Parametern s_u und s_v skaliert sein. Sie dienen u. a. der Einheiten-Überführung von Kamera-Koordinaten in Bildkoordinaten, z. B. von Meter in Pixel. Als letztes kann mit einem Parameter s eine Scherung zwischen der u- und v-Achse modelliert werden. Scherwerte ungleich 0 treten nur in sehr seltenen Fällen auf, z. B. u. U. wenn eine Szene nur indirekt über einen Spiegel beobachtet wird. Mit $f_u = s_u f$ und $f_v = s_v f$ folgt dann für das Mattscheibenmodell:

$$u = f_u \frac{x + sy}{z} + u_\mathbf{c} \tag{C.3}$$

$$v = f_v \frac{y}{z} + v_\mathbf{c} \tag{C.4}$$

Um Welt-Koordinaten W in Kamera-Koordinaten C zu überführen, benötigt man die Transformationsparameter einer RT ${}^\mathrm{C}\check{\mathbf{H}}_\mathrm{W}$ (vgl. Anhang B.1), normalerweise eine Rotationsmatrix ${}^\mathrm{C}\mathbf{R}_\mathrm{W}$, die die Orientierungsänderungen beschreibt und einen Translationsvektor ${}^\mathrm{C}\mathbf{t}$, der von dem Ursprung von C auf den Ursprung von W zeigt:

$${}^\mathrm{C}\mathbf{p} = {}^\mathrm{C}\mathbf{R}_\mathrm{W}\,{}^\mathrm{W}\mathbf{p} + {}^\mathrm{C}\mathbf{t}$$

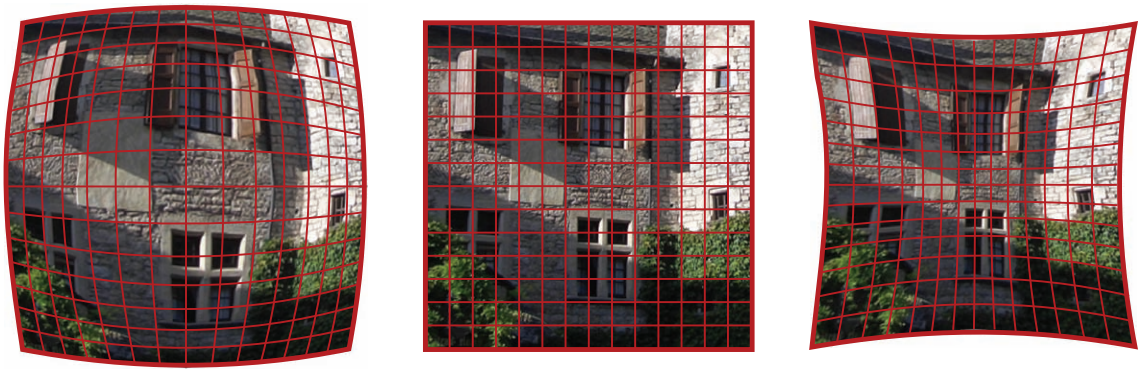

Abbildung C.2: Unverzerrtes Bild (Mitte), tonnenförmige Verzerrung (links) wie sie oft durch Weitwinkelkameras entsteht und kissenförmige Verzerrung (rechts), jeweils mit überlagerter Verzerrungsmaske.

Für eine kompakte Darstellung benötigt man homogene Koordinaten $\check{\mathbf{u}} = (ru, rv, r)^{\mathrm{T}}$ wobei r auf 1 normiert wird. Fasst man alle Parameter geeignet in Matrizen zusammen, erhält man das ideale Mattscheibenmodell[1] für ${}^{\mathrm{W}}\check{\mathbf{p}} = (x_w, y_w, z_w, 1)^{\mathrm{T}}$ zu $\check{\mathbf{u}}_{\mathbf{p}} = \check{\mathbf{K}}\mathbf{M}\,{}^{\mathrm{W}}\check{\mathbf{p}}$ bzw. ausführlich:

$$\underbrace{\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}}_{\check{\mathbf{u}}_{\mathbf{p}}} = \underbrace{\begin{pmatrix} f_u & s & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{pmatrix}}_{\check{\mathbf{K}}} \underbrace{\left(\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{array}\right)}_{\mathbf{M}=({}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}}|{}^{\mathrm{C}}\mathbf{t})} \underbrace{\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}}_{{}^{\mathrm{W}}\check{\mathbf{p}}}$$

$\check{\mathbf{K}}$ wird dabei als Kameramatrix bezeichnet und enthält die *intrinsischen* Parameter, d. h. alle Parameter die von der Kamera beeinflusst werden, nicht aber von dem Szenenaufbau. Im Gegensatz dazu enthält $\mathbf{M}$ die *extrinsischen* Parameter, die unabhängig von der tatsächlich verwendeten Kamera, dafür aber Teil der Szenenmodellierung sind. Man beachte, dass diese Matrix durch die Normierung des Faktors r auch die Projektion $\bar{u} = \frac{x}{z}$ und $\bar{v} = \frac{y}{z}$ mit Einheitsbrennweite 1 durchführt. $\bar{\mathbf{u}} = (\bar{u}, \bar{v})^{\mathrm{T}}$ wird auch als (unverzerrter) normalisierter Bildpunkt bezeichnet. Dieser kann ebenfalls mit der nichthomogenen 2×2 Kameramatrix $\mathbf{K}$ (oberen, linken Einträge von $\check{\mathbf{K}}$) über $\mathbf{u} = \mathbf{K}\bar{\mathbf{u}} + \mathbf{c}$ in die übliche Darstellung umgewandelt werden.

Da es sich bei dem Vorwärtsmodell um eine Projektion handelt, existiert keine eindeutige Umkehrung, da für einen Bildpunkt $\mathbf{u}$ nur der zugehörige Sichtstrahl $\mu\check{\mathbf{K}}^{-1}\check{\mathbf{u}}$, $\mu \in \mathbb{R}_+$ (in C) angegeben werden kann. μ entspricht dabei der unbekannten Tiefe z des zugehörigen Punktes in Kamera-Koordinaten. Ist diese bekannt, lässt sich ${}^{\mathrm{C}}\mathbf{p} = z\check{\mathbf{K}}^{-1}\check{\mathbf{u}}$ allerdings leicht berechnen. Diese Methode wird in dieser Arbeit auch als Rückprojektion bezeichnet, da sie eine Projektion zurück in die Szene abbildet. Man beachte, dass $\check{\mathbf{K}}^{-1}\check{\mathbf{u}}$ zwar die normalisierte, homogene Darstellung $\check{\bar{\mathbf{u}}}$ zurück liefert, durch die Multiplikation mit z der Punkt allerdings in die nichthomogene Darstellung ${}^{\mathrm{C}}\mathbf{p}$ im Kamera-Koordinatensystem überführt wird.Die Welt-Koordinaten erhält man dann durch ${}^{\mathrm{W}}\check{\mathbf{p}} = {}^{\mathrm{C}}\check{\mathbf{H}}_{\mathrm{W}}^{-1}\,{}^{\mathrm{C}}\check{\mathbf{p}}$.

C.1.2 Verzerrungen

Das ideale Lochkamera- bzw. Mattscheibenmodell hat in der Praxis den Nachteil, dass durch den sehr kleinen Punkt im Projektionszentrum der Kamera nur eine sehr kleine Lichtmenge fällt. Um

[1] In der Literatur wird dieses Modell im allgemeinen ebenfalls als Lochkameramodell bezeichnet.

lichtstarke optische Systeme realisieren zu können, werden deshalb in der Praxis Linsensysteme verwendet um eine größere Lichtmenge zu bündeln. Man handelt sich dabei aber mehrere Probleme ein, u. a. einen begrenzten Schärfebereich, Abschattungen an den Linsenrändern und für das geometrische Kameramodell bei der Abbildung nicht zu vernachlässigende Verzerrungen[2] (vgl. Abb. C.2).

Für das allgemeine Verzerrmodell, das in (Ope09) verwendet wird, benötigt man die normalisierten unverzerrten Bildkoordinaten $\breve{\bar{\mathbf{u}}} = (\bar{u}, \bar{v}, 1)^{\mathrm{T}} = \mathbf{M}\,{}^{\mathrm{W}}\breve{\mathbf{p}}$, die unabhängig von den intrinsischen Parametern der Kameramatrix $\mathbf{K}$ sind und aus den idealen Gleichungen des Mattscheibenmodells gewonnen werden. Die verzerrten Bildkoordinaten $\breve{\mathbf{u}} = (u, v, 1)^{\mathrm{T}}$ erhält man dann mittels einer nichtlinearen Verzerrungsfunktion $\mathbf{d}(\cdot) : \mathbb{R}^2 \to \mathbb{R}^2$, wobei $\mathbf{u} = \mathbf{d}(\bar{\mathbf{u}})$ die normale und $\breve{\mathbf{u}} = \breve{\mathbf{d}}(\breve{\bar{\mathbf{u}}})$ die homogene Darstellung repräsentiert. $\mathbf{d}(\cdot)$ ist definiert durch:

$$u = \bar{u}(1 + \kappa_1 r^2 + \kappa_2 r^4) + 2\rho_1 \overline{uv} + \rho_2(r^2 + \bar{u}^2) \tag{C.5}$$

$$v = \bar{v}(1 + \kappa_1 r^2 + \kappa_2 r^4) + 2\rho_2 \overline{uv} + \rho_1(r^2 + \bar{v}^2) \tag{C.6}$$

mit dem Radius $r = \sqrt{\bar{u}^2 + \bar{v}^2}$ bezogen auf den Mittelpunkt $\mathbf{c} = (u_c, v_c)^{\mathrm{T}}$. κ_1 und κ_2 werden als radiale Verzerrungsparameter bezeichnet, da sie auf die verzerrten Bildpunkte nur abhängig von dem Radius wirken. Sie haben dominanten Einfluss auf die Verzerrung und sind insbesondere bei Weitwinkelobjektiven nicht zu vernachlässigen. Die tangentialen Parameter ρ_1 und ρ_2 spielen für optische Systeme eine untergeordnete Rolle, sie werden daher bei (Zha00) durch Setzen auf $\rho_1 = \rho_2 = 0$ vernachlässigt. Da die Verzerrungsparameter unabhängig von der Szene sind, werden sie ebenfalls zu den intrinsischen Parametern gezählt.

Ein alternatives Verzerrmodell wird in (MVT10) verwendet. Es hat den Vorteil, dass sich die Umkehrung $\mathbf{d}^{-1}(\cdot)$ analytisch angeben lässt und nicht wie in (Ope09) numerisch iterativ bestimmt werden muss. Es benötigt nur einen Verzerrungsparameter κ, kann allerdings auch nur radiale Linsenverzerrungen modellieren:

$$\mathbf{u} = \mathbf{d}(\bar{\mathbf{u}}) = \frac{1}{1 + \kappa|\bar{\mathbf{u}}|^2}\bar{\mathbf{u}} \tag{C.7}$$

$$\bar{\mathbf{u}} = \mathbf{d}^{-1}(\mathbf{u}) = \frac{2}{1 + \sqrt{1 - 4\kappa|\mathbf{u}|^2}}\mathbf{u} \tag{C.8}$$

In dieser Arbeit wird ebenfalls die lineare Approximation $\mathbf{d}(\bar{\mathbf{u}} + \bar{\mathbf{u}}_\Delta) \approx \mathbf{d}(\bar{\mathbf{u}}) + \mathbf{D}_{\bar{\mathbf{u}}}\bar{\mathbf{u}}_\Delta + \ldots$ der Verzerrung an der Stelle $\bar{\mathbf{u}}$ verwendet. Dafür wird die Jacobi-Matrix $\mathbf{D}_{\bar{\mathbf{u}}}$ benötigt, die sich für (MVT10) wie folgt ergibt:

$$\mathbf{D}_{\bar{\mathbf{u}}} = \frac{1}{(1 + \kappa|\bar{\mathbf{u}}|^2)^2}\begin{pmatrix} 1 + \kappa(\bar{v}^2 - \bar{u}^2) & -2\kappa\overline{uv} \\ -2\kappa\overline{uv} & 1 + \kappa(\bar{u}^2 - \bar{v}^2) \end{pmatrix} \tag{C.9}$$

Zusammenfassend ergibt sich das Mattscheibenmodell ${}^{\mathrm{W}}\mathbf{h} : \mathbb{R}^3 \to \mathbb{R}^2$ mit Verzerrung zu

$$\breve{\mathbf{u}} = {}^{\mathrm{W}}\mathbf{h}(\breve{\mathbf{p}}) = \mathbf{K}\breve{\mathbf{d}}(\mathbf{M}\,{}^{\mathrm{W}}\breve{\mathbf{p}}) \tag{C.10}$$

Zur Vereinfachung wird in dieser Arbeit an manchen Stellen auf eine formal korrekte homogene Darstellung $\breve{\mathbf{u}} = \mathbf{h}(\breve{\mathbf{p}})$ verzichtet und die normale Schreibweise $\mathbf{u} = \mathbf{h}(\mathbf{p})$ verwendet. Weiterhin bezeichnet $\mathbf{h}(\cdot) = {}^{\mathrm{C}}\mathbf{h}(\cdot)$ das gleiche Modell, allerdings werden diesmal Kamera-Koordinaten ${}^{\mathrm{C}}\mathbf{p}$ erwartet und die extrinsischen Parameter ignoriert.

Das inverse Modell $\mathbf{h}^{-1} : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}^3$ lässt sich nur analytisch geschlossen ausdrücken, falls die Verzerrungsfunktion $\mathbf{d}(\cdot)$ eine Umkehrung $\mathbf{d}^{-1}(\cdot)$ besitzt und die Tiefe z in Kamera-Koordinaten bekannt ist. Dann ergibt sich $\mathbf{h}^{-1}(\cdot)$ ohne extrinsische Parameter zu

$${}^{\mathrm{C}}\mathbf{p} = \mathbf{h}^{-1}(\breve{\mathbf{u}}, z) = z\breve{\mathbf{d}}^{-1}(\breve{\mathbf{K}}^{-1}\breve{\mathbf{u}}) \tag{C.11}$$

[2] engl.: distortions

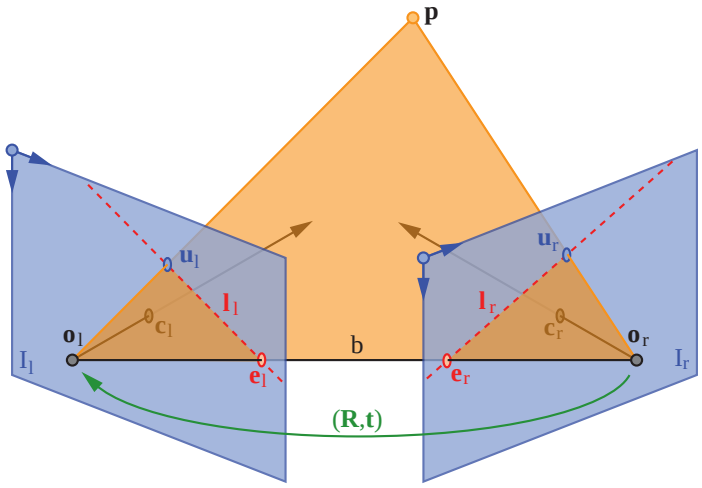

Abbildung C.3: Stereokamerasystem mit eingezeichneter Epipolargeometrie (rot u. orange).

Man beachte, dass $\check{\mathbf{d}}(\cdot)$ zwar wiederum die normalisierte, homogene Darstellung $\check{\mathbf{u}}$ zurück liefert, durch die Multiplikation mit z der Punkt allerdings in ${}^{\mathrm{C}}\mathbf{p}$ wird. Den Punkt in Welt-Koordinaten erhält man dann durch ${}^{\mathrm{W}}\mathbf{p} = {}^{\mathrm{W}}\check{\mathbf{H}}_{\mathrm{C}}\,{}^{\mathrm{C}}\mathbf{p}$.

C.2 Stereokamera

Stereo-Sehen bezeichnet die Möglichkeit eines Systems aus zwei Bildern die Tiefe von Bildpunkten zu bestimmen, um über die Rückprojektion Informationen über die dreidimensionale geometrische Struktur einer Szene zu gewinnen. Beim Stereo-Sehen repräsentieren die Bilder zeitgleiche Aufnahmen der Szene aus verschiedenen Blickwinkeln. Diese Ausführung bezieht sich auf eine Stereokamera, d. h. einen binokularen Aufbau mit zwei Kameras, deren Bilder im Weiteren als $\mathrm{i}_l(\cdot)$ und $\mathrm{i}_r(\cdot)$ bezeichnet werden. Alle Parameter bzw. Größen der Einzelkameras werden mit einem Index l für die linke bzw. r für die rechte Kamera versehen. Ein schematischer Aufbau ist in Abb. C.3, eine praktische Umsetzung in 1.2 zu sehen.

Beim Stereo-Sehen gibt es allgemein zwei Probleme zu lösen:

Korrespondenzfindung Es müssen in beiden Ansichten korrespondierende Punkte einander zugeordnet werden. Je nach Verfahren erhält man entweder ein dünnes oder dichtes Disparitätsfeld, mit dem Versatz der Punkte in den verschiedenen Ansichten.

Rekonstruktion Sind die Korrespondenzen gefunden, muss aus dem Disparitätsfeld die 3D-Information der Szene gewonnen werden. Die Ausgabe ist je nach Eingabe meist ein dünnes oder dichtes Tiefenbild.

In den folgenden Unterabschnitten wird zuerst das Korrespondenzproblem angesprochen und die Epipolarbedingung sowohl algebraisch als auch geometrisch vorgestellt. Im darauf folgenden Abschnitt werden dann Ansätze zur Rekonstruktion behandelt. Auf konkrete Verfahren wird nicht eingegangen wird, da sie für das Verständnis dieser Arbeit im Gegensatz zu den hier vorgestellten Grundlagen keine Rolle spielen. Sie lassen sich z. B. in (SS02) nachschlagen.

C.2.1 Korrespondenzproblem

Das Korrespondenzproblem beim Stereo-Sehen unterscheidet sich in den Grundzügen nicht von anderen Korrespondezproblemen wie z. B. dem des optischen Flusses. In beiden Fällen müssen die korrespondierenden Punkte oder Verschiebungen in verschiedenen Ansichten der Szene gefunden werden. Beim Stereo-Sehen unterscheiden sich diese allerdings nicht zeitlich sondern nur räumlich. Damit der gleiche Aufnahme-Zeitpunkt gewährleistet ist, müssen die Kameras des Stereo-Systems synchronisiert sein. Dies ist insbesondere bei einen bewegtem Stereo-System wie bei dieser Arbeit wichtig, da ansonsten die Basisbreite b des Systems künstlich verändert wird und die Rekonstruktion Fehler liefert. Im weiteren Verlauf wird ein korrespondierendes Punktepaar in $\mathrm{i}_l(\cdot)$ und $\mathrm{i}_r(\cdot)$ bzgl. des Szenenpunktes $\mathbf{p}$ als $(\mathbf{u}_l, \mathbf{u}_r)$ bezeichnet. Als Referenz wird o. B. d. A. das linke Bild $\mathrm{i}_l(\cdot)$ verwendet.

Zur Lösung des Korrespondenzproblems eignet sich für dünne Disparitätsfelder der merkmalsbasierte Ansatz, für den u. a. die in Kap. 3 vorgestellten lokalen Regionen entwickelt wurden. Weiterhin steht bei einem fixen Stereo-Aufbau mit der Epipolarbedingung ein mächtiges Werkzeug zur Verfügung, welches es ermöglicht, die Suche des Korrespondenzproblems um eine Dimension zu verringern. Es soll im folgenden vorgestellt werden.

C.2.1.1 Epipolarbedingung

Die Epipolarbedingung kann sowohl geometrisch als auch algebraisch hergeleitet werden. Beide Darstellungen haben ihre Vorzüge und werden deshalb einzeln betrachtet. Die folgende Ausführung beruht auf (Zha98).

Algebraische Darstellung Gegeben seien zwei Kameras, mit links und rechts bezeichnet, die auf dem im Abschnitt C.1 vorgestellten Mattscheibenmodell ohne Verzerrungen beruhen. Die beiden Kamera-Koordinatensysteme lassen sich durch eine Starrkörpertransformation ${}^{\mathrm{R}}\check{\mathbf{H}}_{\mathrm{L}}$, d. h. eine Rotationsmatrix $\mathbf{R}$ und einem Translationsvektor $\mathbf{t}$ ineinander überführen. Für einen Szenenpunkt $\mathbf{p}$, dargestellt in den beiden Kamera-Koordinatensystemen als ${}^{\mathrm{L}}\mathbf{p}$ und ${}^{\mathrm{R}}\mathbf{p}$ gelte dann die Beziehung:

$$ {}^{\mathrm{R}}\mathbf{p} \quad = \quad \mathbf{R}\,{}^{\mathrm{L}}\mathbf{p} + \mathbf{t} \tag{C.12} $$

Führt man auf diese Gleichung das Kreuzprodukt mit $\mathbf{t}$ aus, transponiert die Gleichung und nimmt das Skalarprodukt mit $\mathbf{R}\,{}^{\mathrm{L}}\mathbf{p}$, folgt daraus:

$$ ({}^{\mathrm{R}}\mathbf{p} \times \mathbf{t})^{\mathrm{T}} \mathbf{R}\,{}^{\mathrm{L}}\mathbf{p} = \underbrace{(\mathbf{R}\,{}^{\mathrm{L}}\mathbf{p} \times \mathbf{t})^{\mathrm{T}} \mathbf{R}\,{}^{\mathrm{L}}\mathbf{p}}_{=0} + (\underbrace{\mathbf{t} \times \mathbf{t}}_{=0})^{\mathrm{T}} \mathbf{R}\,{}^{\mathrm{L}}\mathbf{p} = 0 \tag{C.13} $$

Führt man nun den antisymmetrischen Kreuzproduktoperator

$$ [\mathbf{t}]_{\times} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} $$

für $\mathbf{t} = (t_x, t_y, t_z)^{\mathrm{T}}$ ein, erhält man:

$$ {}^{\mathrm{R}}\mathbf{p}^{\mathrm{T}} \underbrace{[\mathbf{t}]_{\times}^{\mathrm{T}} \mathbf{R}}_{\mathbf{E}}\,{}^{\mathrm{L}}\mathbf{p} = {}^{\mathrm{R}}\mathbf{p}^{\mathrm{T}} \mathbf{E}\,{}^{\mathrm{L}}\mathbf{p} = 0 \tag{C.14} $$

$\mathbf{E}$ wird als *essentielle* Matrix bezeichnet. Sie beinhaltet alle relevanten Daten um einen Szenenpunkt in beiden Kamera-Koordinatensystemen zu verknüpfen. Sie hängt nur von $\mathbf{R}$ und $\mathbf{t}$ ab und gilt für

alle Szenenpunkte. Wendet man auf obige Formel das Mattscheibenmodell $\check{\mathbf{u}} = \check{\mathbf{K}}\check{\mathbf{p}}$ (vgl. Abschnitt C.1.1) an, erhält man die Epipolarbedingung algebraisch dargestellt:

$$\check{\mathbf{u}}_r^{\mathrm{T}} \underbrace{\check{\mathbf{K}}_r^{-\mathrm{T}} [\mathbf{t}]_\times^{\mathrm{T}} \mathbf{R} \check{\mathbf{K}}_l^{-1}}_{\mathbf{F}} \check{\mathbf{u}}_l = \check{\mathbf{u}}_r^{\mathrm{T}} \mathbf{F} \check{\mathbf{u}}_l = 0 \tag{C.15}$$

Die *Fundamentalmatrix* **F** verknüpft ähnlich wie die essentielle Matrix ein korrespondierendes Punktepaar $(\mathbf{u}_l, \mathbf{u}_r)$ miteinander. **F** ist sehr mächtig mächtig, da sie unabhängig von den Bildpunkten nur vom Stereo-Aufbau sowie den ex- und intrinsischen Parametern beider verwendeten Kameras abhängt. Weiterhin kann sie ohne deren Kenntnis nur aus einer Menge von korrespondieren Punktepaaren geschätzt werden[3]. Die Fundamentalmatrix besteht aus 3×3 Elementen. Da allerdings durch $\det([\mathbf{t}]_\times) = 0$ auch $\det(\mathbf{E}) = \det(\mathbf{F}) = 0$ folgt und **F** durch die homogene Gleichung mittels des Skalierungsfaktors λ einen Freiheitsgrad besitzt, genügen 7 Punktepaare zur Ermittlung von **F**.

Neben der wenig intuitiven algebraischen Form der Epipolarbedingung gibt es eine sehr anschauliche geometrische Darstellung, die im nächsten Abschnitt vorgestellt wird.

Geometrische Darstellung Die geometrische Darstellung der Epipolarbedingung wird auch als *Epipolargeometrie* bezeichnet. Betrachtet man in Abb. C.3 das Dreieck $\overline{\mathbf{p}\mathbf{o}_l\mathbf{o}_r}$ aus beliebigen Szenenpunkt **p** und der Basisachse **b** des Stereokamerasystems, so fällt auf, dass es die Bildebenen I_l und I_r in zwei Geraden $\mathbf{l}_l$ und $\mathbf{l}_r$ schneidet, falls **p** im Sichtbereich beider Kameras liegt. $\mathbf{l}_l$ und $\mathbf{l}_r$ werden als *Epipolargeraden* bezeichnet, sie entsprechen der Projektion des Sichtstrahls $\overline{\mathbf{p}\mathbf{o}_r}$ der rechten Kamera in das Bild I_l der linken Kamera bzw. umgekehrt. Da die Grundlinie $\overline{\mathbf{o}_l\mathbf{o}_r}$ des Dreiecks für alle Szenenpunkte dieselbe ist, schneiden sich alle möglichen Epipolargeraden eines Bildes in einem gemeinsamen Punkt, dem *Epipol* **e**. Er entspricht der Projektion des Projektionszentrums der einen Kamera in die Bildebene der jeweils anderen Kamera.

Für das Korrespondenzproblem haben diese geometrischen Beziehungen einen angenehmen Effekt. Da der korrespondierende Punkt $\mathbf{u}_r$ im rechten Bild eines bekannten Bildpunktes $\mathbf{u}_l$ im Referenz-Bild auf der Epipolargeraden $\mathbf{l}_r$ liegt, reduziert sich die Korrespondenzsuche auf eine Dimension entlang $\mathbf{l}_r$. Dies entspricht der algebraischen Form der Epipolarbedingung $\check{\mathbf{u}}_r^{\mathrm{T}} \mathbf{F} \check{\mathbf{u}}_l = 0$, denn es gilt:

$$\begin{aligned} \mathbf{F}\check{\mathbf{u}}_l = \mathbf{l}_r \quad &\text{bzw.} \quad \mathbf{F}^{\mathrm{T}}\check{\mathbf{u}}_r = \mathbf{l}_l \\ \mathbf{F}\check{\mathbf{e}}_l = \mathbf{0} \quad &\text{bzw.} \quad \mathbf{F}^{\mathrm{T}}\check{\mathbf{e}}_r = \mathbf{0} \end{aligned}$$

Die Punkte $\mathbf{u} = (u,v)^{\mathrm{T}}$ einer Gerade $\mathbf{l} = (a,b,c)^T$ werden dabei durch die Gleichung $au + bv + c$ oder kompakt durch $\mathbf{l}^{\mathrm{T}}\check{\mathbf{u}} = 0$ beschrieben (vgl. (Zha98)).

Für eine leichte Auswertung der Epipolarbedingung kann man die Bildebenen I_l und I_r günstig transformieren. Die Vorgehensweise wird als Rektifizierung bezeichnet, deren Idee im Folgenden vorgestellt wird.

C.2.1.2 Rektifizierung

Unter Rektifizierung versteht man die Transformation der Bildebenen, so dass sie koplanar und ihre u-Achse parallel zur Basisachse **b** des Stereo-Systems ausgerichtet sind (vgl. Abb. C.4). Dadurch wandern die Epipole auf der Basisachse ins Unendliche und alle Epipolargeraden richten sich parallel zur Basisachse aus. Weiterhin fallen korrespondierende Epipolargeraden $\mathbf{l}_l$ und $\mathbf{l}_r$ zusammen, so dass für alle korrespondierenden Punktepaare $(\mathbf{u}_l, \mathbf{u}_r)$ die Beziehung $v_l = v_r$ gilt. Zur Korrespondenzsuche muss daher nur horizontal entlang der u-Koordinate von $\mathbf{u}_r$ gesucht werden, was einer erheblichen algorithmischen Vereinfachung entspricht.

[3] wird ausschließlich **F** zur Rekonstruktion benutzt, spricht man auch von *unkalibriertem Stereo*

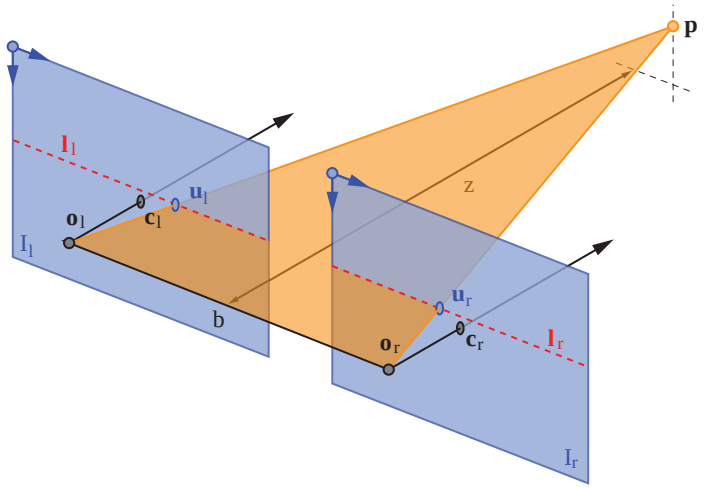

Abbildung C.4: Rektifiziertes Stereokamerasystem mit horizontalen Epipolargeraden (rot). Die Epipole befinden sich im unendlichen.

C.2.2 Rekonstruktion

Unter Rekonstruktion versteht man die Gewinnung der 3D-Information $\mathbf{p}$ oder Tiefe z aus einem korrespondierenden Punktepaar $(\mathbf{u}_l, \mathbf{u}_r)$ bzgl. eines Stereobildes. Den zugehörigen Szenenpunkt $\mathbf{p}$ erhält man prinzipiell sehr einfach als Schnittpunkt der beiden Sichtstrahlen, definiert durch das jeweilige Projektionszentrum und dessen Bildpunkt. In der Praxis werden sich die beiden Strahlen aufgrund von Rauschen und Berechnungsungenauigkeiten allerdings nicht schneiden, sondern windschief zueinander liegen. Ein geometrischer Rekonstruktionsansatz wäre dann z. B. den Mittelpunkt des Lots der einen auf die andere Gerade (d. h. der kürzeste Abstand) als den Szenenpunkt zu wählen. Eine algebraische Möglichkeit besteht darin, die Geradengleichungen der beiden Sichtstrahlen gleich zu setzen und in ein lineares Gleichungssystem über zu führen.

Für ein rektifiziertes Stereokamerasystem existiert eine einfache Formel zur Rekonstruktion. Sei $d = v_r - v_l$ der horizontale Versatz eines korrespondierenden Punktepaars $(\mathbf{u}_l, \mathbf{u}_r)$ bezogen auf normalisierte Bildkoordinaten, dann lässt sich daraus die Tiefe z des zugehörigen Szenenpunktes bezogen auf das rektifizierte Kamera-Koordinatensystem wie folgt ermitteln (vgl. Abb. C.4):

$$z = f\frac{b}{d} \tag{C.16}$$

f entspricht dabei der Brennweite nach der Rektifizierung, die dann für beide Bilder dieselbe ist. Der horizontale Versatz d wird als Disparität bezeichnet und verhält sich umgekehrt proportional zur Tiefe des Szenenpunkts in Kamera-Koordinaten. Mit der Bekanntheit von z kann mittels des inversen Mattscheibenmodells (mit rektifizierten Parametern) die Position von $\mathbf{p} = \mathbf{h}_l^{-1}(\mathbf{u}_l, z)$ berechnet werden.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[AAD06] AZAD, P. ; ASFOUR, T. ; DILLMANN, R.: Combining Apperance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China*, 2006, S. 5339–5344

[AAD07] AZAD, P. ; ASFOUR, T. ; DILLMANN, R.: Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), USA*, 2007, S. 919–924

[AG97] AMIT, Y. ; GEMAN, D.: Shape Quantization and Recognition with Randomized Trees. In: *Neural Computation* (1997), Nr. 9:7, S. 1545–1588

[AHB87] ARUN, K. S. ; HUANG, T. S. ; BLOSTEIN, S. D.: Least-Squares Fitting of Two 3-D Point Sets. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence PAMI* 9 (1987), S. 698–700

[Alt62] ALT, F. L.: Digital Pattern Recognition by Moments. In: *Journal of the ACM (JACM)* 9:2 (1962), S. 240–258

[ANR98] ABDALLAH, S. M. ; NEBOT, E. M. ; RYE, D. C.: Object recognition and orientation via Zernike moments. In: *Asian conference on computer vision (ACCV), Chine, Volume 1352*, 1998, S. 386–393

[Aza08] AZAD, P.: *Visual Perception for Manipulation and Imitation in Humanoid Robots.*, Universität Karlsruhe (TH), Diss., 2008

[BAGC05] BRUSCO, N. ; ANDREETTO, M. ; GIORGI, A. ; CORTELAZZO, G.: 3D registration by textured spin-images. In: *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, 2005

[Bal81] BALLARD, D. H.: Generalizing the Hough Transform to Detect Arbitrary Shapes. In: *Pattern Recognition* 13(2) (1981), S. 111 – 122

[Bau00] BAUMBERG, A.: Reliable Feature Matching AcrossWidely Separated Views. In: *Proc. CVPR* (2000), S. 774–781

[Bie87] BIEDERMAN, I.: Recognition-by-components: A theory of human image understanding. In: *Psychological review* 94 (1987), Nr. 2, S. 115–147

[BKK96] BERCHTHOLD, S. ; KEIM, D. A. ; KRIEGEL, H.-P.: The X-Tree: An Index Structure for High-Dimensional Data. In: *Proceedings of the 22th International Conference on Very Large Data Bases (VLDB).* India, 1996, S. 28–39

[BL95] BLAIS, G. ; LEVINE, M.: Registering Multiview Range Data to Create 3D Computer Objects. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 17(8) (1995), S. 820–824

[BL99] BEIS, S. J. ; LOWE, D. G.: Indexing Without Invariants in 3D Object Recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), Nr. 10, S. 1000–1015

[BL02] BROWN, M. ; LOWE, D. G.: Invariant features from interest point groups. In: *British Machine Vision Conference, Cardiff, Wales*, 2002, S. 656–665

[BM92] BESL, P. ; MCKAY, N.: A Method for Registration of 3-D Shapes. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 14(2) (1992), S. 239 – 256

[BMP02] BELONGIE, S. ; MALIK, J. ; PUZICHA, J.: Shape matching and object recognition using shape contexts. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), S. 509–522

[BPPP98] BOROTSCHNIG, H. ; PALETTA, L. ; PRANTL, M. ; PINZ, A: Active Object Recognition in Parametric Eigenspace. In: *Proc. British Machine Vision Conference (BMVC)*, 1998, S. 629–638

[Bro99] BRONSTEIN, Ilja N.: *Taschenbuch der Mathematik*. 4. überarb. u. erw. Aufl. Deutsch (Harri), 1999. – ISBN 3817120141

[BS01] BURKHARDT, H. ; SIGGELKOW, S.: Invariant Features in Pattern Recognition - Fundamentals and Applications. In: *Nonlinear Model-Based Image/Video Processing and Analysis* (2001), S. 269–307

[BTVG06] BAY, H. ; TUYTELAARS, T. ; VAN GOOL, L.: SURF: Speeded Up Robust Features. In: *Ninth European Conference on Computer Vision*, 2006

[BV92] BRADLEY, C. ; VICKERS, G. W.: Automated rapid prototyping utilizing laser scanning and free-form machining. In: *Annals of the CIRP* 41 (1992), S. 437–440

[BW91] BEINGLASS, A. ; WOLFSON, H. J.: Articulated Object Recognition, or: How to Generalize the Generalized Hough Transform. In: *Proceeding IEEE Computer Vision and Pattern Recognition Conference*, 1991, S. 461 – 466

[Can86] CANNY, J. F.: A computational approach to edge detection. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6)*, 1986, S. 679–698

[CBSF09] COLLET, A. ; BERENSON, D. ; SRINIVASE, S. S. ; FERGUSON, D.: Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation. In: *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, 2009

[CC04] CHANG, W.-Y. ; CHEN, C.-S.: Pose Estimation for Multiple Camera Systems. In: *Proceeding of the International Conference on Pattern Recognition (ICPR), Volume 2*, 2004, S. 262–265

[CF01] CAMPBELL, R. J. ; FLYNN, P. J.: A Survey Of Free-Form Object Representation and Recognition Techniques. In: *Computer Vision and Image Understanding* 81 (2001), S. 166 – 210

[CG00] CHANG, K.-Y. ; GHOSH, J.: Three-Dimensional Model-based Object Recognition and Pose Estimation using Probabilistic Principal Surfaces. In: *Proc. SPIE Applications of Artificial Neural Networks in Image Processing V* Bd. 3962, 2000, S. 192–203

[CG01] CHANG, K. ; GHOSH, J.: A unified model for probabilistic principal surfaces. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 1, S. 22–41

[Che95] CHENG, Y.: Mean shift, mode seeking, and clustering. In: *IEEE Transactions on Pattern Analysis & Machinge Intelligence* 17(8) (1995), S. 790–799

[CJ03] CARNEIRO, G. ; JEPSON, A.: Multi-scale phase-based local features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Bd. 1, 2003

[CK99] CHANG, P. ; KRUMM, J.: Object Recognition with Color Cooccurrence Histograms. In: *IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, USA*, 1999

[CM92] CHEN, Y. ; MEDIONI, G.: Object modeling by registration of multiple range images. In: *Image and Vision Computing* 10 (1992), Nr. 3, S. 145–155

[CM06] CHUM, O. ; MATAS, J.: Geometric hasing with Local Affine Frames. In: *Proceedings of the 2006 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

[COI10] COIL: *Columbia object image library*. Zugriff am 7. Oktober, 2010. `http://www.cs.columbia.edu/CAVE`

[CR97] CHUA, C. S. ; RAY, J.: Point Signatures: A New Representation for 3D Object Recognition. In: *Internation Journal of Computer Vision* 25(1) (1997), S. 63 – 85

[DD95] DEMENTHON, D. F. ; DAVIS, L. S.: Model-Based Object Pose in 25 Lines of Code. In: *International Journal of Computer Vision (IJCV)* 15 (1995), Nr. 1-2, S. 335–343

[Dev54] DEVOL, G.: *Programmed Article Transfer.* United States Patent 2 988 237, 1954

[DHS00] DUDA, R. O. ; HART, P. E. ; STORK, D. G.: *Pattern Classification*. 2nd Edition. Wiley-Interscience, 2000. – ISBN 0471056693

[Dun89] DUNTEMAN, G.: *Principal Component Analysis.* Sage Puplications, 1989

[DWJ97] DORAI, C. ; WENG, J. ; JAIN, A.: Optimal Registration of Object Views Using Range Data. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 19(10) (1997), S. 1131–1138

[DWJ98] DORAI, C. ; WENG, J. ; JAIN, A.: Registration and Integration of Multiple Object Views for 3D Model Construction. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 30(1) (1998), S. 83–89

[DWJM98] DORAI, C. ; WANG, G. ; JAIN, A.K. ; MERCER, C.: Registration and integration of multiple object views for 3 D model construction. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), Nr. 1, S. 83–89

[EHK03] EKVALL, S. ; HOMANN, F. ; KRAGIC, D.: Object Recognition and Pose Estimation for Robotic Manipulation using Color Cooccurrence Histograms. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, Volume 2*, 2003, S. 1284–1289

[EKH05] EKVALL, S. ; KRAGIC, D. ; HOFFMANN, F.: Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. In: *Image and Vision Computing* 23:11 (2005), S. 943–955

[ESK99] EDIE, A. ; SING, J. ; KANG, B.: Registration and Integration of Textured 3-D Data. In: *Image and Vision Computing* 17 (1999), Nr. 2, S. 135–147

[FA91] FREEMAN, W.T. ; ADELSON, E.H.: The design and use of steerable filters. In: *IEEE Transactions on Pattern analysis and machine intelligence* 13 (1991), Nr. 9, S. 891–906

[FB81] FISCHLER, M.A. ; BOLLES, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications ACM* 24:6 (1981), S. 381–395

[FF95] FUNT, B. V. ; FINLAYSON, G. D.: Color Constant Color Indexing. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 17:5 (1995), S. 522–529

[FHK+04] FROME, A. ; HUBER, D. ; KOLLURI, R. ; BÜLOW, T. ; MALIK, J.: Recognizing Objects in Range Data Using Regional Point Descriptors. In: *Lecture Notes in Computer Science* (2004), S. 224–237

[FHRKV94] FLORACK, LMJ ; HAAR ROMENY, BM ter ; KOENDERINK, JJ ; VIERGEVER, MA: General intensity transformations and differential invariants. In: *Journal of Mathematical Imaging and Vision* 4 (1994), Nr. 2, S. 171–187

[FLK96] FUNG, P.-F. ; LEE, W.-S. ; KING, I.: Randomized Generalized Hough Transform for 2-D grayscale object detection. In: *International Conference on Pattern Recognition* Bd. 13, 1996, S. 511–515

[Gab46] GABOR, D.: Theory of communication. Part 1: The analysis of information. In: *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of* 93 (1946), Nr. 26, S. 429–441

[GDB08] GARCIA, V. ; DEBREUVE, E. ; BARLAUD, M.: Fast k nearest neighbor search using GPU. In: *CVPR Workshop on Computer Vision on GPU*. USA, June 2008

[GH90a] GRIMSON, W. E. L. ; HUTTENLOCHER, D. P.: On the sensitivity of geometric hashing. In: *Proceeding of the 3rd International Conference on Computer Vision, Osake, Japan*, 1990, S. 334–338

[GH90b] GRIMSON, W. E. L. ; HUTTENLOCHER, D. P.: On the sensitivity of the Hough transform for object recognition. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 12(3) (1990), S. 255 – 274

[Gil98] GILLES, S.: *Robust Description and Matching of Images.*, University of Oxford, Diss., 1998

[GL06] GORDON, I. ; LOWE, D. G.: What and where: 3d object recognition with accurate pose. In: *Toward Category-Level Object Recognition* 4170 (2006), S. 67–82

[GRB94] GODING, G. ; RIOUX, M. ; BARIBEAU, R.: Three-dimensional Registration Using Range and Intensity Information. In: *Proceeding SPIE: Viometrics III* Bd. 2350, 1994, S. 279

[HCK97] HUANG, C. Y. ; CAMPS, O. I. ; KANUNGO, T.: Object Recognition Using Appearance-Based Parts and Relations. In: *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, S. 877–883

[HHN88] HORN, B. K. P. ; HILDEN, H. M. ; NEGAHDARIPOUR, S.: Closed-form solution of absolute orientation using orthonormal matrices. In: *Journal of the Optical Society of America* 5 (1988), S. 1127–1136

[HID95] HEBERT, M. ; IKEUCHI, K. ; DELINGETTE, H.: A Spherical Representation for Recognition of Free-Form Surfaces. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 17(7) (1995), S. 681 – 689

[HKM+99] HUANG, J. ; KUMAR, S. R. ; MITRA, M. ; ZHU, W.-J. ; ZABIH, R.: Spatial Color Indexing and Applications. In: *International Journal of Computer Vision* 35:3 (1999), S. 245–268

[HLLS01] HETZEL, G. ; LEIBE, B. ; LEVI, P. ; SCHIELE, B.: 3D Object Recognition from Range Images using Local Feature Histograms. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Bd. 2, 2001

[Hor84] HORN, B. K. P.: Extended Gaussian images. In: *Proceedings of the IEEE* 72(12) (1984), S. 1671 – 1686

[Hor87] HORN, B. K. P.: Closed-form solution of absolute orientation using unit quaternions. In: *Journal of the Optical Society of America* 4 (1987), S. 629–642

[Hou62] HOUGH, P.: *Methods and means for recognising complex patterns.* United States Patent 3 069 654, 1962

[HS88a] HARRIS, C. ; STEPHENS, M.: A combined corner and edge detector. In: *Alvey Vision Conference*, 1988, S. 147–151

[HS88b] HASTIE, T. ; STUETZLE, W.: Principal curves. In: *Journal of the American Statistical Association* (1988), Nr. 84, S. 502–516

[HS90] HARRIS, C. G. ; STENNETT, C.: 3D object tracking at video rate - RAPiD. In: *British Machine Vision Conference (BMVC)*, 1990, S. 73 – 78

[Hu62] HU, M. K.: Visual Pattern Recognition. In: *IEEE Transactions on Information Theory* 8:2 (1962), S. 179–187

[Hur97] HURWITZ, A.: Über die Erzeugung der Invarianten durch Integration. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen* (1897), März, S. 71–90

[HZ04] HARTLEY, R. ; ZISSERMAN, A: *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004. – ISBN 0521540518

[IVT10] IVT: *Integrating Vision Toolkit*. Zugriff am 1. Juni, 2010. `http://www.ivt.sourceforge.net/`

[JH97] JOHNSON, A. ; HEBERT, M.: Surface Registration by Matching Oriented Points. In: *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, S. 121–128

[JH99] JOHNSON, A. ; HEBERT, M.: Using Spin Images for Efficient Object Recognition in Clutterded 3D Scenes. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 21(5) (1999), S. 433 – 449

[Jäh05] JÄHNE, B.: *Digitiale Bildverarbeitung*. 6. Auflage. Springer Verlag, 2005. – ISBN 3540249990

[KB01] KADIR, T. ; BRADY, M.: Saliency, Scale and Image Description. In: *Int. Journal of Computer Vision, 45(2)* (2001), S. 83–105

[KD92] KOENDERINK, J. J. ; DOORN, A. J.: Surface shape and curvature scales. In: *Image and Vision Computing* 10(8) (1992), S. 557 – 565

[KDNS94] KANUNGO, T. ; DOM, B. ; NIBLACK, W. ; STEELE, D.: A Fast Algorithm for MDL-based Multi-band Image Segmentation. In: *Proc. IEEE Computer Vision and Pattern Recognition, Seatlle, Washington*, 1994, S. 609–616

[Kef98] KEFALEA, E.: Object Localization and Recognition for a Grasping Robot. In: *In Proceedings of the 2Jth Annual Conference of the IEEE Industrial Electronics Society (IECON '98)*, 1998, S. 2057–2062

[Kef01] KEFALEA, E.: *Flexible Object Recognition for a Grasping Robot.*, Ruhr-Universität Bochum, Diss., 2001

[KH90] KHOTANZAD, A. ; HONG, Y. H.: Invariant Image Recognition by Zernike Moments. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 12:5 (1990), S. 489–497

[KI93] KANG, S. B. ; IKEUCHI, K.: The Complex EGI: A New Representation for 3-D Pose Determination. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 15(7) (1993), S. 707 – 721

[KK08] KÖSER, K. ; KOCH, R.: Differential Spatial Resection-Pose Estimation Using a Single Local Image Feature. In: *Computer Vision–ECCV 2008* (2008), S. 312–325

[KMN+02] KANUNGO, T. ; MOUNT, D. M. ; NETANYAHU, N. S. ; PIATKO, C. D. ; SILVERMAN, R. ; WU, A. Y.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), S. 881–892

[KRM97] KEFALEA, E. ; REHSE, O. ; MALSBURG, C. v.d.: Object classification based on contours with elastiv graph matching. In: *Proc. of the 3rd Int. Workshop on Visual Form, Italy*, 1997

[KS04] KE, Y. ; SUKTHANKAR, R.: PCA-SIFT: A more distinctive representation for local image descriptors. (2004)

[KVD87] KOENDERINK, J.J. ; VAN DOORN, AJ: Representation of local geometry in the visual system. In: *Biological cybernetics* 55 (1987), Nr. 6, S. 367–375

[KZB04] KADIR, T. ; ZISSERMAN, A. ; BRADY, M.: An affine invariant salient region detector. In: *Proceedings of the 8th European Conference on Computer Vision*, 2004, S. 345–457

[LÁJA06] LEJSEK, H. ; ÁSMUNDSSON, F.H. ; JÓNSSON, B.T. ; AMSALEG, L.: Scalability of local image descriptors: a comparative study. In: *Proceedings of the 14th annual ACM international conference on Multimedia* ACM, 2006, S. 598

[LAT02] LABAYRADE, R. ; AUBERT, D. ; TAREL, J.-P.: Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through 'V-disparity' Representation. In: *IEEE Intelligent Vehicle Symposium, Volume 2*, 2002, S. 646–651

[LF04] LEPETIT, V. ; FUA, P.: Towards Recognizing Feature Points using Classification Trees. / EPFL. 2004 (IC/2004/74). – Forschungsbericht

[LF06] LEPETIT, V. ; FUA, P.: Keypoint Recognition Using Randomized Trees. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006), Nr. 9, S. 1465–1479

[LG94] LINDEBERG, T. ; GÅRDING, J.: Shape-Adapted Smoothing in Estimation of 3-D Depth Cues from Affine Distortions of Local 2-D Brightness Structure. In: *Proc. 3rd European Conf. on Computer Vision*, 1994, S. 389–400

[LH94] LIU, J.-J. ; HUMMEL, R.: Geometric hasing with attributed features. In: *Proceedings of the 1994 Second CAD-Based Vision Workshop*, 1994, S. 9–16

[LHM00] LU, C.-P. ; HAGER, G. D. ; ; MJOLSNESS, E.: Fast and Globally Convergent Pose Estimation from Video Images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 22:6 (2000), S. 610–622

[Lin98] LINDEBERG, T.: Feature detectionwith automatic scale selection. In: *International Journal of Computer Vision (IJVC)* 30 (1998), S. 79–116

[Liu96] LIU, J-.J.: *A Model-Based 3-D Object Recognition System using Geometric Hashing with Attributed Features.*, New York University, Diss., 1996

[Low87] LOWE, D. G.: Three-Dimensional Object Recognition from Single Two-Dimensional Images. In: *Artificial Intelligence* 31(3) (1987), S. 355 – 395

[Low99] LOWE, D. G.: Object recognition form local scale-invariant features. In: *International Conference on Computer Vision, Greece*, 1999, S. 1150–1157

[Low01] LOWE, D. G.: Local Feature View Clustering for 3D Object Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Springer, 2001, S. 682–688

[Low04] LOWE, D. G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision (IJVC)* 60 (2004), S. 91–110

[LPF04] LEPETIT, V. ; PILET, J. ; FUA, P.: Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), USA, Volume 2*, 2004, S. 244–250

[LSP03] LAZEBNIK, S. ; SCHMID, C. ; PONCE, J.: A sparse texture representation using affine-invariant regions. (2003)

[LSS08] LIEBELT, J. ; SCHMID, C. ; SCHERTLER, K.: Viewpoint-Independent Object Class Detection using 3D Feature Maps. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, S. 1–8

[LSW88] LAMDAN, Y. ; SCHWARTZ, J. T. ; WOLFSON, J. H.: Object Recognition by Affine Invariant Matching. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), USA*, 1988, S. 335–344

[LW88] LAMDAN, Y. ; WOLFSON, J. H.: Geometric Hashing: A General and Efficient Model-based Recognition Scheme. In: *IEEE International Conference on Computer Vision (ICCV), USA*, 1988, S. 238–249

[Mar63] MARQUARDT, D. W.: An algorithm for least-squares estimation of nonlinear parameters. In: *1963* 11:2 (1963), S. 431–441

[MBCM99] MARCHAND, E. ; BOUTHEMY, P. ; CHAUMETTE, F. ; MOREAU, V.: Robust Real-Time Visual Tracking using a 2D-3D Model-based Approach. In: *International Conference on Computer Vision (ICCV), Greece*, 1999, S. 262 – 268

[MC88] MA, D. ; CHEN, X.: Hough Transform Using Slope and Curvature as Local Properties to Detect Arbitrary 2D Shapes. In: *Proceeding 9th International Conference on Pattern Recognition*, 1988, S. 511–513

[MCUP04] MATAS, J. ; CHUM, O. ; URBAN, M. ; PAJDLA, T.: Robust wide-baseline stereo from maximally stable extremal regions. In: *Image and Vision Computing* 22 (2004), Nr. 10, S. 761–767

[Mik02] MIKOLAJCZYK, K.: *Detection of local features invariant to affine transformations*. France, Institut National Polytechnique de Grenoble, Diss., 2002

[MMVG98] MINDRU, F. ; MOONS, T. ; VAN GOOL, L.: Color-based moment invariants for viewpoint and illumination independent recognition of planar color patterns. In: *International Conference on Advances in Pattern Recognition, ICAPR* Bd. 98 Citeseer, 1998, S. 113–122

[MN95] MURASE, H. ; NAYAR, S. K.: Visual Learning and Recognition of 3-D Objects from Appearance. In: *International Journal of Computer Vision* 14 (1995), S. 5–24

[MNLF07] MORENO-NOGUER, F. ; LEPETIT, V. ; FUA, P.: Accurate Non-Iterative O(n) Solution to the PnP Problem. In: *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007, S. 1–8

[MO04] MATAS, J. ; OBDRZALEK, S.: Object recognition methods based on transformation covariant features. In: *European Signal Processing Conference (EUSIPO), Austria*, 2004

[MOA04] MASHOR, M. Y. ; OSMAN, M. K. ; ARSHAD, M. R.: 3D Object Recognition Using 2D Moments and HMLP Network. In: *Internation Conference on Computer Graphics, Imaging and Visualization (CGIV), Penang, Malaysia*, 2004, S. 126–130

[MS01] MIKOLAJCZYK, K. ; SCHMID, C.: Indexing based on scale invariant interest points. In: *Proceedings of the 8th International Conference on Computer Vision*, 2001, S. 525–531

[MS02] MIKOLAJCZYK, K. ; SCHMID, C.: An Affine Invariant Interest Point Detector. In: *ECCV*, 2002, S. 128–142

[MS04] MIKOLAJCZYK, K. ; SCHMIDT, C.: Scale & Affine Invariant Interest Point Detectors. In: *International Journal of Computer Vision (IJVC)* 60 (2004), S. 63–86

[MS05] MIKOLAJCZYK, K. ; SCHMID, C.: A performance evaluation of local descriptors. In: *PAMI 27*, 2005, S. 1615–1630

[MTS+05] MIKOLAJCZYK, K. ; TUYTELAARS, T. ; SCHMID, C. ; ZISSERMAN, A. ; MATAS, J. ; SCHAFFALITZKY, F. ; KADIR, T. ; VAN GOOL, L.: A comparison of affine region detectors. In: *IJCV 65*, 2005, S. 43–72

[MVT10] MVTEC: *Halcon*. Zugriff am 2. März, 2010. http://www.mvtec.com/halcon/

[MZ92] MALLAT, S. ; ZHONG, S.: Characterization of signals from multiscale eges. In: *IEEE Transactions on PAMI* (1992), S. 710–732

[NA98] NAIR, D. ; AGGARWAL, J. K.: Moving Obstacle Detection From a Navigating Robot. In: *IEEE Transactions on Robotics and Automation* 14 (1998), Nr. 3, S. 404–416

[Nei64] NEISSER, U.: Visual search. In: *Scientific American* (1964), June, Nr. 210(6), S. 94–102

[NN96] NENE, S. ; NAYAR, S.: Closest Point Search in High Dimensions. In: *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, 1996, S. 859 – 865

[NNM96] NAYAR, S. K. ; NENE, S. A. ; ; MURASE, H.: Real-time 100 Object Recognition System. In: *IEEE International Conference on Robotics and Automation (ICRA), Minneapolis, USA, Volume 3*, 1996, S. 2321–2325

[NS08] NISTÉR, David ; STEWÉNIUS, Henrik: Linear Time Maximally Stable Extremal Regions. In: *ECCV 2008, Part II, LNCS 5303*, Springer-Verlag, 2008, S. 183–196

[ODD96] OBERKAMPF, D. ; DEMENTHON, D. F. ; ; DAVIS, L. S.: Iterative Pose Estimation Using Coplanar Points. In: *Computer Vision and Image Understanding* (1996), Nr. 63, S. 495–511

[OM02] OBDRZALEK, S. ; MATAS, J.: Object Recognition using Local Affine Frames on Distinguished Regions. In: *British Machine Vision Conference (BMVC), UK, Volume 1*, 2002, S. 113–122

[Ope09] OPENCV: *Open Computer Vision Library*. Zugriff am 3. März, 2009. http://www.sourceforge.net/projects/opencvlibrary

[Ope10] OPENMP: *The OpenMP API specification for parallel programming*. Zugriff am 1. Juni, 2010. http://www.openmp.org

[PI97] PROCTER, S. ; ILLINGWORTH, J.: ForeSight: fast object recognition using geometric hashing with edge-triple features. In: *International Conference on Image Processing* 1 (1997), S. 889 – 892

[PLRS04] PONCE, J. ; LAZEBNIK, S. ; ROTHGANGER, F. ; SCHMID, C.: Toward True 3D Object Recognition. In: *In Reconnaisance de Formes et Intelligence Artificelle*, 2004

[Pul99] PULLI, K.: Multiview Registration for Large Data Sets. In: *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling* Bd. 8, 1999

[PW08] PELE, O. ; WERMAN, M.: A linear time histogram metric for improved sift matching. In: *Computer Vision–ECCV 2008* (2008), S. 495–508

[Ray10] RAYTRIX (Hrsg.): *CluViz*. Zugriff am 4. Juni: Raytrix, 2010. `http://www.raytrix.de/index.php/CluViz.html`

[RCB04] ROY, S. D. ; CHAUDHURY, S ; BANERJEE, S.: Active Recognition through Next View Planning: A Survey. In: *Pattern Recognition* (2004)

[RCSM01] RUIZ-CORREA, S. ; SHAPIRO, L. G. ; MEILA, M.: A New Signature-Based Method for Efficient 3-D Object Recognition. In: *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)* Bd. 1, 2001

[Rei91] REIS, T. H.: The Revised Fundamental Theorem of Moment Invariants. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 13:8 (1991), S. 830–834

[RL01] RUSINKIEWICZ, S. ; LEVOY, M.: Efficient variants of the ICP algorithm. In: *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, 2001, S. 145–152

[RLSP06] ROTHGANGER, F. ; LAZEBNIK, S. ; SCHMID, C. ; PONCE, J.: 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. 66(3) (2006), S. 231–259

[Roc03] ROCKETT, P. I.: Performance Assessment of Feature Detection Algorithms: A Methodology and Case Study on Corner Detectors. In: *IEEE Transactions on Image Processing* 12:12 (2003), S. 1668–1676

[RPM96] REHSE, Ö. ; PÖTZSCH, M. ; MALSBURG, C. v.d.: Edge Information: A Confidence Based Algorithm Emphazising Steady Curves. In: *Proc. of Int. Conf. on Artificial Neural Networks*, 1996, S. 851–856

[SB91] SWAIN, M. J. ; BALLARD, D. H.: Color Indexing. In: *International Journal of Computer Vision* 7:1 (1991), S. 11–32

[Sch08] SCHLEGEL, H.: *Schnelle Tiefenrekonstruktion interessanter Punkte aus Stereofolgen.*, Technische Universität Dresden, Diplomarbeit, 2008

[SD96] STRICKER, M. A. ; DIMAI, A.: Color indexing with weak spatial constraints. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* Bd. 2670, 1996, S. 29–40

[SE03] SCHNEIDER, P. J. ; EBERLY, D. H.: *Geometric Tools for Computer Graphics*. Elsevier Science USA, 2003. – ISBN 1–55860–594–0

[Sed88] SEDGEWICK, R.: *Algorithmns*. 2nd Edition. Addison-Wesley, 1988. – ISBN 0201066734

[SGS09] SANDE, K. van d. ; GEVERS, T. ; SNOEK, C.: Evaluating color descriptors for object and scene recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2009)

[Sim96] SIMON: *Fast and Accurate Shape-Based Registration.*, Carnegie Mellon University, Diss., 1996

[SK94] SZELISKI, R. ; KANG, S. B.: Recovering 3D shape and motion from image streams using nonlinear least squares. In: *Journal of Visual Communication and Image Representation* (1994), Nr. 5:1, S. 10–28

[SK00] SCHNEIDERMAN, H. ; KANADE, T.: A Statistical Method for 3D Object Detection Applied to Faces and Cars. In: *IEEE Computer Society Conference on Computer Vision And Pattern Recognition (CVPR)* Bd. 1, 2000, S. 1746–1751

[SM92] STEIN, F. ; MEDIONI, G.: Structural Indexing: Efficient 3-D Object Recognition. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* 14(2) (1992), S. 125 – 145

[SM97] SCHMID, C. ; MOHR, R.: Local Grayvalue Invariants for Image Retrieval. In: *IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 15, No. 5*, 1997, S. 530–535

[SMFF07] SALVI, J. ; MATABOSCH, C. ; FOFI, D. ; FOREST, J.: A review of recent range image registration methods with accuracy evaluation. In: *Image and Vision Computing* 25 (2007), Nr. 5, S. 578–596

[SS94] SER, P.-K. ; SIU, W.-C.: Non-analytic Object Recognition Using the Hough Transform with Matching Technique. In: *Computers and Digital Techniques* 141(1) (1994), S. 231–235

[SS02] SCHARSTEIN, D. ; SZELISKI, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: *International journal of computer vision* 47 (2002), Nr. 1, S. 7–42

[ST94] SHI, J. ; TOMASI, C.: Good Features to Track. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, 1994, S. 153–158

[SVLP98] SINGH, R. ; VOYLES, R. M. ; LITTAU, D. ; PAPANIKOLOPOULOS, N. P.: Grasping Real Objects Using Virtual Images. In: *Proceedings of the 37th IEEE Conference on Decision & Control, Tampe, Florida, USA*, 1998, S. 3269–3274

[SZ02] SCHAFFALITZKY, F. ; ZISSERMAN, A.: Multi-view matching for unordered image sets. In: *Computer Vision-ECCV 2002* (2002), S. 414–431

[TK03] TAYLOR, G. ; KLEEMAN, L.: Fusion of Multimodal Visual Cues for Model-Based Object Tracking. In: *Australasian Conference on Robotics and Automation (ACRA), Australia*, 2003

[TL94] TURK, G. ; LEVOY, M.: Zippered Poygon Meshes from Range Images. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994, S. 318

[TLCT09] TANG, F. ; LIM, S.H. ; CHANG, N.L. ; TAO, H.: A novel feature descriptor invariant to complex brightness changes. (2009)

[TM08] TUYTELAARS, T. ; MIKOLAJCZYK, K.: A Survey on Local Invariant Features. In: *Foundations and Trends in Computer Graphics and Vision (FnTCGV)* 1:1 (2008), S. 1–94

[Tra09] TRAN, K. Q.: *Effiziente Ähnlichkeitssuche in hochdimensionalen Bilddatenbanken.*, Hochschule Darmstadt, Diplomarbeit, 2009

[TVG04] TUYTELAARS, T. ; VAN GOOL, L.: Matching Widely Separated Views based on Affine Invariant Regions. In: *International Journal of Computer Vision (IJVC)* 59 (2004), S. 61–85

[TW09] TOEWS, M. ; WELLS, W.: SIFT-Rank: Ordinal description for invariant feature correspondence. (2009)

[UN00] ULRICH, I. ; NOURBAKHSH, I.: Appearance-Based Obstacle Detection with Monocular Color Vision. In: *Proc. of the AAAI National Conference on Artificial Intelligence, Austin*, 2000

[USBE03] ULRICH, M. ; STEGER, C. ; BAUMGARTNER, A ; EBNER, H.: Real-Time Object Recognition Using a Modified Generalized Hought Transform. In: *Pattern Recognition* 36(11) (2003), S. 2557 – 2570

[VF10] VEDALDI, A. ; FULKERSON, B.: *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. `http://www.vlfeat.org/`. Version: 2010

[VJ01] VIOLA, P. ; JONES, M.: Rapid object detection using a boosted cascade of simple features. In: *Conference on Computer Vision and Pattern Recoginition*, 2001, S. 511–518

[VK95] VETTERLI, M. ; KOVAČEVIĆ, J.: *Wavelets and subband coding*. Citeseer, 1995

[VL07] VALGREN, C. ; LILIENTHAL, A.: Sift, Surf and seasons: Long-term outdoor localization using local features. In: *Proc. of 3rd European Conference on Mobile Robots*, 2007

[WB07] WINDER, S.A.J. ; BROWN, M.: Learning local image descriptors. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, 2007, S. 1–8

[Wei97] WEIK, S.: Registration of 3-D Partial Surface Models Using Luminance and Depth Information. In: *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, S. 93–100

[WR97] WOLFSON, H. J. ; RIGOUTSOS, I.: Geometric Hashing: An Overview. In: *IEEE Computational Science and Engineering* 4:4 (1997), S. 10–21

[WSB98] WEBER, R. ; SCHEK, H.-J. ; BLOTT, S.: A Quantitative Analysis and Performance Study for Similarity Search Methods in High-Dimensional Spaces. In: *Proceedings of the 22th International Conference on Very Large Data Bases (VLDB)*. USA, 1998, S. 194–205

[Wu09] WU, C.: *SiftGPU: A GPU Implementation of Scale Invariant Feature Transform*. Zugriff am 7. Mai, 2009. `http://cs.unc.edu/~ccwu/siftgpu`

[XO93] XU, L. ; OJA, E.: Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. In: *CVGIP: Image Understanding* 57(2) (1993), S. 331 – 338

[XOK90] XU, L. ; OJA, E. ; KULTANEN, P.: A new curve detection method: Randomized Hough Transform (RHT). In: *Pattern Recognition Letters* 11 (1990), S. 331 – 338

[YFEB99] YAMANY, S. M. ; FRAAG, A. A. ; EL-BIALY, A.: Free-form surface registration and object recognition using surface signatures. In: *IEEE International Conference on Computer Vision, Grece*, 1999

[Zer34] ZERNIKE, F.: *Physica*. 1934

[Zha98] ZHANG, Z.: Determining the Epipolar Geometry and its Uncertainty: A Review. In: *International Journal of Computer Vision* (1998), Nr. 27(2), S. 161–198

[Zha00] ZHANG, Z.: A Flexible New Technique for Camera Calibration. In: *IEEE Transactions on Pattern Analysis and Macine Intelligence 22 (11)*, 2000, S. 1330–1334

[Zoe10] ZOELLNER, C.: *Beschleunigung der Musterklassifikation in einem Bildverarbeitungssystem durch den Einsatz von Grafikprozessoren.*, Fachhochschule Schmalkalden, Diplomarbeit, 2010