Automatische Kalibrierung von Kameranetzwerken basierend auf lokaler Bewegung

Diplomarbeit

Institut für Theoretische Informatik
Prof. A. Waibel

von

cand. inf. Cem Taylan Aslan

31. Mai 2007

Betreuer:

Prof. A. Waibel
Dr.-Ing. Rainer Stiefelhagen
Dipl.-Inf. Keni Bernardin

Hilfsmittel

Die dieser Arbeit zugrunde liegenden Programme wurden in C/C++ unter Linux auf den Rechnern des Instituts für Theoretische Informatik programmiert und ausgeführt. Alle Abbildungen wurden mit Persistence of Vision Ray Tracer (POV-Ray) Version 3.6 erzeugt. Die Arbeit selbst wurde mit LaTeXgeschrieben.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angeführten Hilfsmittel angefertigt habe.

Karlsruhe, den 31. Mai 2007

Cem T. Aslan

Cem T. Aslan

Zusammenfassung

In dieser Arbeit wird ein automatisches Kalibrierungssystem entwickelt, das durch die Beobachtung der Bewegung in einem Raum die extrinsische Kalibrierung eines Kameranetzwerks berechnet. Das System untersucht dabei aufgezeichnete Videoaufnahmen auf verwertbare Merkmale für die Kalibrierung. Hierbei wird hauptsächlich auf den höchsten Punkt eines sich bewegenden Objektes geachtet. Sofern sich mehrere Objekte in einer Szene bewegen, wird im Gegensatz zu traditionellen Ansätzen nicht nach Korrespondenzen zwischen den Merkmalen gesucht. Stattdessen betrachtet der entwickelte Kalibrierungsalgorithmus zur Bestimmung der paarweise externen Parameter alle Merkmalskombinationen. Mit Hilfe einer Houghtransformation ist der Algorithmus jedoch trotzdem in der Lage, mit genügend Daten die richtige Kalibrierung zu berechnen. Anhand der paarweisen Kalibrierungen werden anschließend alle möglichen Kamerakonfigurationen mittels einer Bewertungsfunktion getestet und die beste Konfiguration ausgewählt.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mir während meiner Diplomarbeit geholfen haben und zur Seite standen. Insbesondere bei Keni Bernardin, der mir helfend mit Rat und Tat zur Seite stand, geduldig diese Arbeit las und korrigierte. Weiterhin möchte ich mich auch bei Björn Keuter fürs Korrekturlesen bedanken. Auch möchte ich Tobias Huck und Sebastian Janczik danken, da sie für Testaufnahmen zur Verfügung standen. Der größte Dank gilt meinen Eltern für ihr Vertrauen und den Rückhalt, den ich immer wieder bei ihnen finden konnte.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Motivation

Vor nicht allzu langer Zeit waren intelligente Räume noch Zukunftsvisionen, die nur aus Science-Fiction Romanen oder Fernsehserien bekannt waren. Die Szenarien reichten dabei von Begrüßungen, der Steuerung der Stereoanlage bis hin zu komplexen Arbeitsumgebungen. Alle Szenarien hatten aber eins gemein: Der Raum versuchte, die Personen in ihm so gut wie möglich bei Ihren Tätigkeiten zu unterstützen [12]. Der intelligente Raum ist dabei in der Lage, mit der Person zu kommunizieren und zu interagieren. Die Kommunikation wird dabei auf multimodaler Ebene geführt, beispielsweise durch Zeigegesten zusammen mit Sprachkommandos. Mit der gestiegenen Prozessorleistung heutiger Rechner und dem Preisverfall der Komponenten (Prozessoren, Kameras, Mikrofone) rückten diese Szenarien jedoch immer weiter in den Mittelpunkt der Forschung. Das Bestreben der Forscher besteht dabei darin, einen Raum mit kognitiven Fähigkeiten auszustatten, so dass der Raum wahrnehmen kann, was in ihm geschieht. Aufgrund dieser Wahrnehmung kann der Raum dann hilfreiche Aktionen durchführen. Inzwischen forschen bereits weltweit viele Gruppen an dem Themengebiet: Intelligente Räume.

Auch das Institut für Theoretische Informatik an der Universität Karlsruhe forscht an einem intelligenten Raum. Der "Smartroom" besteht dabei aus mehreren festen und beweglichen Kameras und einigen Mikrofonarrays, um das Geschehen im Raum *beobachten* zu können. Die Forschung konzentriert sich dabei auf die Algorithmen, die die gewünschten perzeptiven Fähigkeiten erbringen [21] [4] [15]. Entsprechend diesen Beobachtungen kann der Raum erforderliche Aktionen durchführen und somit die gewünschten Dienste erbringen.

Zu den typischen perzeptiven Fähigkeiten eines intelligenten Raumes gehören unter anderem:

Personentracking
Auffinden einer oder mehrerer Personen in einer oder mehreren Bildfolgen und Berechnung der dreidimensionalen Position dieser Person im Raum

Zeigegestenerkennung
Erkennung der Richtung, in die eine Person zeigt

Spracherkennung
Erkennung von Sprache auf Distanz

Aufmerksamkeitsbestimmung
Erkennen, worauf eine Person ihre Aufmerksamkeit richtet

Identifikation
Erkennung einer Person anhand von visuellen und akustischen Merkmalen

Um bessere Ergebnisse zu liefern, greifen einige dieser Algorithmen auf zusätzliche Informationen über die Beschaffenheit des Raumes zurück. Beispielsweise die Raumgrenzen, bekannte Gegenstände im Raum oder die Position, Ausrichtung und Eigenschaften der Sensoren.

Für die Berechnung der dreidimensionalen Position einer Person können beispielsweise die Kalibrierungen der Kameras (siehe Abschnitt 3.1) und die zweidimensionalen Positionen der Person auf den einzelnen Kamerabildern verwendet werden. Mit diesen Daten kann dann durch Triangulation die dreidimensionale Position bestimmt werden. Weiterhin können aber auch Personentracker durch Hinzunahme der Kalibrierungsinformationen ihre Ergebnisse verbessern. Beispielsweise kann eine Hypothese anhand des Triangulationsfehlers weiter betrachtet oder verworfen werden. Mit Hilfe der Kalibrierung lässt sich auch abschätzen, wo ein Punkt aus dem Bild einer Kamera im Bild einer anderen Kamera zu suchen ist, wodurch sich der Suchaufwand stark verringert. Auch lassen sich so Entfernungen bzw. die Größe von Objekten abschätzen.

Insbesondere die externen Parameter, d.h. die Position und Ausrichtung einer Kamera (siehe Abschnitt 3.1.1), sind in gewissem Maße störanfällig. Abhängig von der Befestigung der Kameras kann sich deren Position oder Ausrichtung im Laufe der Zeit ändern. In diesem Falle muss die betroffene Kamera neu kalibriert werden. Generell empfiehlt es sich in regelmäßigen Abständen die Kalibrierung aller Kameras zu erneuern, jedoch mindestens zu überprüfen. Die gängigsten Verfahren zur Kalibrierung von Kameras benutzen meist ein Kalibrierungsobjekt, im Allgemeinen ein Brett mit Schachbrettmuster, dessen Aussehen und Ausmaße bekannt sein müssen. Generell hat sich das Schachbrett als Kalibrierungsobjekt durchgesetzt, da zum einen das Muster einfach zu erkennen ist und zum anderen die Gitternetzlinien sich gut zum Abschätzen der Verzerrungen durch die Linse eignen. Für die Kalibrierung muss das Objekt aufgestellt und seine Position im Raum vermessen werden. Sofern sich mehrere Kameras im Raum aufhalten, muss dieser Schritt für jede Kamera wiederholt werden. Falls die Kameras einen überlappenden Sichtbereich haben, kann das Objekt auch so positioniert werden, dass es im Sichtbereich von allen Kameras liegt. In diesem Fall muss nur einmal die Position manuell gemessen werden. Die Problematik besteht dabei darin, dass ein kleines Kalibrierungsobjekt auf den Kameras auch relativ klein abgebildet wird. Abbildung 1.1 verdeutlicht dieses Problem anhand von zwei Beispielen. Entsprechend schwierig gestaltet sich die Markierung der entsprechenden Merkmale. Abweichungen von einem Pixel können dabei die Kalibrierung verschlechtern. Aus diesem Grund werden hauptsächlich größere Objekte verwendet. Entsprechend aufwendig gestaltet sich dann aber auch der gesamte Kalibrierungsprozess.

In dieser Diplomarbeit wird ein Kalibrierungsverfahren entwickelt, das diesen Prozess vereinfacht und ohne spezifisches Kalibrierungsobjekt auskommt. Hierbei konzentriert sich die Arbeit ausschließlich auf die extrinsische Kalibrierung. Die internen Parameter, beispielsweise Verzerrungen und die Brennweite der Linse, werden vorausgesetzt. Dies ist jedoch nicht von Nachteil, da bedingt durch den Kameraaufbau diese Werte sich mit der Zeit nahezu nicht ändern. Es reicht daher aus, vor der Montage der Kameras, einmal die intrinsische Kalibrierung durchzuführen.

Für die Kalibrierung sollen Aufnahmen von vier fest installierten Kameras des Smartrooms benutzt werden, in denen sich eine oder mehr Personen bewegen. Im ersten Schritt

(a) (b)

Abbildung 1.1: Ein großer Kalibrierungsteppich (a) mit einer Größe von 3m×3m und ein kleineres Kalibrierungsschachbrett (b) mit einer Größe von 84cm×84cm.

werden dabei die interessanten Merkmale aus den aufgezeichneten Videodaten extrahiert. Hierbei ist es theoretisch egal, welche Merkmale verwendet werden. Da sich im Smartroom jedoch meist Menschen bewegen, wird dieser Schritt mit Hilfe dieses Kontextwissens angepasst. Im darauf folgenden Schritt werden diese Merkmale dann für die automatische Kalibrierung der Kameras verwendet. In traditionellen Kalibrierungsansätzen ist es hierbei wichtig, dass die jeweiligen Merkmale zwischen den einzelnen Kamerabildern korrespondieren. D.h. ein Merkmal im Bild von Kamera 1 und das korrespondierende Merkmal im Bild von Kamera 2 markieren das gleiche Objekt. Die Bestimmung einer Relation zwischen diesen Merkmalen ist jedoch nicht trivial. Aus diesem Grund soll dieses Kalibrierungssystem keine korrekten Korrespondenzen voraussetzen. Es werden stattdessen alle möglichen Merkmalskombinationen dem System übergeben. Dies vereinfacht die Merkmalsextraktion deutlich, da dieser Schritt je nach Kameraausrichtung sehr aufwendig werden kann. Die Kalibrierung erfolgt danach paarweise in einem Hybrid-Verfahren. Mit einem Suchalgorithmus werden die besten Parameter für die relative Position und Ausrichtung der Kameras in einem dreidimensionalen Suchraum ermittelt. Die Hypothesen (Parameter) werden mit einer Houghtransformation bewertet. Aus den errechneten Kalibrierungen wird abschließend das Kameranetzwerk aufgebaut.

1.2 Überblick

Im Folgenden wird zuerst auf die gängigen Kalibrierungsverfahren eingegangen, der aktuelle Stand der Forschung gezeigt und auf Vor- bzw. Nachteile eingegangen. In Kapitel 3 werden einige Grundlagen vermittelt, die für das Verständnis dieser Arbeit notwendig sind. Darauf folgend wird das entwickelte Verfahren zur automatischen Kalibrierung im Detail vorgestellt. Anschließend werden Experimente, die zur Bewertung des Verfahrens während dieser Arbeit durchgeführt wurden, beschrieben und deren Ergebnisse aufgelistet. In Kapitel 6 endet die Arbeit mit einer Zusammenfassung und Ausblick.

2 Stand der Forschung

Im Folgenden werden einige Arbeiten bezüglich der Kalibrierung von Kameras vorgestellt und näher betrachtet. Das grundlegende Ziel bei der Kalibrierung ist die Abbildung eines Punktes im dreidimensionalen Raum auf einen Punkt im Bild der Kamera durch ein mathematisches Modell (siehe Abschnitt 3.1) so anzupassen, dass es mit der realen Abbildung durch die Kamera übereinstimmt. Es werden dabei hauptsächlich Verfahren betrachtet, die die externen Parameter ermitteln, d.h. die Parameter, die die Orientierung und Position der Kamera im Raum beschreiben. Die Verfahren lassen sich dabei grob in zwei Kategorien aufteilen: die traditionelle Kalibrierung und die Selbstkalibrierung. Die traditionelle Kalibrierung benutzt dabei ein vordefiniertes Kalibrierungsobjekt. Beim Selbstkalibrierungsansatz muss dieses Kalibrierungsobjekt nicht weiter spezifiziert werden, d.h. es müssen keine dreidimensionalen Koordinaten bekannt sein. Nichtsdestotrotz wird aber ein Objekt benötigt.

2.1 Traditionelle Kalibrierung

Bei der traditionellen Kalibrierung wird von einer Kamera ein in seinem Aussehen und seinen Ausmaßen bekanntes Objekt betrachtet. Meist handelt es sich dabei um ein Schachbrett. Bereits Roger Y. Tsai [20] hat ein einfaches Schachbrettmuster benutzt, um handelsübliche Kameras zu kalibrieren. Das Verfahren versucht dabei das zugrunde liegende Kameramodell (siehe Abschnitt 3.1) so anzupassen, dass die Abbildung der realen dreidimensionalen Koordinaten des Kalibrierungsobjektes in Bildkoordinaten mit dem tatsächlich aufgezeichneten Bild übereinstimmt. Als Merkmale werden dabei die inneren Ecken eines Schachbretts benutzt. Prinzipiell können dabei auch andere Merkmale benutzt werden. Wichtig ist nur, dass die dreidimensionalen Koordinaten der Merkmale bekannt sind und diese auf den aufgezeichneten Bildern leicht erkannt und markiert werden können. Zusätzlich zu den externen Parametern werden bei diesem Verfahren auch die internen Parameter (z.B. Verzerrungen der Linse) ermittelt. Die Problematik dieses Verfahrens besteht jedoch darin, dass die exakten dreidimensionalen Positionen des Objektes, relativ zum Nullpunkt des Raumes, vorhanden sein müssen. Die manuelle Vermessung der Objektposition ist im Allgemeinen sehr aufwendig. Sofern weiterhin Fehler in der Messung vorliegen, beeinträchtigt dies natürlich auch die Kalibrierung. Trotzdem war dieses Verfahren ein Meilenstein in der Forschung und wird in seinen Grundlagen auch heute noch verwendet.

Eine sehr weit verbreitetes Programm für die Kalibrierung ist die *Camera Calibration Toolbox for Matlab* [5]. Hierbei wird ähnlich wie bei [20] ein Schachbrett als Kalibrierungsobjekt benutzt. Die Kalibrierung teilt sich dabei in die intrinsische und die extrinsische Kalibrierung auf. Die intrinsische Kalibrierung ermittelt die internen Parameter einer

Kamera, d.h. die Brennweite, den Bildmittelpunkt und Verzerrungen, die durch die Linse entstehen. Da die intrinsische Kalibrierung unabhängig von der Position ist, kann diese für jede Kamera einzeln durchgeführt werden. Dies gilt im Übrigen für alle Kalibrierungsverfahren, da die internen Parameter nicht von den externen abhängen. Darauf folgend kann dann für alle Kameras die extrinsische Kalibrierung durchgeführt werden. Hierfür muss die Position des Schachbretts im Raum bekannt sein. Die Schwierigkeit liegt auch hier in der Handhabung des evtl. sehr großen Kalibrierungsobjektes und der Vermessung der korrekten Position.

2.2 Selbstkalibrierung

Im Gegensatz zur traditionellen Kalibrierung setzt der Selbstkalibrierungsansatz [8] nicht voraus, dass Position und Ausmaße des Kalibrierungsobjektes bekannt sein müssen. Diese Verfahren finden selbstständig Korrespondenzpunkte aus unterschiedlichen Aufnahmen der gleichen Szene. Hierfür sind jedoch zwei Kameras, die einen überlappenden Sichtbereich haben, notwendig. Mit Hilfe der Homographie, die die einzelnen Korrespondenzpunkte aufeinander abbildet, kann die Kalibrierung bestimmt werden. Insbesondere die Fundamental- und die Essential-Matrix (siehe Abschnitt 3.2) werden hierfür oft verwendet. Ohne Wissen über die Ausmaße und Position des betrachteten Objektes ist die extrinsische Kalibrierung jedoch nur bis auf einen Skalierungsfaktor genau möglich. Weiterhin liegen die externen Parameter nur relativ zu einer der Kameras vor. D.h. die korrekte dreidimensionale Position im Raum ist nicht bekannt, nur die Richtung, aus der Sicht der anderen Kamera.

In [2] werden beispielsweise der Kopf und die beiden Hände einer Person als Merkmale für die Kalibrierung benutzt. Das System sucht dabei in beiden Kamerabildern nach dem Kopf, der linken und der rechten Hand. Anhand der Korrespondenzen kann die relative Position und Ausrichtung der beiden Kameras bis auf einen Skalierungsfaktor ausgerechnet werden. In einem zusätzlichen Schritt wird der Skalierungsfaktor bestimmt. Hierbei bewegt die Person eine Hand entlang einer quadratischen Fläche mit bekanntem Ausmaß. Der Skalierungsfaktor ergibt sich dann durch die gemessene dreidimensionale Bewegung und der tatsächlichen Bewegung im Weltkoordinatensystem. Die Schwierigkeit hierbei liegt darin, die detektierten Körperteile einer Kamera auf die Merkmale im anderen Kamerabild richtig abzubilden (d.h. die linke Hand in Bild 1 wird auf die linke Hand in Bild 2 abgebildet). Insbesondere wenn der Winkel zwischen beiden Kameras groß ist oder wenn sich mehrere Personen ganz oder teilweise im Aufnahmebereich bewegen, wird das Problem erschwert.

[6] hingegen betrachtet das Problem von Kamera-Netzwerken, die sich über eine größere Fläche erstrecken. Mit dem traditionellen Ansatz könnte hierfür einerseits ein sehr großes Kalibrierungsobjekt benutzt werden, das von allen Kameras betrachtet werden kann. Andernfalls müsste das Kalibrierungsobjekt für jede Kamera gesondert positioniert werden, so dass es gut sichtbar ist. Jedoch müsste jedesmal die Position des Objektes neu vermessen werden. Beide Alternativen des traditionellen Ansatzes haben einen hohen Arbeitsaufwand. Stattdessen wird in [6] jedoch ein virtuelles Kalibrierungsobjekt erzeugt, sprich in einem abgedunkelten Raum wird mit einer LED oder Taschenlampe ein virtuelles Kalibrierungsobjekt *in die Luft gezeichnet*. Anhand der Punktkorrespondenzen wird mit dem Selbstkalibrierungsansatz die relative extrinsische Kalibrierung zwischen jeweils

zwei Kamerapaaren ermittelt. Das Punktkorrespondenzproblem ist hierbei trivial, da zu jedem Zeitpunkt immer nur ein Marker im Bild sichtbar ist. Abschließend werden die einzelnen Kalibrierungen zur endgültigen Kalibrierung zusammengesetzt. Das Verfahren zeigt dabei eine einfache alternative der Kalibrierung ohne ein sperriges Kalibrierungsobjekt, jedoch ist es nur in abdunkelbaren Räumen, bedingt durch die verwendeten Merkmale (LED oder Taschenlampe), nutzbar.

Ein vollautomatischer Ansatz wird in [7] aufgeführt. Hierbei werden PTZ-Kameras[1] verwendet, deren Aussehen und Ausmaße bekannt sind. Durch oszillierende Bewegungen einer Kamera kann eine andere Kamera diese selbstständig finden. Als Kalibrierungsmerkmal wird die Linse der Kamera benutzt, da deren Aussehen und Ausmaße bekannt sind. In [7] wird beschrieben, wie anhand dieses Merkmals der Abstand zwischen den Kameras und die Orientierung der gegenüberstehenden Kamera ermittelt werden kann. Eine intrinsische Kalibrierung wird vorausgesetzt und kann wie in [11] beschrieben automatisch ermittelt werden. Vorteilhaft ist hier die komplett automatische Kalibrierung des Kameranetzwerkes ohne weiteres Zutun. Für ein anderes Kamera-Modell müsste das Verfahren aber entsprechend angepasst werden. Dies gilt insbesondere für ein heterogenes Kameranetzwerk. Für starre Kameras ist dieses Verfahren nicht anwendbar. Sofern sich im Suchbereich auch Menschen aufhalten können, beispielsweise wenn die Kameras nicht an der Decke sondern auf Kopfhöhe montiert sind, müsste der Raum geräumt werden, um das Kalibrierungsverfahren nicht zu stören.

Sofern kein eindeutig zu erkennendes Merkmal für die Kalibrierung benutzt wird, liegt die größte Problematik in der korrekten Abbildung der Merkmale zwischen zwei Kamerabildern. [19] versucht diese Problematik mithilfe von Co-Motion Statistiken [17] [18] zu lösen. Der Vorteil dieses Verfahrens liegt dabei darin, dass nur auf Bewegungen geachtet wird. Die Farbwerte der Bilder werden nicht betrachtet. Beispielsweise kann ein Objekt von zwei Seiten unterschiedlich aussehen, eine Abbildung ist daher nicht trivial. Die Abbildung mit Hilfe von Bewegungs-Merkmalen ist jedoch meist einfacher, vor allem bei sehr markanten Bewegungen lässt sich diese Problematik leicht lösen. Die Bestimmung der extrinsischen Kalibrierung erfolgt danach mit Fundamental-Matrizen (s. Abschnitt 3.2). Die Korrektheit der Kalibrierung hängt hierbei hauptsächlich von den gefundenen Korrespondenzpunkten ab. Je nach Situation (bspw. mit viel Bewegung) kann das Finden von korrekten Korrespondenzen jedoch erheblich erschwert werden.

Einen weiteren Ansatz, bei dem ein gewisses Maß an Informationen über die betrachtete Szene liegen muss, beschreibt [16]. Hierbei wird die Position und Ausrichtung von Stadionkameras während der Aufzeichnung eines Fußballspiels berechnet. Als Merkmale dienen dabei die Spielfeldmarkierungen in der Umgebung der beiden Tore. Unter Zunahme eines Modells der Spielfeldmarkierungen können die gefundenen Merkmale entsprechend abgebildet und die extrinsische Kalibrierung bestimmt werden. Die Nachteile dieses Verfahrens liegen wie bei [7] darin, dass ein bestimmtes Szenario vorausgesetzt wird und daher das Verfahren relativ starr ist.

[14] betrachtet für die Kalibrierung die Umrisse der betrachteten Person. Unter der Voraussetzung, dass alle Kameras aufrecht ausgerichtet sind, sind auf allen Aufnahmen zwei Punkte zu sehen, die ungefähr korrespondieren. Das ist hierbei der höchste und der tiefste Punkt des betrachteten Objektes. Anhand dieser Informationen wird mit Hilfe der Epipolargeometrie (s. Abschnitt 3.2) die extrinsische Kalibrierung durchgeführt. Der Vorteil

[1] PTZ-Kamera: Pan-Tilt-Zoom-Kamera, dieser Kameratyp kann sich motorgestützt um zwei Achsen (Pan, Tilt) drehen und zusätzlich den Zoom ändern.

dieses Ansatzes liegt darin, dass der Kopf einer Person relativ selten verdeckt ist und dadurch Verfahren durch eventuelle Überdeckungen nicht gestört wird. Die Güte der Kalibrierung hängt dabei wie bei [19] von den gefundenen Korrespondenzpunkten ab. Die Problematik erschwert sich beispielsweise, wenn sich mehrere Personen auf den Aufnahmen befinden.

Abschließend lässt sich sagen, dass der Selbstkalibrierungsansatz die Kalibrierung wesentlich vereinfacht, da keine (unter Umständen auch sperrige) Kalibrierungsobjekte benötigt werden. Weiterhin existieren sogar einige vollautomatische Ansätze [7][16], um die Kalibrierung von Kameranetzwerken ohne weitere Hilfe durchzuführen. Jedoch setzen diese noch einige Randbedingungen voraus. Trotzdem sind einige Ansätze sehr inspirierend. Diese werden in dieser Arbeit aufgegriffen und in teilweise abgewandelter Form verwendet.

3 Grundlagen

Dieses Kapitel wendet sich den Grundlagen zu, die in dieser Arbeit benutzt werden. Dabei wird das nötige Grundwissen vermittelt, auf dem im weiteren Verlauf aufgebaut wird.

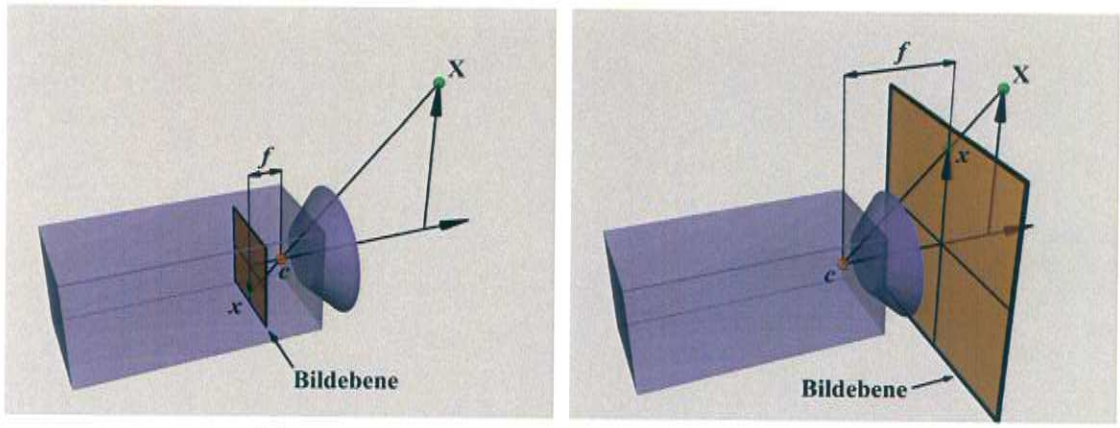
3.1 Kameramodelle und Kalibrierung

Kameramodelle beschreiben mathematisch die Abbildung eines Punktes im dreidimensionalen Raum auf die Bildkoordinaten eines aufgezeichneten Kamerabildes und umgekehrt. Mit Hilfe dieser Modelle ist es beispielsweise möglich, die Bildkoordinaten eines betrachteten bekannten Objektes vorherzusagen oder anhand von mehreren aufgezeichneten Bildern die dreidimensionale Position des betrachteten Objektes zu berechnen. Die bekanntesten Modelle sind dabei das Lochkameramodell, das Mattscheibenmodell (s. Abb. 3.1) und das Modell nach Tsai [20].

Das Lochkameramodell ist ein mathematisches Modell, das von der Lochkamera abgeleitet wurde. Alle Lichtstrahlen passieren dabei die Kamera durch ein unendlich kleines Loch c und projizieren so die Umgebung auf die Bildebene innerhalb der Kamera (s. Abb. 3.1 (a)). Es handelt sich hierbei um ein idealisiertes Modell, da keine Verzerrungen auftreten. Im Vergleich zu einer realen Kamera entspricht der Punkt c einer idealen Linse ohne Verzerrungen und die Bildebene entspricht dem Bildsensor einer Digitalkamera oder dem lichtempfindlichen Film einer analogen Kamera. Die Brennweite entspricht dem Abstand zwischen dem Einfallsloch c und der Bildebene. Da jedoch bei dieser Projektion das Bild seitenverkehrt ist und auf dem Kopf steht, wurde ein leicht abgewandeltes Modell eingeführt. Im Mattscheibenmodell befindet sich die Bildebene vor der Kamera und bildet die einfallenden Lichtstrahlen ab, bevor sie den Punkt c erreichen (s. Abb. 3.1 (b)). Das Modell nach Tsai basiert auf dem Mattscheibenmodell, modelliert aber zusätzlich Verzerrungen, die durch die Linse entstehen.

Im Folgenden wird hierbei auf eine Abwandlung des Kameramodells nach Tsai eingegangen, das in der Praxis häufig zum Einsatz kommt. Der Unterschied hierbei besteht in einer genaueren Modellierung der radialen und tangentialen Verzerrungen der Linse [23].

Jedes Kameramodell ist abhängig von Parametern, die das allgemeine Modell auf einen entsprechenden Kameratypus und die zugrunde liegende Konfiguration anpassen. Diese Parameter werden durch die Kalibrierung ermittelt und teilen sich in zwei Bereiche auf: die extrinsische und die intrinsische Kalibrierung, auch externe und interne Parameter genannt.



(a) (b)

Abbildung 3.1: Die Abbildung eines Punktes $\mathbf{X}$ auf den Punkt x auf der Bildebene nach dem Lochkameramodell (a) und dem Mattscheibenmodell (b).

3.1.1 Externe Parameter

Die externen Parameter sind für die Transformation des betrachteten Punktes aus dem Weltkoordinatensystem in das Kamerakoordinatensystem zuständig. Diese Transformation ist nur von der Position und Ausrichtung der Kamera abhängig und besteht aus:

$\mathcal{T}$: Dem Translationsvektor der Kamera

$\mathcal{R}$: Der Rotationsmatrix der Kamera

Ein Punkt $\vec{x}_{\text{world}}$ des Weltkoordinatensystems wird mit Hilfe der Gleichung (3.1) in den Punkt $\vec{x}_{\text{cam}}$ im Kamerakoordinatensystem transformiert.

$$\vec{x}_{\text{cam}} = \mathcal{R}^T \cdot \left(\vec{x}_{\text{world}} - \mathcal{T}\right) \tag{3.1}$$

3.1.2 Interne Parameter

Die internen Parameter sind für die Abbildung eines Punktes vor der Kamera auf die Bildebene verantwortlich. Neben der Abbildung werden auch zusätzlich Verzerrungen der Linse berücksichtigt. Im Folgenden werden kurz die Parameter aufgelistet.

f: die Brennweite der Linse

c: der Bildmittelpunkt

s: der Scherungsfaktor

κ: die radialen und tangentialen Verzerrungskoeffizienten der Linse[1]

[1] κ besteht hierbei aus insgesamt fünf Koeffizienten. Das ursprüngliche Modell nach Tsai benutzte nur einen.

Die Abbildung eines Punktes $\vec{x}_{\text{cam}}$ im Kamerakoordinatensystem auf den Punkt $\vec{x}_{\text{bild}}$ in Bildkoordinaten geschieht mit den Gleichungen (3.2) bis (3.7):

$$\vec{x}_n = \begin{pmatrix} \frac{\vec{x}_{\text{cam}_x}}{\vec{x}_{\text{cam}_z}} & \frac{\vec{x}_{\text{cam}_y}}{\vec{x}_{\text{cam}_z}} \end{pmatrix}^T = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \tag{3.2}$$

$$r^2 = \mathbf{x}^2 + \mathbf{y}^2 \tag{3.3}$$

$$\vec{x}_d = \begin{pmatrix} \vec{x}_{d_x} \\ \vec{x}_{d_y} \end{pmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6)\ \vec{x}_n + \vec{xd}_{\text{tangential}} \tag{3.4}$$

$$\vec{xd}_{\text{tangential}} = \begin{pmatrix} 2\kappa_3 \mathbf{xy} + \kappa_4 (r^2 + 2\mathbf{x}^2) \\ \kappa_3 \left(r^2 + 2\mathbf{y}^2\right) + 2\kappa_4 \mathbf{xy} \end{pmatrix} \tag{3.5}$$

$$K = \begin{pmatrix} f_x & s f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

$$\vec{x}_p = \begin{pmatrix} \vec{x}_{\text{bild}_x} \\ \vec{x}_{\text{bild}_y} \\ 1 \end{pmatrix} = K \begin{pmatrix} \vec{x}_{d_x} \\ \vec{x}_{d_y} \\ 1 \end{pmatrix} \tag{3.7}$$

Mit Gleichung (3.2) wird $\vec{x}_{\text{cam}}$ auf normalisierte Koordinaten entsprechend dem Lochkameramodell projiziert. Gleichung (3.3) bis (3.5) modellieren die radialen und tangentialen Verzerrungen der Linse. Mit Hilfe der Kamera-Matrix K in Gleichung (3.6) und (3.7), die die restlichen intrinsischen Parameter enthält, wird schließlich $\vec{x}_d$ auf die Bildkoordinaten abgebildet.

3.2 Epipolargeometrie

Mit Hilfe der Epipolargeometrie kann zwischen zwei Sichten ein Punkt aus der einen Sicht auf eine Linie in das Bild der anderen Sicht abgebildet werden, wobei die Linie den entsprechenden korrespondierenden Punkt enthält. Im dreidimensionalen Raum hilft die folgende Konstruktion zur Veranschaulichung (s. Abb 3.2). Die Punkte c und c' repräsentieren die Kameramittelpunkte zweier Kameras. $\mathbf{X}$ bezeichnet den betrachteten Punkt. Die Epipolarebene π wird durch diese drei Punkte aufgespannt. Der Schnitt zwischen der Epipolarebene und der Bildebene definiert die Epipolarlinie. Die interessante Eigenschaft der Epipolarlinie ist, dass die abgebildeten Punkte x und x' sich auf dieser Linie befinden. Anders betrachtet wird mit Hilfe der Epipolargeometrie die Sichtgerade von Kamera c auf Punkt $\mathbf{X}$ in den Bildbereich von Kamera c' abgebildet (s. Abb. 3.3). Im Folgenden wird auf die Fundamentalmatrix und die Essentialmatrix eingegangen, mit deren Hilfe sich diese Abbildung realisieren lässt. Der Vorteil liegt darin, dass dafür nicht unbedingt die Position der Kameras bekannt sein muss. Weiterführende Informationen sind in [10] zu finden.

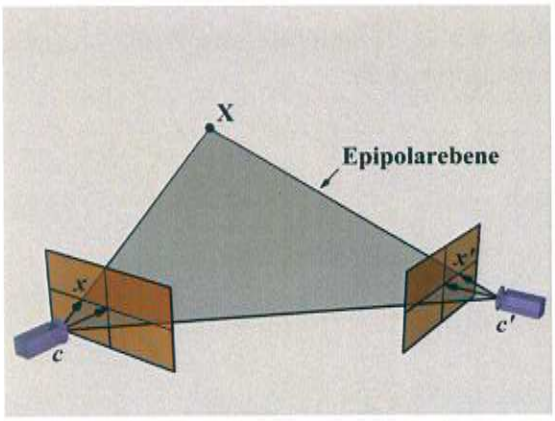


Abbildung 3.2: Die Epipolarebene π im Zusammenhang mit den Kameramittelpunkten c, c' und dem betrachteten Punkt X.

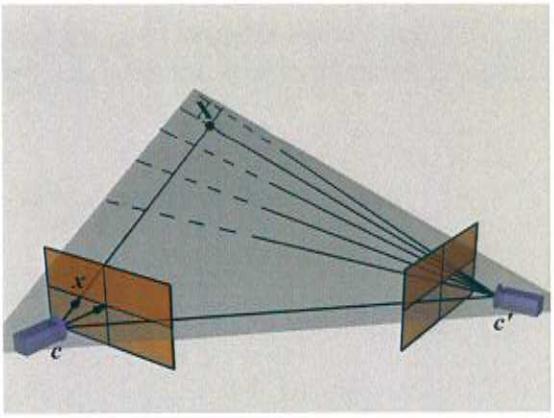


Abbildung 3.3: Die Abbildung der Sichtgeraden von c nach X auf die Bildebene von Kamera c' (Epipolarlinie).

3.2.1 Die Fundamentalmatrix

Die Fundamentalmatrix F ist eine 3×3-Matrix, die die oben beschriebene Abbildung für beliebige Punkte ermöglicht und Bedingung (3.8) erfüllt.

$$x'^T F x = 0 \tag{3.8}$$

Mit Hilfe von korrekten Punktreferenzen zwischen beiden Bildebenen kann die Matrix F bestimmt werden [3] [9].

3.2.2 Die Essentialmatrix

Bei der Essentialmatrix E handelt es sich um eine spezielle Form der Fundamentalmatrix, für den Fall von normalisierten Bildkoordinaten. So kann mit Hilfe der Kamera-Matrizen die Essentialmatrix aus der Fundamentalmatrix berechnet werden.

$$E = K'^T F K \text{ mit } K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.9}$$

Weiterhin gilt auch für die Essentialmatrix

$$\hat{x}'^T E \hat{x} = 0 \text{ mit } \hat{x} = K^{-1} x \tag{3.10}$$

Der wichtigste Aspekt der Essentialmatrix (bzgl. der Kalibrierung) ist, dass die Rotation $\mathcal{R}$ und Translation $\mathcal{T}$ aus der Essentialmatrix E mit einer Singularwertzerlegung berechnet werden können. Dabei gilt:

$$E = \begin{pmatrix} 0 & -\mathcal{T}_z & \mathcal{T}_y \\ \mathcal{T}_z & 0 & -\mathcal{T}_x \\ -\mathcal{T}_y & \mathcal{T}_x & 0 \end{pmatrix} \cdot \mathcal{R} \tag{3.11}$$

Weiterführende Informationen bzgl. der Zerlegung der Essential-Matrix in Translation und Rotation sind in [10] [9] und [22] zu finden.

3.3 Hintergrundsubtraktion

Bei der Background Subtraction (Hintergrundsubtraktion) handelt es sich um ein Verfahren mit dem eine Aufnahme in Hinter- und Vordergrund aufgeteilt und somit die interessanten Bereiche (i.d.R. befinden sich diese im Vordergrund) extrahiert werden. Hierfür wird der Hintergrund vorgegeben oder automatisch eingelernt ($P_{\text{background}}$). Der Vordergrund lässt sich dann mit Hilfe einer einfachen Subtraktion aus aktuellem Bild (P_{current}) und Hintergrund ermitteln. Jedes Pixel im Bild $P_{\text{current}}(x, y)$ wird als Vordergrund markiert, wenn gilt: $P_{\text{diff}}(x, y) > threshold$ (s. Abb. 3.4).

$$P_{\text{diff}}(x, y) = |P_{\text{background}}(x, y) - P_{\text{current}}(x, y)| \tag{3.12}$$

$$P_{\text{current}}(x, y) \in Vordergrund \equiv P_{\text{diff}}(x, y) > threshold \tag{3.13}$$

In dieser Arbeit werden die aufgezeichneten Bilder zuerst in Graubilder umgewandelt, daher reicht es hier nur die Grauwerte zu betrachten. Das Verfahren kann jedoch auch im RGB-Farbraum genutzt werden, indem die Differenzbildung für jeden Farbkanal getrennt durchgeführt wird. Beispielsweise kann dann ein Pixel als Vordergrund markiert werden, sobald die Differenz auf einem Farbkanal größer als der vorgegebene Grenzwert ist.
Weiterhin ist es vor allem bei längeren Aufnahmen nötig den Hintergrund zu adaptieren. Zum Beispiel, wenn Gegenstände aus dem Hintergrund bewegt werden und somit in den Vordergrund gelangen. Mit Hilfe der Hintergrundadaption werden Objekte, die sich über einen vorgegebenen Zeitraum nicht bewegen, wieder in den Hintergrund aufgenommen. Hierfür wird in jedem Bild der als Hintergrund klassifizierte Bereich verwendet $P_{\text{currentBack}}$ (Gleichung 3.14).

$$P_{\text{background}_{i+1}}(x, y) = (1 - \alpha) \cdot P_{\text{background}_i}(x, y) + \alpha \cdot P_{\text{currentBack}}(x, y) \tag{3.14}$$

Die Lernrate wird hierbei durch α bestimmt.

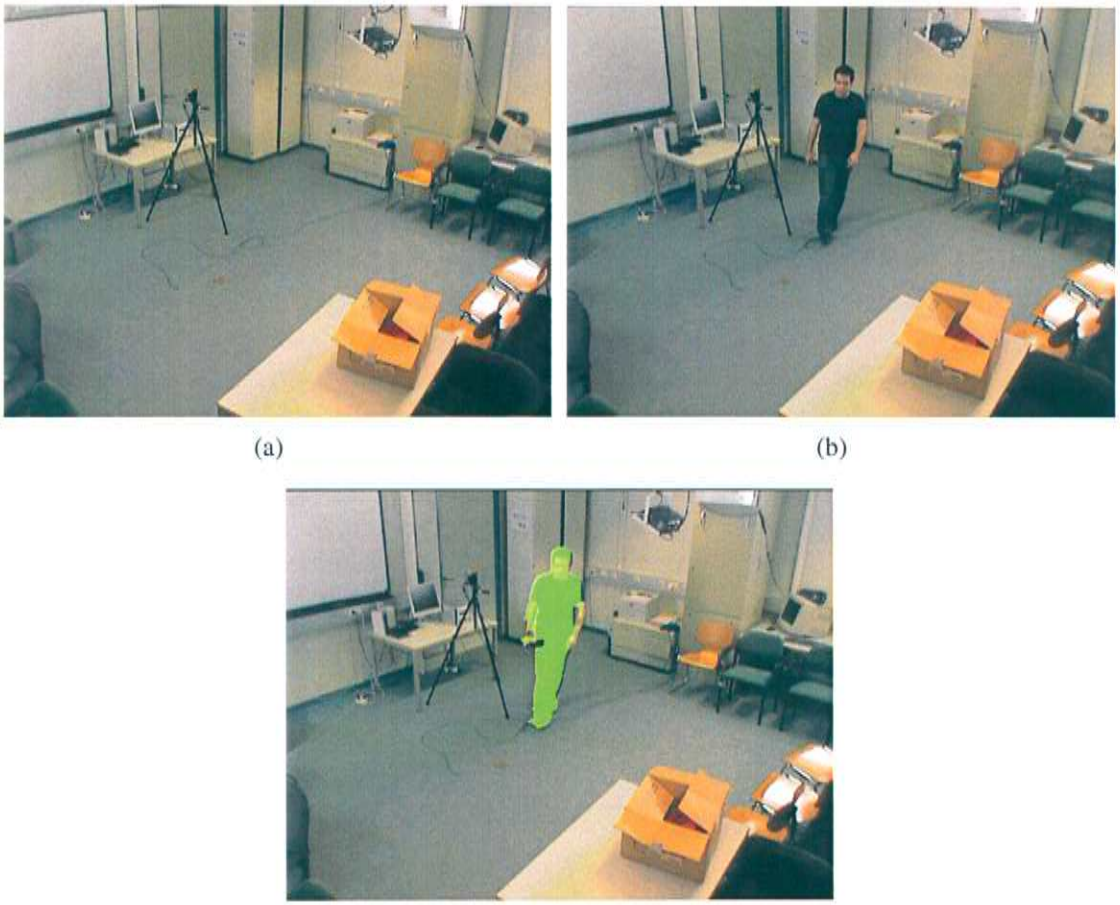
3.4 Morphologische Operatoren

Morphologische Operatoren werden hauptsächlich zur Glättung in der Bildvorverarbeitung genutzt. Die Operatoren werden dabei auf Binär- bzw. Graustufenbilder angewandt. In dieser Arbeit werden der Dilatations- und der Erosionsoperator benutzt und daher näher erläutert.

3.4.1 Dilatation

Mit Hilfe der Dilatation werden Bildobjekte vergrößert. Dadurch lassen sich mehrere, nah beieinander liegende Objekte vereinen bzw. Risse schließen. Hierfür wird jedes Pixel des Bildes und dessen 3×3 Umgebung untersucht. Jedem untersuchten Pixel wird dabei als Wert das Maximum seiner 3×3 Umgebung zugewiesen.

$$P_{\text{dilate}}(x, y) = \max_{\substack{x-1 \leq x_c \leq x+1 \\ y-1 \leq y_c \leq y+1}} P(x_c, y_c) \tag{3.15}$$



(a) (b)

(c)

Abbildung 3.4: Hintergrundsubtraktion: (a) der eingelernte Hintergrund, (b) die aktuelle Aufnahme mit einer Person, (c) der als Vordergrund erkannte Bereich (grün eingefärbt).

3.4.2 Erosion

Im Gegensatz zur Dilatation verkleinert die Erosion Bildobjekte. Dadurch können bspw. kleine Objekte aus dem Bild gelöscht werden. Außerdem werden einzelne falsch markierte Pixel (bspw. durch Rauschen, etc.) gefiltert. Auch wenn die Erosion gegensätzlich zur Dilatation arbeitet, handelt es sich hierbei nicht um eine inverse Funktion. Ähnlich wie bei der Dilatation wird jedes Pixel und seine 3×3 Umgebung betrachtet. Allerdings wird einem Pixel nun das Minimum seiner 3×3 Umgebung zugewiesen.

$$P_{\text{erode}}(x, y) = \min_{\substack{x-1 \leq x_c \leq x+1 \\ y-1 \leq y_c \leq y+1}} P(x_c, y_c) \tag{3.16}$$

3.5 Houghtransformation

Bei der Houghtransformation handelt es sich um ein robustes Verfahren, um geometrische Formen in Bildern zu erkennen [13]. Die ursprüngliche Anwendung lag in der Detektion von Geraden. Das Verfahren kann aber zur Detektion von Ellipsen und anderen parametrisierbaren geometrischen Formen erweitert werden. Bei der Detektion von Geraden wird hierfür das Bild zuerst in ein Kantenbild umgewandelt. Für jeden Punkt auf einer Kante wird nun eine Hypothese erstellt, wie die zugehörige Gerade im Bild liegen könnte. Zusätzlich wird mitgezählt, wie oft eine Hypothese vorgeschlagen wurde. Die Hypothesen, die am meisten Vorgeschlagen wurden, sind mit hoher Wahrscheinlichkeit auch korrekt.

Für die Erkennung von Geraden wird beispielsweise folgende Parameterform verwendet:

$$d = x \cdot \cos(\varphi) + y \cdot \sin(\varphi) \tag{3.17}$$

x und y entsprechen dabei den Bildkoordinaten, d dem senkrechten Abstand der Geraden zum Ursprung und φ dem Winkel zu einer Koordinaten-Achse. Mit dieser Gleichung können nun Kantenpunkte (x,y) und deren Gradienten im Bild auf eine Gerade (d,φ) im Parameterraum (auch Houghraum genannt) abgebildet werden. Jede Kombination (d,φ) entspricht dabei einer möglichen Geraden (Hypothese) im Bildraum und werden in eine Akkumulatormatrix eingetragen. Die Houghtransformation an Stelle (d,φ) entspricht dabei dem Wert der Akkumulatormatrix an dieser Stelle, und repräsentiert wie oft diese Hypothese vorgeschlagen wurde. Die Punkte (d,φ) im Houghraum mit hohen Werten entsprechen mit hoher Wahrscheinlichkeit Geraden im Bild. Abbildung 3.5 zeigt ein Beispiel.

Der Algorithmus ist relativ einfach erweiterbar. Bei einem Kreis handelt es sich beispielsweise um einen dreidimensionalen Parameterraum (C_x, C_y, r). C_x und C_y stehen für den Kreismittelpunkt und r für den Radius.

3.6 Triangulation

Mit Hilfe der Triangulation ist es ausgehend von zwei oder mehreren Kamerasichten möglich, die Koordinaten des betrachteten Objektes zu berechnen. Dabei werden von den Kamerapositionen die entsprechenden Sichtgeraden zum betrachteten Objekt berechnet. Der

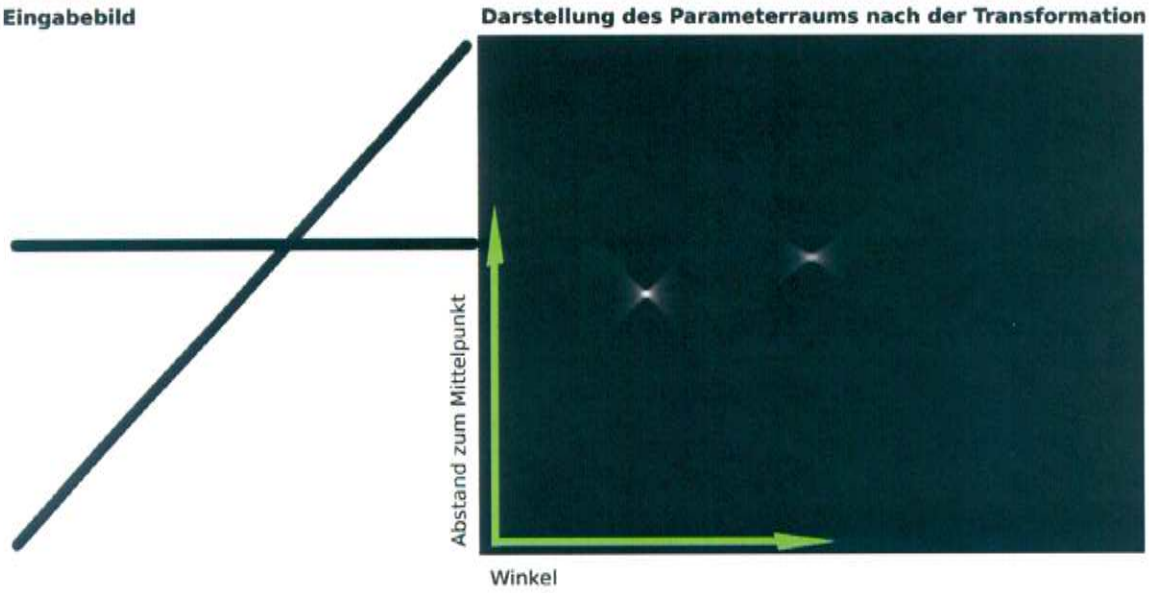


Abbildung 3.5: Links: ein Beispielbild mit zwei Geraden. Rechts: die Summe für die Wertepaare (d,φ). Die hellen Punkte entsprechen den Houghtransformierten der beiden Geraden. (aus: http://de.wikipedia.org/wiki/Bild:Hough-example-result.png)

Schnittpunkt entspricht der gesuchten Position. Abbildung (3.6) zeigt diesen Sachverhalt. Das Objekt wird in diesem Fall von drei Kameras betrachtet, deren Position bekannt ist. Mit Hilfe der internen und externen Parameter (s. Abschnitt 3.1.1 und 3.1.2) lässt sich die Sichtgerade (blau) im dreidimensionalen Raum für jede Kamera berechnen. Hierfür muss zuerst der Bildpunkt entzerrt und auf die virtuelle Bildebene projiziert werden. Das Vorgehen entspricht dabei dem inversen der in Abschnitt 3.1.2 beschriebenen Projektion eines Punktes im dreidimensionalen Raum in Bildkoordinaten. Das Ergebnis ist der Richtungsvektor $\vec{lov}$ der Sichtgeraden im Kamerakoordinatensystem. Mit den externen Parametern lässt sich dieser nun ins Weltkoordinatensystem transformieren. Auch hier ist das Vorgehen invers zu dem in Abschnitt 3.1.1 beschriebenen Transformation. Die Sichtgerade lässt sich mit dem Richtungsvektor $\vec{lov}$ und der Position des Kamerazentrums c durch die Parameterform in Gleichung (3.18) beschreiben.

$$L_i = \vec{c}_i + s \cdot \vec{lov}_i \tag{3.18}$$

Bedingt durch Messfehler schneiden sich die Sichtgeraden nicht exakt. Die Lösung $\vec{x}$ entspricht in diesem Fall dem Punkt, der den kleinsten Abstand (s. Gleichung 3.19) zu allen Sichtgeraden hat. Der einfachste Weg besteht hierbei über eine Minimierung der Fehlerquadrate (s. Gleichung 3.20). Als Fehlermaß kann zusätzlich der Abstand des berechneten Punktes zu den Sichtgeraden berechnet werden.

$$d\left(L_i, \vec{x}\right) = \frac{\left|\left(\vec{x} - \vec{c}_i\right) \times \vec{lov}_i\right|}{\left|\vec{lov}_i\right|} \tag{3.19}$$

$$\min \sum_{\forall i} d\left(L_i, \vec{x}\right)^2 \tag{3.20}$$

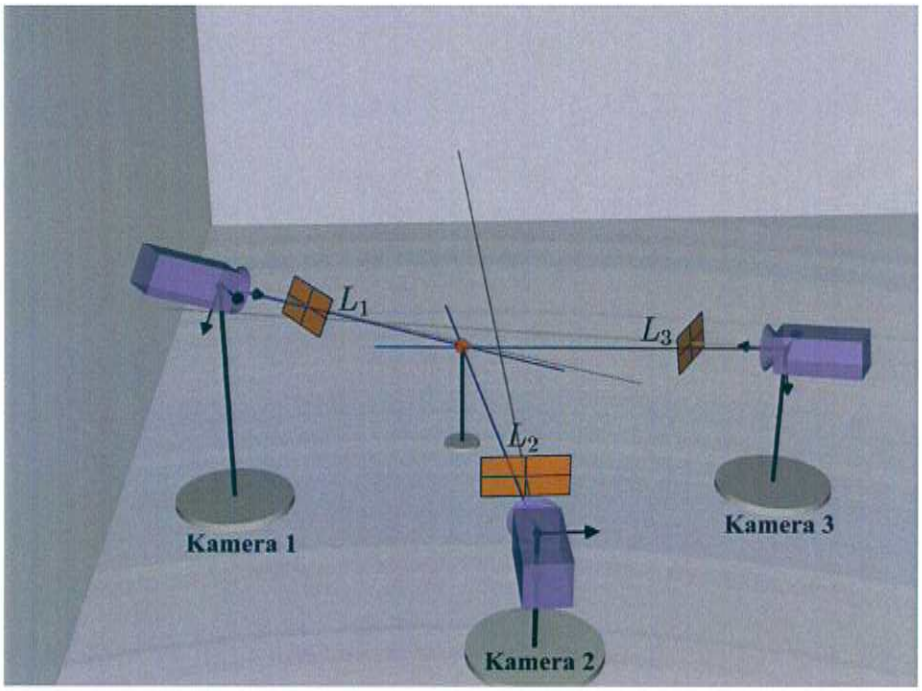


Abbildung 3.6: Die Position des betrachteten Objektes wird aus den Sichtgeraden L_1, L_2 und L_3 (blau) von drei Kameras trianguliert.

4 Automatische Kalibrierung eines Kameranetzwerks mittels lokaler Bewegungsmerkmale

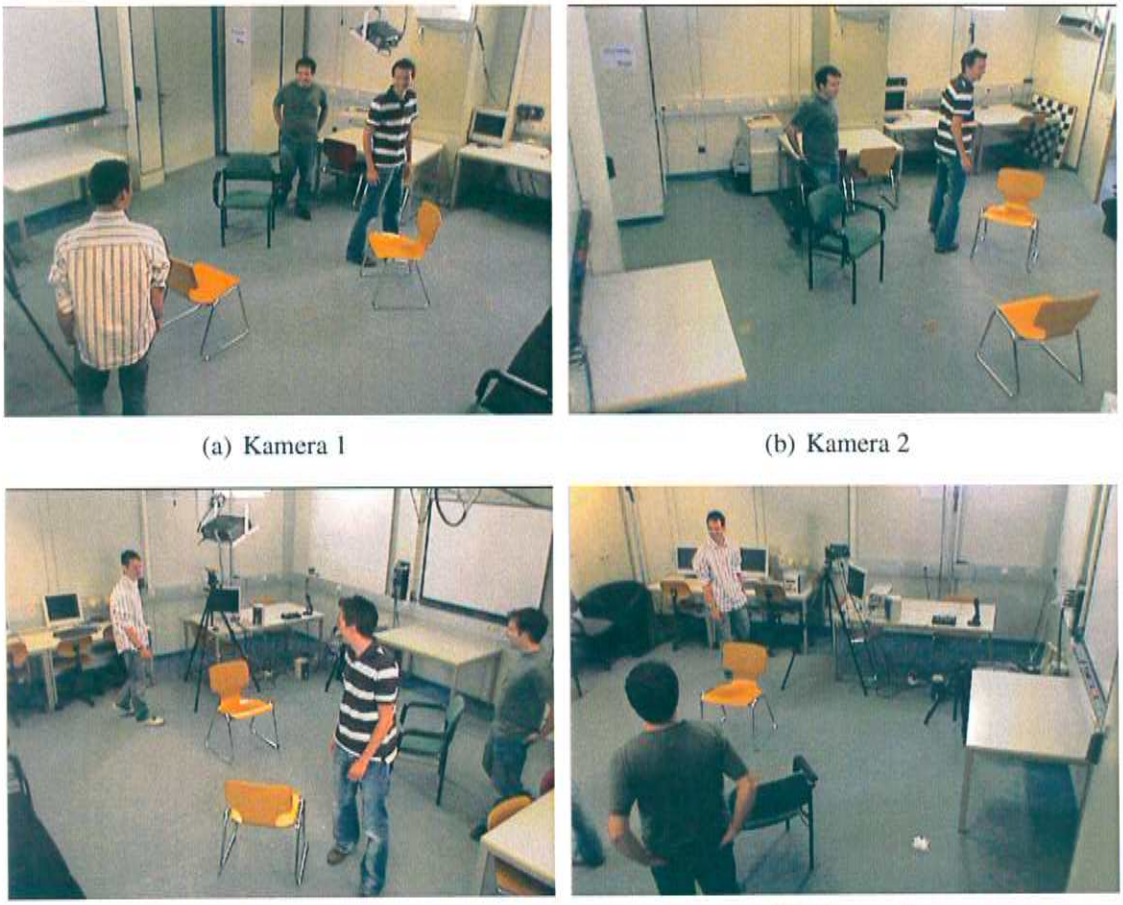
Im folgenden Kapitel wird nun näher auf den Aufbau des automatischen Kalibrierungssystems eingegangen. Hierfür werden die einzelnen Schritte im Detail erklärt.

Der Kalibrierungsprozess lässt sich in drei grobe Schritte unterteilen. Zuerst werden die aufgezeichneten Videodaten ausgewertet und verwertbare Merkmale extrahiert. Hierbei wird nur die Bewegung berücksichtigt; Farbinformationen werden ignoriert. Durch das Einbeziehen von Kontextwissen über die Umgebung wird zusätzlich die Erkennung optimiert. Im Gegensatz zu den bisher bekannten Ansätzen wird jedoch nicht nach entsprechenden Korrespondenzen zwischen den extrahierten Merkmalen gesucht. Anhand der Merkmale der einzelnen Kamerabilder wird mit Hilfe des in dieser Arbeit entwickelten Verfahrens die relative extrinsische Kalibrierung paarweise bestimmt. Das Verfahren kommt dabei ohne korrekte Punktkorrespondenzen aus. Stattdessen werden alle Kombinationen zwischen den extrahierten Merkmalen in Betracht gezogen. Hierbei werden implizit falsche und korrekte Kombinationen erkannt.

Da die berechneten Kalibrierungen nur relativ zueinander vorliegen, wird abschließend das gesamte Kameranetzwerk aufgespannt. Hierfür werden die einzelnen Positionen der Kameras trianguliert. Da mehr Informationen als benötigt zum Aufspannen des Kameranetzwerks vorhanden sind, werden alle möglichen Kameranetzwerke erzeugt, und aus dieser Menge mit Hilfe einer Bewertungsfunktion das beste ausgesucht.

4.1 Auswertung der Videodaten

Bevor eine Kalibrierung durchgeführt werden kann, müssen zuerst Aufnahmen der entsprechenden Kameras vorliegen. Hierzu werden für jeweils ein Kamerapaar die entsprechenden Aufnahmen Bild für Bild betrachtet und nach verwendbaren Merkmalen durchsucht. Es ist dabei wichtig, dass die Aufnahmen synchron vorliegen, d.h. jedes Bildpaar wurde zur gleichen Zeit aufgenommen. Abbildung 4.1 zeigt synchrone Beispielbilder von allen vier Kameras. Grundsätzlich wird dabei zuerst jedes Bild für sich auf Merkmale untersucht. Danach werden die Bilder immer paarweise miteinander verglichen. Sofern jedes der zwei betrachteten Bilder mindestens ein Merkmal aufweist, werden alle möglichen Kombinationen zwischen den Merkmalen als mögliche Korrespondenzen abgespeichert, andernfalls wird dieses Bildpaar ignoriert und mit dem nächsten fortgefahren. Vor



(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 4.1: Beispielaufnahmen von allen vier Kameras im Smartroom mit drei Personen.

der Weitergabe werden die Koordinaten jedoch noch entzerrt, da durch die Kameralinse radiale- und tangentiale Verzerrungen auftreten. Die so entzerrten Merkmale werden abschließend an die Kalibrierung weitergereicht.

4.1.1 Merkmalsextraktion

Die Kalibrierung erfolgt während dieser Arbeit für ein Kameranetzwerk, das in einem geschlossenem Raum installiert ist. Aus diesem Grund lassen sich schon im Voraus einige Annahmen über die aufgezeichneten Bilder machen, die sich für eine Verbesserung der Merkmalsextraktion ausnutzen lassen. Speziell handelt es sich dabei um folgende Annahmen:

- Nur sich bewegende Objekte sind interessant
- Es bewegen sich größtenteils Menschen im Raum
- Die Kameras sind alle ungefähr aufrecht ausgerichtet

Im Folgenden wird kurz auf die einzelnen Annahmen näher eingegangen.

Nur sich bewegende Objekte sind interessant

Das Finden von Punktkorrespondenzen in nicht bewegten Szenen anhand von Farbinformationen ist im Allgemeinen ein recht komplexes Problem in der digitalen Bildverarbeitung, besonders wenn der Winkel zwischen den Kameras sehr groß ist. Beispielsweise können Objekte aus unterschiedlichen Winkeln betrachtet völlig unterschiedlich aussehen, wie z.B. Personen mit einem hellen Pullover und einer dunklen Jacke. Von vorne ist der helle Pullover gut zu sehen, während von hinten die dunkle Jacke den Pullover verdeckt. Um dieser Problematik aus dem Weg zu gehen, werden nur sich bewegende Objekte beachtet, da anhand von Bewegungsmerkmalen mögliche Korrespondenzen einfacher gefunden werden können. Wenn sich ein Objekt in einer Kamera bewegt, bewegt es sich auch in allen anderen Kameras, in denen es sichtbar ist. Auf den Sonderfall der Verdeckung wird im nächsten Punkt weiter eingegangen.

Es bewegen sich größtenteils Menschen im Raum

Da ein intelligenter Raum hauptsächlich für Menschen entworfen wird, ist die Wahrscheinlichkeit sehr groß, dass eine erkannte Bewegung sich auf einen Menschen zurückführen lässt. Dies ist jedoch nicht immer der Fall. Häufig werden Schatten auf dem Boden oder den Wänden fälschlicherweise als sich bewegende Menschen erkannt. Genau genommen wird nicht direkt nach Personen gesucht, sondern nur die Tatsache ausgenutzt, dass der Kopf einer Person der höchste Punkt derselben ist. In einem intelligenten Raum ist der Kopf so gut wie nie aus den Blickwinkeln der Kameras verdeckt. Da sich im Allgemeinen Lichtquellen an der Decke befinden, werden so auch Schatten nicht detektiert. Wichtig ist hierbei, dass der höchste Punkt aus allen Kamerasichten der gleiche ist.

Die Kameras sind alle ungefähr aufrecht ausgerichtet

Im Allgemeinen werden Kameras immer aufrecht ausgerichtet montiert, da ein Mensch so die aufgezeichneten Bilder besser erkennen und auswerten kann. Mit Hilfe dieses Wissen kann der höchste Punkt einer Person sehr einfach gefunden werden. Der höchste Punkt aus einer Menge von Punkten ist der mit der kleinsten Y-Komponente[1].

Die Merkmalsextraktion lässt sich in zwei Schritte unterteilen. Zuerst werden mögliche Hypothesen anhand einer einfachen Hintergrundsubtraktion erzeugt und in einem zweiten Schritt falsche Hypothesen verworfen.

Die Hintergrundsubtraktion (s. Abschnitt 3.3) bietet eine einfache Möglichkeit interessante Bereiche eines Bildes zu ermitteln. Dabei wird das Bild in Vordergrund und Hintergrund aufgeteilt. Für die Merkmalsextraktion ist nur der Vordergrund interessant. Dieser wird mit der Hintergrundsubtraktion mit einem Schwellenwert von 30 berechnet. Damit verstellte Objekte nach einer gewissen Zeit wieder in den Hintergrund aufgenommen werden, wird der Hintergrund mit einer Lernrate von 0,0022 adaptiert. Die Hypothesen werden anhand der als Vordergrund markierten zusammenhängenden Bereiche gebildet. Jedoch können einzelne Bereiche zerrissen sein, obwohl sie zusammengehören. Auch können durch Rauschen einzelne Pixel falsch markiert werden. Um dieses Problematik zu umgehen, wird der Vordergrundbereich mit Morphologischen Operatoren geglättet. In dieser

[1] Bei Bildern ist der Null-Punkt oben links, die X-Achse verläuft nach rechts und die Y-Achse nach unten.

Arbeit wird die folgende Kombination und Reihenfolge von Morphologischen Operatoren auf den Vordergrund angewandt:

1. zwei Dilatationen
2. drei Erosionen
3. eine Dilatation

Der geglättete Vordergrund wird nun zu Blobs[2] zusammengefasst und der höchste Punkt jedes Blobs als mögliche Hypothese für den höchsten Punkt einer Person in Betracht gezogen. Jedoch sind so auch noch viele falsche Hypothesen in der Menge der möglichen Merkmale enthalten. Beispielsweise ist es möglich, dass eine Person in mehrere Fragmente zerfällt (s. Abb. 4.2 (c)). Da die Kalibrierung im nächsten Schritt sehr robust gegenüber falschen Korrespondenzen ist, gilt diese Filterung hauptsächlich dem Beschleunigen der Berechnungen im nächsten Schritt. Der Aufwand der Kalibrierung steigt mit der Anzahl der übergebenen Merkmale.

Um falsche Hypothesen herauszufiltern, werden alle Hypothesen anhand der folgenden Kriterien überprüft und gegebenenfalls verworfen.

Minimale Fläche
Jeder Blob, dessen Fläche kleiner als 128 Pixel ist, wird verworfen.

Höchster Blob
Jeder Blob, dessen Boundingbox[3] sich unter einer anderen Boundingbox befindet, wird verworfen. Hierfür wird die Boundingbox nach links und rechts um 20 Pixel erweitert. Jede Boundingbox, die sich unter der erweiterten Boundingbox teilweise oder ganz befindet, wird verworfen. Bei zwei Personen (eine näher an der Kamera, eine weiter entfernt) kann der Fall auftreten, dass sie aus einer Kamerasicht übereinander erscheinen. In diesem Fall wird die untere Person ignoriert. Da die Videoaufnahmen jedoch über einen längeren Zeitraum gehen, fällt diese fehlende Korrespondenz nicht ins Gewicht.

Minimale Bewegung
Jeder Blob, in dessen Boundingbox sich weniger als zehn Pixel im Vergleich zum letzten Bild verändert haben, wird verworfen. Hierfür wird zusätzlich eine weitere Hintergrundsubtraktion parallel durchgeführt jedoch mit einer Lernrate von eins. Dadurch wird das letzte Bild als Hintergrund angenommen und der Vordergrundbereich entspricht der Differenz zwischen beiden Bildern. Sofern im Bereich [-20,0] bis [20,10] um den hypothetischen höchsten Punkt weniger als zehn Pixel im Differenzbild markiert wurden, wird die Hypothese verworfen.

Nicht am Rand
Jeder Blob, dessen Boundingbox weniger als zehn Pixel Abstand zum rechten oder linken Bildrand hat, wird verworfen. Dies ist notwendig, wenn eine Person noch nicht vollständig im Bild ist. In diesem Fall könnte der Kopf gar nicht auf dem Bild sein.

[2]Der Term "Blob" beschreibt im Allgemeinen etwas mit einer wagen bzw. nicht definierten Form. Bzgl. der Bildverarbeitung trifft jedoch auch folgende Definition zu: Eine Menge an Pixel, die eine Eigenschaft gemein haben, die auf alle umgebenden Pixel nicht zutrifft. (Hier: markiert als Vordergrund)

[3]Die Boundingbox eines Blobs ist das kleinste Rechteck, das alle Pixel des Blobs enthält.

Minimale Boundingbox
Jeder Blob, dessen Boundingbox-Breite oder -Höhe kleiner als 40 Pixel ist, wird verworfen. Zusammen mit der Überprüfung der Blob-Fläche wird hierdurch sichergestellt, dass fehlerhaft als Vordergrund markierte Bereiche verworfen werden. Dies geschieht beispielsweise durch Schattenwurf.

Die Auswirkungen dieser Regeln sind in Abbildung 4.2 zu sehen. Sofern auf der zweiten Aufnahme auch mindestens ein Merkmal gefunden wurde, werden alle möglichen Kombinationen aus den nicht verworfenen Hypothesen als mögliche Korrespondenzen zusammen mit der zugehörigen Bildnummer gespeichert. Abbildung 4.3 zeigt das Ergebnis der Merkmalsextraktion für die Aufnahmen aus Abbildung 4.1. In diesem Falle sind es zwischen Kamera 1 und Kamera 2 sechs Korrespondenzen, zwischen Kamera 1 und Kamera 3 sechs Korrespondenzen und Kamera 1 und Kamera 4 drei Korrespondenzen. Um den Aufwand der folgenden Kalibrierung zu verkleinern, werden nur die Korrespondenzen jedes fünften Bildes weitergegeben. Da zwischen einzelnen aufeinander folgenden Bildern nur kleine Änderungen zu detektieren sind, verfälscht diese Reduzierung der Merkmale nicht das Ergebnis, beschleunigt aber die Berechnungen.

4.1.2 Entzerren der Merkmalskoordinaten

Da durch die Linse einer Kamera Verzerrungen auftreten, müssen die Bildkoordinaten der ermittelten Merkmale vor der Kalibrierung entzerrt werden. Die Verzerrungen werden dabei nach dem Modell von [23] modelliert (s. Abschnitt 3.1.2). Die Entzerrung erfolgt in einem iterativen Verfahren, das sich startend vom ursprünglichen Bildpunkt $\vec{x_0} = (\mathbf{x}_0, \mathbf{y}_0)^T$ den entzerrten Koordinaten nähert. Die Näherung wird mit den Gleichungen (4.1) bis (4.4) berechnet. In dieser Arbeit werden fünf Iterationen verwendet, um die Koordinaten zu entzerren. Die so entzerrten Korrespondenzen werden an den nächsten Schritt des Kalibrierungssystems weitergeben.

$$r^2 = \mathbf{x}_{i-1}^2 + \mathbf{y}_{i-1}^2 \tag{4.1}$$

$$d_{\text{radial}} = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6 \tag{4.2}$$

$$\vec{x_\Delta} = \begin{pmatrix} 2\kappa_3 \mathbf{x}_{i-1}\mathbf{y}_{i-1} + \kappa_4 \left(r^2 + 2\mathbf{x}_{i-1}^2\right) \\ 2\kappa_4 \mathbf{y}_{i-1}\mathbf{x}_{i-1} + \kappa_3 \left(r^2 + 2\mathbf{y}_{i-1}^2\right) \end{pmatrix} \tag{4.3}$$

$$\vec{x_i} = \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix} = \frac{1}{d_{\text{radial}}} \left(\vec{x_0} - \vec{x_\Delta}\right) \tag{4.4}$$

4.2 Kalibrierung von Kamerapaaren

Dieser Schritt bestimmt die extrinsischen Parameter zwischen zwei Kameras K_1 und K_2 anhand der möglichen Punktkorrespondenzen aus der Merkmalsextraktion. Dabei werden der Translationsvektor und die Rotationsmatrix der zweiten Kamera K_2 relativ zur ersten Kamera berechnet, d.h. im Koordinatensystem von Kamera K_1. Zusätzlich kann dieser Translationsvektor nur bis auf einen Skalierungsfaktor genau berechnet werden. Damit ist die Richtung des Translationsvektors bekannt, aber nicht dessen Länge. Dies hängt damit

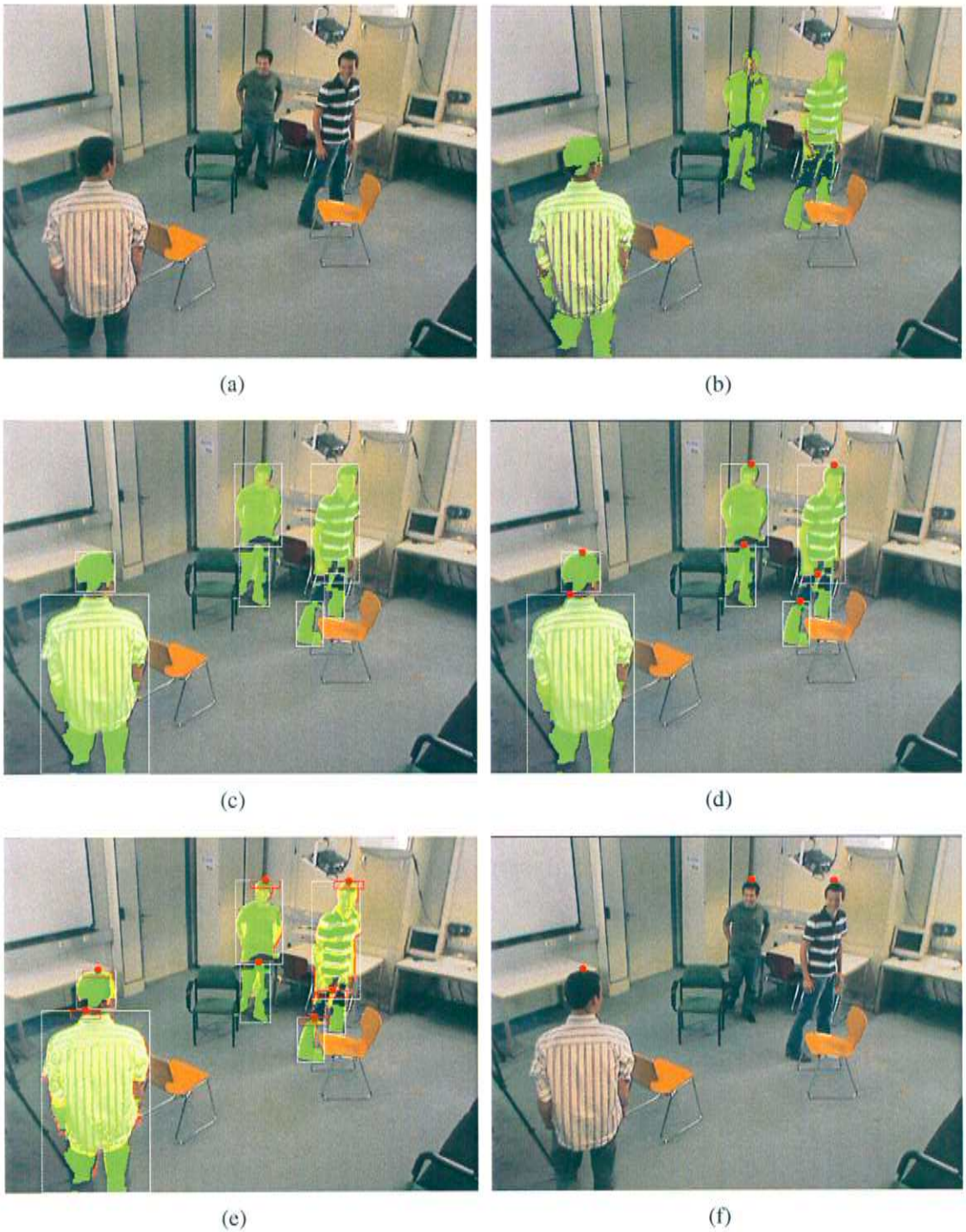


Abbildung 4.2: Die einzelnen Schritte der Merkmalsextraktion. (a) zeigt die Aufnahme einer Kamera. (b) zeigt das direkte Ergebnis der Hintergrundsubtraktion. Der Vordergrund wurde grün eingefärbt. (c) Der geglättete Vordergrund und die Boundingboxen der gefundenen Blobs (weiß eingezeichnet). (d) Die Menge der Hypothesen (rote Punkte). (e) Zusätzlich wurde die Bewegung zum letzten Bild in rot und gelb eingezeichnet. Innerhalb der roten Rechtecke wird nach dieser Bewegung gesucht. (f) Das Ergebnis der Merkmalsextraktion.

(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 4.3: Das Ergebnis der Merkmalsextraktion markiert durch weiße Punkte. Im Bild von Kamera 3 und 4 wurde eine Person nicht korrekt markiert, weil dessen zugehöriges Blob zu nahe am rechten bzw. linken Bildrand war.

zusammen, dass nur Punkte zwischen den Bildern für die Berechnungen verwendet werden, jedoch keine Längenangaben oder Distanzen. Dadurch ist nicht bekannt, in welchem Maßstab das betrachtete Objekt vorliegt. Es kann beispielsweise nicht festgestellt werden, ob ein Spielzeugmodell eines Flugzeugs oder ein richtiges Flugzeug betrachtet wird (eine gewisse Ähnlichkeit zwischen Modell und Original vorausgesetzt). Dieser Trick wird auch sehr oft in der Filmbranche eingesetzt, um kostspielige Aufnahmen im Kleinen zu reproduzieren und so einen Großteil der Kosten einzusparen.

Der hier verwendete Ansatz entspricht dem Selbstkalibrierungsansatz, da hier keine dreidimensionalen Positionen des betrachteten Objekts benötigt werden, sondern das Problem automatisch anhand von Punktkorrespondenzen gelöst wird. Jedoch wird in diesem Verfahren nicht vorausgesetzt, dass es sich um korrekte Punktkorrespondenzen handeln muss. Vielmehr wird die Menge aller möglichen Merkmalskombinationen betrachtet.

Durch die Kalibrierung werden die externen Parameter $\mathcal{T}_{K_1 \to K_2}$, dem Translationsvektor zu Kamera K_2 im Koordinatensystem von Kamera K_1, und $\mathcal{R}_{K_1 \to K_2}$, der Rotation von Kamera K_2 relativ zum Koordinatensystem von Kamera K_1, berechnet. In diesem Abschnitt werden diese beiden Parameter vereinfacht $\mathcal{T}$ und $\mathcal{R}$ benannt. Weiterhin ist es wichtig zu erwähnen, dass sämtliche Betrachtungen im dreidimensionalen Raum mit einem rechtshändigen Koordinatensystem durchgeführt werden.

Der verwendete Kalibrierungsalgorithmus entstand aus der Idee, dass bei korrekter Kalibrierung der Triangulationsfehler (siehe Kapitel 3.6) über alle Punktkorrespondenzen am Kleinsten ist. Hierfür kann man für feste Punktkorrespondenzen x, x' eine Funktion $f(\mathcal{R}, \mathcal{T}, x, x')$ definieren, die den Triangulationsfehler in Abhängigkeit von Rotation $\mathcal{R}$ und Translation $\mathcal{T}$ berechnet. Zur Lösung müsste *nur* die Summe der Triangulationsfehler für alle Punktkorrespondenzen $\sum_{\forall i} f(\mathcal{T}, \mathcal{R}, x_i, x_i')$ minimiert werden. Problematisch hierbei ist jedoch der Aufwand, da es sich immerhin um einen sechs-dimensionalen Suchraum handelt (drei Dimensionen für $\mathcal{T}$ und drei weitere Dimensionen für $\mathcal{R}$). Aus diesem Grund wird zuerst der Suchraum bzgl. seiner Dimensionen verkleinert. Anschließend wird eine Heuristik definiert, mit deren Hilfe die korrekten extrinsischen Parameter mit einer Suche ermittelt werden können.

4.2.1 Reduzierung des Suchraums bezüglich der Tranlsation

Der Translationsvektor $\mathcal{T}$ gibt die Position der zweiten Kamera relativ zur ersten Kamera an. Da bei diesem Verfahren Punktkorrespondenzen benutzt werden, kann dieser nur bis auf einen Skalierungsfaktor genau berechnet werden. Diese Tatsache kann ausgenutzt werden, um die Paramteranzahl zu reduzieren, indem der Suchraum bezüglich $\mathcal{T}$ auf alle Vektoren mit Länge eins beschränkt wird: $|\mathcal{T}| = 1$. Ohne Beschränkung der Allgemeinheit kann der Vektor $\mathcal{T} = (x, y, z)^T$ in die Form $\mathcal{T}_{\text{sphere}} = (r, \theta, \varphi)^T$ in Kugelkoordinaten transformiert werden. Da jedoch $|\mathcal{T}| = 1$ folgt daraus $r = 1$, und r muss nicht weiter beachtet werden. Daher wird die folgende vereinfachte Schreibweise verwendet: $\mathcal{T}' = (\theta, \varphi)^T$. Der Vektor $\mathcal{T}'_0 = (0, 0)^T$ entspricht dabei den dreidimensionalen Koordinaten $(0, 0, 1)^T$. θ entspricht einer Rotation um die Y-Achse, φ einer Rotation um die X-Achse. Abbildung 4.4 verdeutlicht den Zusammenhang. Der Wertebereich liegt dementsprechende für θ im Bereich $[-\pi : \pi[$ und für φ im Bereich $\left[-\frac{1}{2}\pi : \frac{1}{2}\pi\right]$. Die Umrechnung eines normierten Vektors $\mathcal{T}$ in $\mathcal{T}'$ zeigen die Gleichungen (4.5) und (4.6). Den umgekehrten Weg zeigt Gleichung (4.7).

$$\varphi = \operatorname{sgn}(-\mathcal{T}_y) \cdot \cos^{-1}\left(\frac{\operatorname{abs}(\mathcal{T}_z)}{\sqrt{\mathcal{T}_y^2 + \mathcal{T}_z^2}}\right) \tag{4.5}$$

$$\theta = \operatorname{sgn}(\mathcal{T}_x) \cdot \cos^{-1}\left(\frac{\mathcal{T}_z}{\sqrt{\mathcal{T}_x^2 + \mathcal{T}_z^2}}\right) \tag{4.6}$$

$$\mathcal{T} = \mathcal{R}_y(\theta) \cdot \mathcal{R}_x(\varphi) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.7}$$

Gleichung (4.5) berechnet dabei den Winkel zwischen $\mathcal{T}$ und der X-Z-Ebene während Gleichung (4.6) den Rotationswinkel um die Y-Achse zwischen $\mathcal{T}$ und der Z-Achse berechnet.

Dieses System wird auch gewöhnlich verwendet, um u.A. die Orientierung einer PTZ-Kamera zu beschreiben. Außerdem ist es relativ intuitiv zu benutzen. Im Beispiel der PTZ-Kamera entspricht θ dem Pan, einer Drehung der Kamera nach links bzw. rechts, und φ entspricht dem Tilt, einer Drehung der Kamera nach oben bzw. unten. Mit anderen

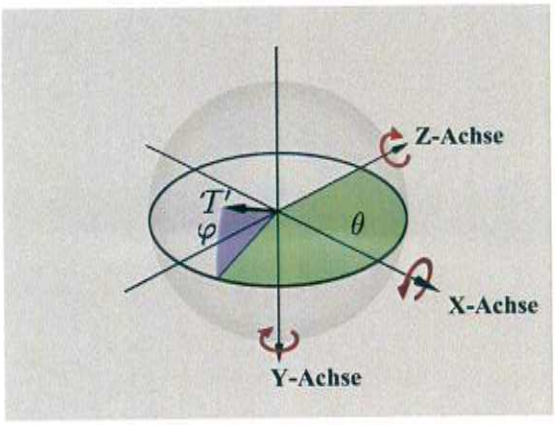


Abbildung 4.4: Der Translationsvektor $\mathcal{T}' = (160°, 30°)$ im Zusammenhang mit θ und φ.

Worten entspricht $\mathcal{T}'$ der Pan-Tilt-Ausrichtung von Kamera 1, so dass Kamera 2 direkt in der Sichtline von Kamera 1 liegt.

Mit dieser Umformung reduziert sich der Suchraum auf fünf Dimensionen (zwei für $\mathcal{T}'$ und drei für $\mathcal{R}$).

4.2.2 Reduzierung des Suchraums bezüglich der Rotation

Die Reduzierung des Suchraums bezüglich $\mathcal{R}$ hingegen, erfolgt auf komplexere Art. Hierfür wird zuerst der Aufbau der relativen Rotationsmatrix $\mathcal{R}$ und deren Zerlegung festgelegt. Jede Rotationsmatrix $\mathcal{R}$ kann dabei aus einer Multiplikation dreier Rotationsmatrizen, jeweils um die X- (Tilt), Y- (Pan), und Z-Achse (Roll), erzeugt werden. Umgekehrt ist auch eine Zerlegung in diese möglich.

$$\mathcal{R} = \mathcal{R}_{\text{pan}} \cdot \mathcal{R}_{\text{tilt}} \cdot \mathcal{R}_{\text{roll}} \tag{4.8}$$

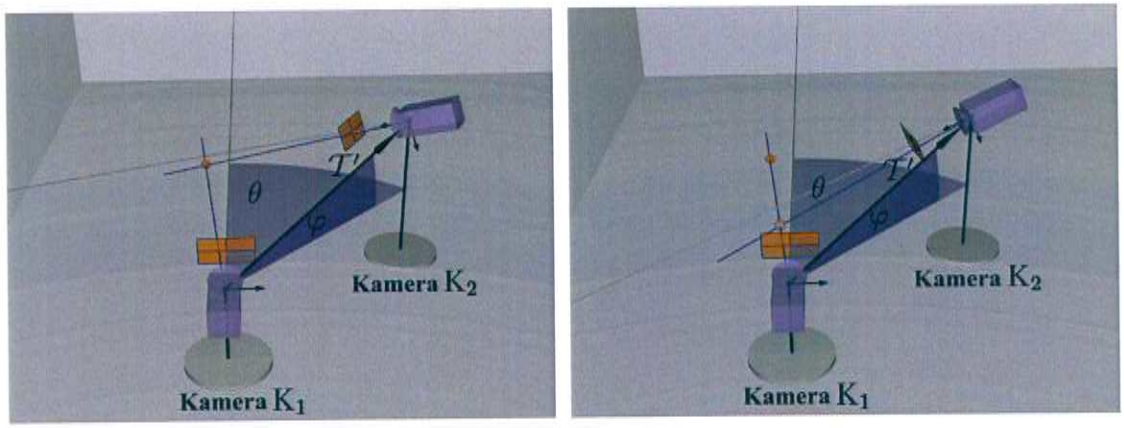
$$\mathcal{R}_{\text{pan}} = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix} \tag{4.9}$$

$$\mathcal{R}_{\text{tilt}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{pmatrix} \tag{4.10}$$

$$\mathcal{R}_{\text{roll}} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.11}$$

Im Folgenden wird die Tatsache genutzt, dass sich für korrekte $\mathcal{T}'$ und $\mathcal{R}$ die LOVs[4] von korrekten Punktkorrespondenzen auf jeden Fall schneiden. In diesem Fall ist der Triangulationsfehler gleich null.

[4]Die Line of View (kurz: LOV) entspricht der Sichtgeraden vom Kamerazentrum einer Kamera zu einem betrachteten Punkt.



(a) (b)

Abbildung 4.5: Für feste $\mathcal{T}'$ und $\mathcal{R}_{\text{roll}}$ gibt es mehrere Ausrichtungen von Kamera 2, bei der sich beide LOVs (blau) schneiden. (a) ist die korrekte Ausrichtung. (b) ist eine weitere Lösung.

Durch dieses Wissen lässt sich für gegebene relative Position $\mathcal{T}'$ und Eigendrehung um die Z-Achse $\mathcal{R}_{\text{roll}}$ von Kamera K_2 mit einer Menge von Punktkorrespondenzen die Eigendrehung um die Y-Achse $\mathcal{R}_{\text{pan}}$ und X-Achse $\mathcal{R}_{\text{tilt}}$ ermitteln. Dies wird durch eine Houghtransformation (s. Abschnitt 3.5) realisiert, da eine einzige Beobachtung (Korrespondenz) ein Indiz für eine Untermenge von Hypothesen im Parameterraum ist. Über mehrere Beobachtungen hinweg können somit die wahrscheinlichsten Hypothesen gefunden werden. Der Parameterraum der Houghtransformation besteht dabei aus den relativen Drehungen um die Y- und X-Achse $\mathcal{R}_{\text{pan}} \in [-\pi : \pi[$ und $\mathcal{R}_{\text{tilt}} \in \left[-\frac{1}{2}\pi : \frac{1}{2}\pi\right]$. Vereinfachend wird davon ausgegangen, dass sich Kamera K_1 im Ursprung befindet und entlang der Z-Achse ausgerichtet ist. Für eine gegebene Punktkorrespondenz muss Kamera K_2 so ausgerichtet werden, dass sich die zugehörigen LOVs von beiden Kameras schneiden. Sofern $\mathcal{T}'$ und $\mathcal{R}_{\text{roll}}$ fest vorgegeben sind, lassen sich alle möglichen Lösungen für $\mathcal{R}_{\text{pan}}$ und $\mathcal{R}_{\text{tilt}}$ ermitteln. Da es in diesem Fall zwei zu bestimmende Freiheitsgrade gibt, liefert eine Punktkorrespondenz keine eindeutige Lösung, sondern eine Menge von Lösungen, die aber auf einer Epipolarebene π liegen(s. Kap. 3.2). Abbildung 4.5 verdeutlicht das Problem.

Die Berechnung der jeweiligen Pan-Tilt-Winkelkombinationen für eine gegebene Punktkorrespondenz erfolgt dabei nach folgendem Schema:

1. Berechne LOV_1 von Kamera 1
2. Berechne LOV_2 von Kamera 2
3. Für jeden Punkt x auf LOV_1: drehe LOV_2 um die Pan und Tilt Achse, so dass gilt: LOV_2 zeigt auf x

Die Berechnung der $LOVs$ geschieht mit Hilfe der Formeln in Abschnitt 3.1.2. LOV_2 muss jedoch noch zusätzlich mit $\mathcal{R}_{\text{roll}}$ multipliziert werden, um die vorgegebene Eigenrotation um die Z-Achse von Kamera K_2, $\mathcal{R}_{\text{roll}}$, mit einzubeziehen. Die Lösungsmenge der berechneten Pan- und Tilt-Winkel für jede Punktkorrespondenz wird entsprechend in den Houghraum eingetragen. Da die LOV_1 in eine Richtung nicht begrenzt ist, wird folgender Trick angewendet, um den Berechnungsaufwand zu beschränken:

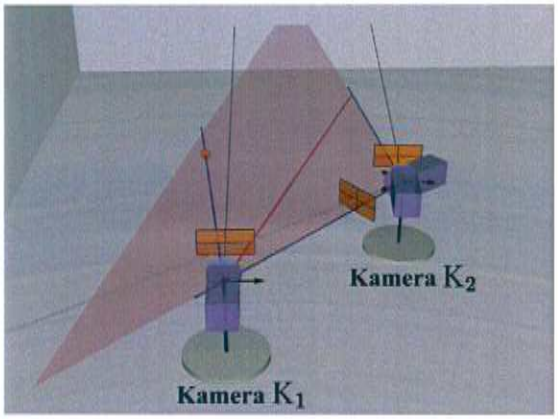


Abbildung 4.6: Die zwei möglichen Extremfälle für die Ausrichtung von Kamera K_2 und die Linie L_{join}, auf der alle Schnittpunkte für die Pan-Tilt-Winkel Berechnungen liegen.

Es gibt zwei Extrempunkte, an denen sich das betrachtete Objekt aufhalten kann.

- Das Objekt befindet sich auf LOV_1 direkt im Kamerazentrum von Kamera K_1
- Das Objekt befindet sich auf LOV_1 unendlich entfernt von Kamera K_1

Für den ersten Extremfall würde dies bedeuten, dass Kamera K_2 so ausgerichtet werden muss, dass LOV_2 durch das Kamerazentrum von Kamera K_1 verläuft. Im zweiten Extremfall muss Kamera K_2 so ausgerichtet werden, dass LOV_2 parallel zu LOV_1 verläuft. Alle weiteren Lösungen befinden sich dazwischen. Abbildung 4.6 veranschaulicht beide Extremfälle. Da alle Lösungen jedoch auf der Epipolarebene π (in Abb. 4.6 hellrot eingezeichnet) liegen, ist die Lösungsmenge äquivalent zu der, die entsteht, wenn LOV_2 mit einer anderen, geeignet gewählten Linie L_{join} geschnitten wird. Dabei verbindet die Linie L_{join} das Kamerazentrum von Kamera K_1 mit dem Punkt, der mit Abstand eins zum Zentrum von Kamera K_2 in Richtung LOV_1 liegt (rote Linie in Abb. 4.6).

Um die Berechnung etwas zu vereinfachen, wird Kamera 2 in den Ursprung verschoben. Damit liegt Kamera 1 bei Position $-\mathcal{T}$. Gleichung (4.12) definiert die Menge aller Schnittpunkte, die für die Houghtransformation berechnet werden müssen. Weiterhin bezeichnet lov_1 den normierten Vektor LOV_1.

$$L_{\text{join}} = -\mathcal{T} + s \cdot (lov_1 + \mathcal{T}) \, \forall s \in \mathbb{R} : 0 \leq s \leq 1 \tag{4.12}$$

Die Berechnung der Pan und Tilt Winkel für ein gebenes $\vec{x}$ aus L_{join} wird im Folgenden näher erläutert. Die Berechnungen können am besten auf der Oberfläche einer Einheitskugel mit Kamera 2 im Zentrum gezeigt werden. Hierfür wird der Vektor LOV_2 zu $\vec{v}_2^{\,n}$ und der Punkt $\vec{x}$ zu $\vec{x}^{\,n}$ normiert. Sowohl $\vec{v}_2^{\,n}$ als auch $\vec{x}^{\,n}$ befindet sich auf der Kugeloberfläche (Abb. 4.7 (b)). Ziel ist es, die Pan und Tilt Winkel zu berechnen, die $\vec{v}_2^{\,n}$ auf $\vec{x}^{\,n}$ drehen (Abb. 4.7 (a)). Hierfür wird $\vec{v}_2^{\,n}$ zuerst um die X-Achse (Tilt) gedreht, so dass $\vec{v}_2^{\,n}$ und $\vec{x}^{\,n}$ die gleiche Y-Komponente haben (Abb. 4.7 (c)). Anschließend wird $\vec{v}_2^{\,n}$ zusätzlich um die Y-Achse (Pan) gedreht, bis diese $\vec{x}^{\,n}$ entspricht (Abb. 4.7 (d)).

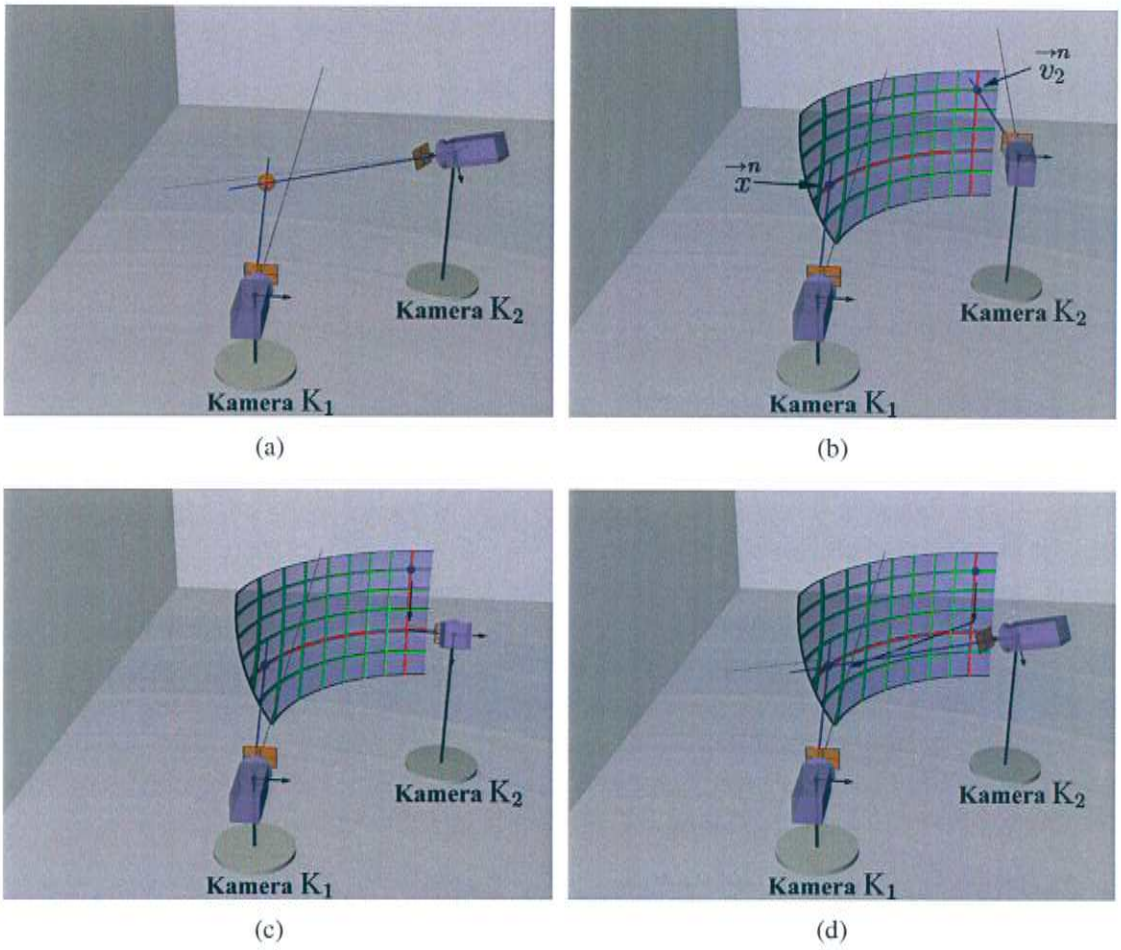


Abbildung 4.7: Die Berechnung der Pan- und Tilt-Winkel. (a) zeigt die korrekte Ausrichtung. (b) Kamera 2 in der Ursprungsaurichtung Pan = Tilt = 0. (b)→(c) Rotation um Tilt-Achse. (c)→(d) Rotation um Pan-Achse. Die Rotation wird auf einer Kugeloberfläche dargestellt.

Dieses Schema läßt sich wie folgt zusammenfassen:

1. Normiere LOV_2 zu $\vec{v}_2^{\,n}$
2. Berechne den normierten Richtungsvektor $\vec{x}^{\,n}$ von Kamera K_2 zu $\vec{x}$
3. Berechne den Tilt-Winkel mit Gleichung (4.13), so dass gilt: $\mathcal{R}_{\text{tilt}} \cdot \vec{v}_2^{\,n} = \vec{v}_2^{\,n'}$ mit $\vec{v}_{2_y}^{\,n'} = \vec{x}_y^{\,n}$
4. Berechne den Pan-Winkel mit Gleichung (4.14), so dass gilt: $\mathcal{R}_{\text{pan}} \cdot \vec{v}_2^{\,n'} = \vec{x}^{\,n}$

$$\beta = \sin^{-1}\left(\frac{\vec{v}_{2_y}^{\,n}}{\sqrt{\left(\vec{v}_{2_y}^{\,n}\right)^2 + \left(\vec{v}_{2_z}^{\,n}\right)^2}}\right) - \sin^{-1}\left(\frac{\vec{x}_y^{\,n}}{\sqrt{\left(\vec{v}_{2_y}^{\,n}\right)^2 + \left(\vec{v}_{2_z}^{\,n}\right)^2}}\right) \tag{4.13}$$

$$\alpha = \operatorname{sgn}\left(\vec{x}_x^{\,n}\right) \cdot \cos^{-1}\left(\frac{\vec{x}_z^{\,n}}{\sqrt{\left(\vec{x}_x^{\,n}\right)^2 + \left(\vec{x}_z^{\,n}\right)^2}}\right) - \operatorname{sgn}\left(\vec{v}_{2_x}^{\,n'}\right) \cdot \cos^{-1}\left(\frac{\vec{v}_{2_z}^{\,n'}}{\sqrt{\left(\vec{v}_{2_x}^{\,n'}\right)^2 + \left(\vec{v}_{2_z}^{\,n'}\right)^2}}\right) \tag{4.14}$$

Gleichung (4.13) berechnet hierfür zuerst den Tilt-Winkel (Rotation um X-Achse). Der linke Term drückt den Neigungswinkel zwischen $\vec{v}_2^{\,n}$ und der X-Z-Ebene aus. Der rechte Term drückt entsprechend den Winkel zwischen der X-Z-Ebene und $\vec{v}_2^{\,n}$ aus, wenn dieser die gleiche Y-Komponente hätte, wie $\vec{x}^{\,n}$. Die Differenz gibt den Winkel an, um den $\vec{v}_2^{\,n}$ um die X-Achse gedreht werden muss, damit gilt:

$$\mathcal{R}_{\text{tilt}} \cdot \vec{v}_2^{\,n} = \vec{v}_2^{\,n'} \text{ mit } \vec{v}_{2_y}^{\,n'} = \vec{x}_y^{\,n} \tag{4.15}$$

Somit befinden sich sowohl $\vec{v}_2^{\,n'}$ als auch $\vec{x}^{\,n}$ auf der Einheitskugel auf gleicher Höhe (s. Abbildung 4.7 (c)). Abschließend muss mit Gleichung (4.14) der Pan-Winkel (Rotation um die Y-Achse) bestimmt werden, damit $\vec{v}_2^{\,n'}$ auf $\vec{x}^{\,n}$ gedreht wird. Hierfür wird jeweils der Winkel zwischen $\vec{x}^{\,n}$ (linker Term) und $\vec{v}_2^{\,n'}$ (rechter Term) zur Z-Achse berechnet. Die Differenz zwischen beiden ergibt den Pan-Winkel. Da der Arcus Cosinus jedoch nur von 0 bis π definiert ist, wird anhand des Vorzeichens der X-Komponente unterschieden, ob der Winkel im Bereich $[-\pi : 0]$ oder im Bereich $[0 : \pi]$ liegt. Da für $\vec{x}^{\,n}$ und $\vec{v}_2^{\,n'}$ in Gleichung (4.14) gilt: $\vec{v}_{2_y}^{\,n'} = \vec{x}_y^{\,n}$ und sich beide Vektoren auf der Oberfläche einer Einheitskugel befinden, existiert immer eine Rotation um die Y-Achse, so daß gilt: $\mathcal{R}_{\text{pan}} \cdot \vec{v}_2^{\,n'} = \vec{x}^{\,n}$. Eine Lösung für Gleichung (4.13) ist jedoch nicht immer gegeben. Abbildung 4.8 zeigt die Problematik. Durch eine Pan-Tilt-Rotation kann ein beliebiger Vektor nicht in jeden anderen beliebigen Vektor überführt werden. Abbildung 4.8 zeigt den LOV-Vektor (a), alle erreichbaren Positionen durch eine Tilt-Rotation (b) und alle erreichbaren Positionen durch eine Pan-Tilt-Rotation (c). Eine Position ist genau dann nicht erreichbar, wenn deren Y-Komponente größer oder kleiner als $\pm\sqrt{\left(\vec{v}_{2_y}^{\,n}\right)^2 + \left(\vec{v}_{2_z}^{\,n}\right)^2}$ ist. Aus diesem Grund muss die folgende Bedingung beachtet werden:

$$\forall\, \vec{v}_2^{\,n}, \vec{x}^{\,n} : \vec{x}_y^{\,n} \leq \sqrt{\left(\vec{v}_{2_y}^{\,n}\right)^2 + \left(\vec{v}_{2_z}^{\,n}\right)^2} \tag{4.16}$$

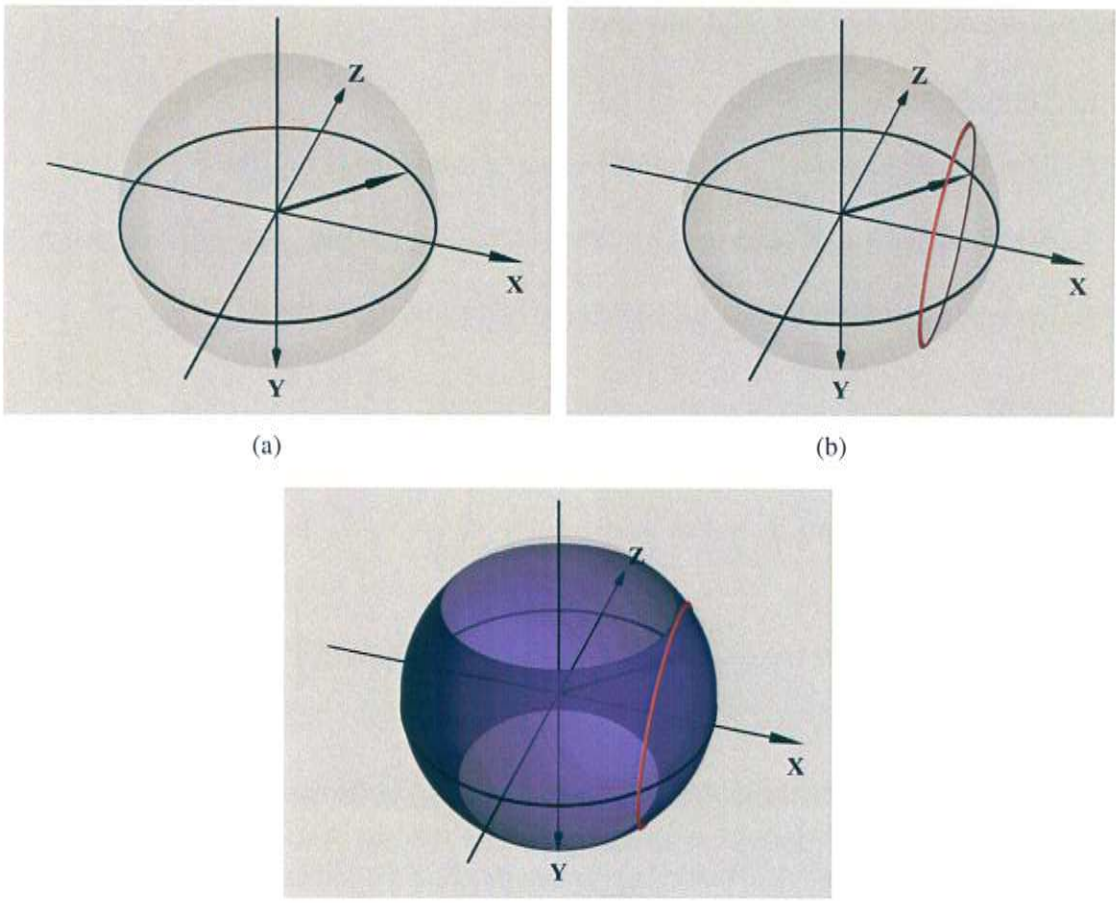


Abbildung 4.8: Problemfall bei der Tilt-Winkel Berechnung. (a) zeigt einen möglichen LOV-Vektor. (b) zeigt alle möglichen Positionen, die durch eine Drehung um die X-Achse erreicht werden (rot). (c) zeigt alle Möglichkeiten, die zusätzlich mit einer Drehung um die Y-Achse erreicht werden (blau). Dabei können nicht alle Positionen auf der Einheitskugel erreicht werden.

Andernfalls wird der Term in der rechten Arcus Sinus Funktion in Gleichung (4.13) größer eins oder kleiner minus eins. Da der Arcus Sinus für diesen Bereich nicht definiert ist, kann keine Lösung berechnet werden. Dies kommt jedoch nur vor, wenn sich $\mathcal{T}'$ und $\mathcal{R}_{\text{roll}}$ stark von den korrekten externen Parametern unterscheiden.

Die Houghtransformation lässt sich mit der vorgestellten Methode für feste $\mathcal{T}'$, $\mathcal{R}_{\text{roll}}$ und einer Menge an Punktkorrespondenzen durchführen. Hierfür wird für jede Punktkorrespondenz zuerst die Menge der zu betrachtenden Schnittpunkte L_{join} mit Gleichung (4.12) berechnet. Hierbei muss die Schrittweite, mit der die Gerade L_{join} diskretisiert wird, entsprechend der Auflösung des Houghraums gewählt werden. Wird eine zu große Schrittweite gewählt, entstehen Lücken zwischen den einzeln berechneten Pan-Tilt-Winkeln. Wird die Schrittweite zu klein gewählt, erhöht sich der Rechenaufwand bei gleichbleibenden Ergebnissen. In dieser Arbeit wurde für die Pan- und Tilt-Winkel Dimensionen des Houghraums eine Auflösung von 0.5° benutzt. Weiterhin wurde keine feste Schrittweite benutzt. Stattdessen wurde die Gerade L_{join} mit einer Ebene geschnitten, die in 0.5° Schritten, entsprechend der Auflösung des Houghraums, um den Koordinatenursprung (dem Kamerazentrum von Kamera K_2) gedreht wurde. Abbildung 4.9 zeigt dieses Vorgehen am Beispiel von Abbildung 4.6. Die Schnittpunkte entsprechen der zu betrachtenden

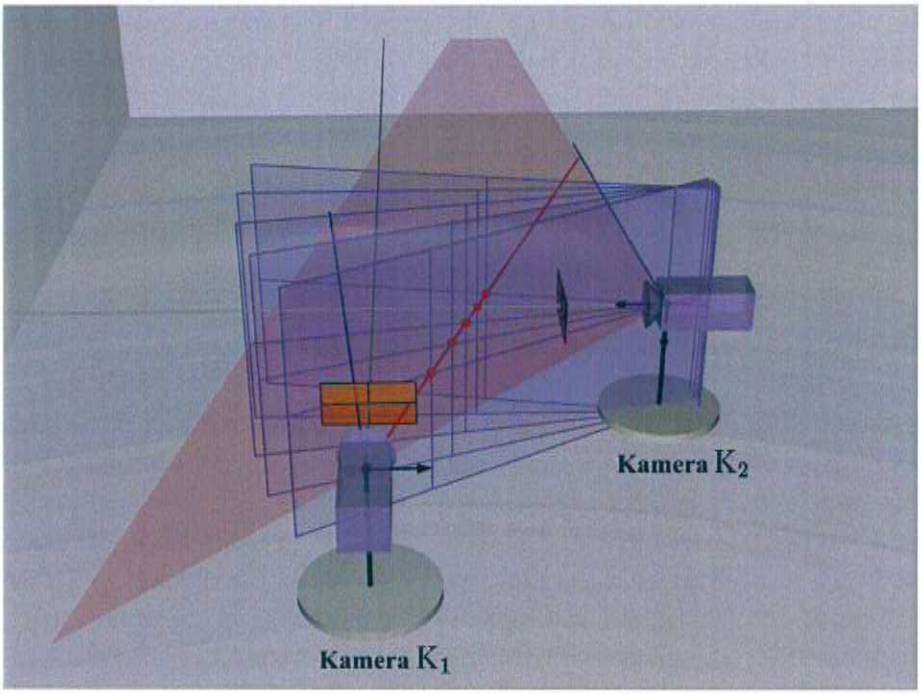


Abbildung 4.9: Berechnung der Punktmenge für die Houghtransformation (rote Punkte) durch Schnitte der rot eingefärbten Linie L_{join} mit einer rotierenden Ebene (blau eingefärbt) um das Zentrum von Kamera 2 berechnet.

Punktmenge X die für die Pan-Tilt-Winkel Berechnung benötigt wird. Mit diesem Verfahren konnten die erwähnten Lücken im Houghraum vermieden werden. Sollten trotzdem Lücken durch Diskretisierungsfehler entstehen, werden diese mit einer Verbindungslinie im Parameterraum gefüllt.

Für jeden der zu betrachtenden Punkte aus L_{join} wird somit die zugehörige Pan- und Tilt-Winkelkombination ausgerechnet und als Hypothese in den Houghraum eingetragen. Dieses Vorgehen wird für alle Punktkorrespondenzen wiederholt. Zum Schluss repräsentiert die Pan-Tilt-Hypothese, die am meisten berechnet wurde, den korrekten Pan-Tilt-Wert.

Das Ergebnis der Houghtransformation für eine korrekte Punktkorrespondenz wird in Abbildung 4.10 (a) gezeigt. Wie bereits erwähnt, reicht eine Punktkorrespondenz nicht für eine eindeutige Lösung aus. Durch das Hinzunehmen einer zweiten korrekten Punktkorrespondenz ergibt sich Abb. 4.10 (b). Jetzt ist das Ergebnis eindeutig. Die korrekte Lösung liegt auf dem Schnittpunkt der beiden Kurven, denn diese Hypothese wurde zweimal berechnet. Da aus der Merkmalsextraktion jedoch nicht bekannt ist, wie die einzelnen Punktmerkmale zueinander korrespondieren, gibt die Merkmalsextraktion alle möglichen Kombinationen zurück. Dies wird durch die Houghtransformation bewältigt mit der Annahme, dass die korrekten Punktkorrespondenzen im Merkmalsraum zu Häufungen bei der richtigen Parameterkombination führen, während Hypothesen aus falschen Korrespondenzen sich im Parameterraum eher streuen. Abbildung 4.10 (c) zeigt die Houghtransformation von allen vier Kombinationen der Merkmale, die für Abb. 4.10 (a) und (b) verwendet wurden. Der höchste Wert liegt nach wie vor auf dem korrekten Schnittpunkt. Zur besseren Verdeutlichung zeigt (d) eine Houghtransformation über 400 ausgewertete Bilder, jeweils mit zwei Objekten, deren Korrespondenzen nicht bekannt sind. Insgesamt wurde die Houghtransformation mit 1600 möglichen Punktkorrespondenzen bestimmt. Je heller der gezeichnete Punkt, desto häufiger wurde die entsprechende Hypothese berechnet. Das

Maximum ist durch einen roten Kreis gekennzeichnet und stimmt mit der korrekten Lösung überein. Das Maximum wird während den Berechnungen fortlaufend mitberechnet. Es ist daher keine zusätzliche Suche nach dem Maximum im Parameterraum notwendig.

Durch die Houghtransformation kann die Anzahl der freien Parameter in der Suche nach den externen Parametern um zwei weitere reduziert werden, da $\mathcal{R}_{\text{pan}}$ und $\mathcal{R}_{\text{tilt}}$ mit Hilfe von $\mathcal{T}$, $\mathcal{R}_{\text{roll}}$ und den extrahierten Punktkorrespondenzpaaren ermittelt werden können.

4.2.3 Suche der korrekten externen Parameter

Durch die Umwandlung der Translation $\mathcal{T}$ in Kugelkoordinaten und der Reduktion der Rotationsparameter mittels einer Houghtransformation wurde der ursprüngliche sechsdimensionale Suchraum auf drei Dimensionen verkleinert. Die noch zu suchenden Parameter sind nun $\theta \in [-\pi : \pi[$, $\varphi \in \left[-\frac{1}{2}\pi : \frac{1}{2}\pi\right]$ für den Translationsvektor $\mathcal{T}'$ und $\gamma \in [-\pi : \pi[$ für die Rotation $\mathcal{R}_{\text{roll}}$. Ziel der Suche ist es die Parameter zu finden, die der entsprechenden extrinsischen Kalibrierung entsprechen. In diesem Fall ist der Triangulationsfehler über alle Punktkorrespondenzpaare am kleinsten. Äquivalent dazu ist bei der Houghtransformation die Bewertung der entsprechenden Hypothese am höchsten, denn genau dann gehört diese Hypothese bei jeder korrekten Punktkorrespondenz zur berechneten Lösungsmenge. Dann entsprechen die Werte θ, φ und γ dieser Hypothese den extrinsischen Parametern der korrekten Kalibrierung.

Abgeleitet von diesem Wissen lässt sich eine Bewertungsfunktion für eine Parameterkombination θ, φ und γ definieren, nämlich der höchste Wert in der Akkumulatormatrix der zugehörigen Houghtransformation. Dieser Wert wird im Folgenden als Score einer Houghtransformation bezeichnet. Ziel der Suche ist es, diesen Score zu maximieren. Sofern die Parameter sich von der korrekten Kalibrierung unterscheiden, schneiden sich die Lösungsmengen im Houghraum nicht mehr an einer Stelle, d.h. der Score der Houghtransformation nimmt ab. Abhängig von den Punktkorrespondenzen schneiden sich die Kurven jedoch auch bei falschen Parametern an einigen Stellen und erhöhen so den Score. Jedoch bildet sich bei diesen Houghtransformationen nie ein genauso hohes Maximum wie bei der korrekten Lösung. Leider bilden sich dadurch aber dennoch lokale Maxima, die die Suche erschweren.

Da der Aufwand zu groß ist, den kompletten Parameterraum durchzurechnen, um so gezielt nach dem globalen Maximum zu suchen, mussten andere Suchverfahren versucht bzw. entworfen werden.

Der grundlegende Ansatz bestand in einer Bergsteigersuche (Hill Climbing) [1]. Die Bergsteigersuche verhält sich ähnlich zu einem Bergsteiger, der den Gipfel eines Berges erklimmen will. Sie versucht ausgehend von einem Startpunkt das Maximum einer Funktion zu finden. Hierfür werden alle Nachbarpunkte betrachtet, die einen höheren Wert haben, und in die Richtung des höchsten Anstiegs gegangen (Steepest Ascent Hill Climbing). Dies wird solange durchgeführt, bis kein Nachbarpunkt einen höheren Funktionswert hat. Jedoch verfängt sich dieser Ansatz leicht in lokalen Maxima oder bleibt auf einem Plateau stehen. Im Folgenden werden kurz die Probleme erklärt und bestehende Lösungsvorschläge gezeigt:

Falsche lokale Entscheidung

Im Suchraum kann die Situation auftreten, dass die Richtung des steilsten Anstiegs glo-

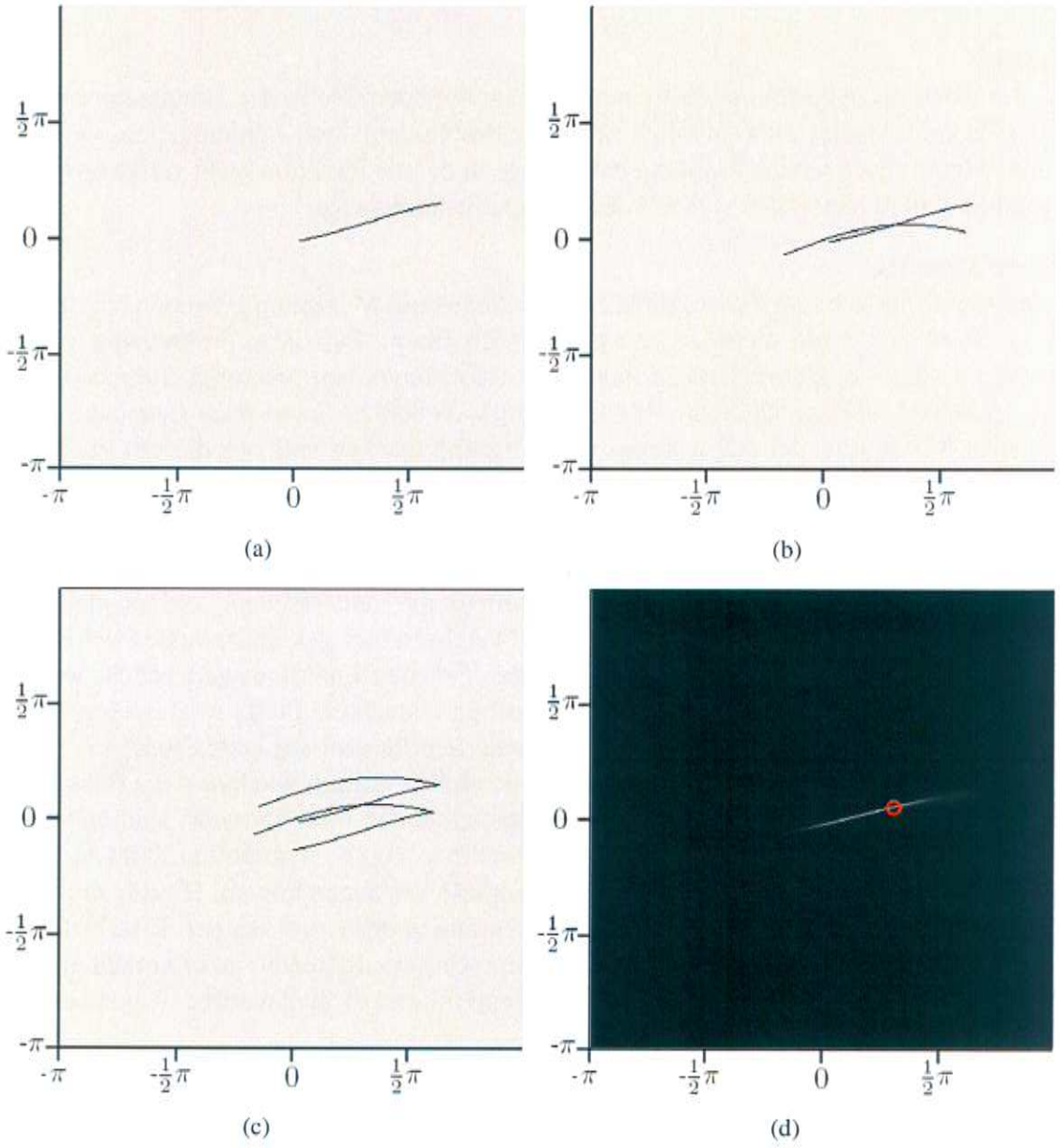


Abbildung 4.10: Beispiele der Houghtransformation: die X-Achse entspricht dem Pan-Winkel, die Y-Achse entspricht dem Tilt-Winkel. (a) eine korrekte Punktkorrespondenz (b) zwei korrekte Punktkorrespondenzen (c) zwei korrekte und zwei falsche Punktkorrespondenzen (d) 1600 Punktkorrespondenzen (800 korrekt, 800 falsch)

bal gesehen nicht zum gewünschten Maximum führt. Aus lokaler Sicht trifft der Algorithmus die richtige Entscheidung, geht aber global gesehen in die falsche Richtung. Der Ursprung dieses Problems liegt darin, dass der Algorithmus nur eine Lösung wählt, auch wenn mehrere zur Verfügung stehen. Zur Lösung des Problems müsste der Algorithmus zu der Stelle zurückgehen und die anderen möglichen Richtungen auch ausprobieren (Backtracking). Dies wird am einfachsten mit einer Liste erreicht, die alle möglichen Richtungen speichert und nacheinander diese Möglichkeiten ausprobiert.

Plateau
Sofern sich der Algorithmus auf einem Plateau befinden, bleibt der Standardansatz stehen, da kein Anstieg mehr möglich ist. Diese Problematik lässt sich umgehen, wenn der Algorithmus auch in eine Richtung gehen darf, in der die Funktion nicht verbessert wird, sondern gleich bleibt, d.h. sich auf dem Plateau weiterbewegt.

lokale Maxima
Der Algorithmus bleibt stehen, sobald dieser ein lokales Maximum gefunden hat. Jedoch handelt es sich dabei nicht um das globale Maximum. Für diese Problematik gibt es mehrere Ansätze. Beispielsweise kann der Algorithmus von mehreren Startpositionen aus gestartet werden (Shotgun Hill Climbing). Außerdem kann nach Erreichen eines lokalen Maximums der Algorithmus einen Sprung machen und von diesem Punkt aus eine neue Suche starten (Random Restart Hill Climbing).

In dieser Arbeit wurde ursprünglich der Hill Climbing Algorithmus, erweitert mit den Lösungsvorschlägen zu "Falsche lokale Entscheidung" und "Plateau", verwendet. Entsprechend dem Algorithmus wird zuerst die Nachbarschaft des Startpunktes betrachtet. Sofern die Nachbarpunkte einen besseren oder gleichen Funktionswert haben, werden diese in eine Liste `Todo` hinzugefügt. Der soeben betrachtete Punkt wird in eine weitere Liste `Processed` hinzugefügt. Im nächsten Schritt wird der erste Punkt aus `Todo` genommen und wiederum die Nachbarschaft betrachtet. Sofern wiederum die Funktionswerte der Nachbarschaft besser oder gleich dem aktuellen Funktionswert sind und diese nicht bereits untersucht wurden, werden die Punkte in `Todo` hinzugefügt. Dies ist wichtig, da der Algorithmus sich in einer Endlosschleife verfangen könnte. Hierfür muss nur überprüft werden, ob der aktuelle Punkt sich bereits in der `Processed` Liste befindet. Der Algorithmus stoppt, sobald in `Todo` keine weiteren Elemente mehr enthalten sind. Der höchste Punkt kann dabei wieder fortlaufend mitgespeichert werden. Algorithmus 1 zeigt den entsprechenden Pseudo-Code.

Der maximal gefunde Funktionswert wird in `maxValue` gespeichert. Das Ergebnis ist die Menge aller Punkte in `Processed` deren Wert `maxValue` ist.

Da der Suchraum dreidimensional ist, besteht die Nachbarschaft aus der $3 \times 3 \times 3$ Umgebung. Die Parameter θ, φ und γ müssen jedoch noch diskretisiert werden, entsprechend der gewünschten Genauigkeit. Bei einer Genauigkeit von 0.1° für alle Parameter braucht der Suchalgorithmus jedoch sehr lange. Aus diesem Grund wurde ein gestuftes Verfahren entwickelt. In diesem wird der Algorithmus iterativ mit adaptiven Schrittweiten gestartet. Dabei ist der Endpunkt des Durchlaufs $i - 1$ der Startwert des Durchlaufs i. Die Schrittweite wird bei jedem Durchlauf halbiert. Der Algorithmus terminiert sobald eine vorgegebene Schrittweite unterschritten wird.

In dieser Arbeit wird eine initiale Schrittweite von 5° verwendet. Der Algorithmus terminiert, sobald die Schrittweite unter $0{,}05^\circ$ fällt. Die Erfahrung zeigt, daß sich mit diesem

Algorithm 1 Suchalgorithmus

```
1: procedure SEARCH(start)                    ▷ Liste mit Startpunkten für die Suche
2:     Todo ← start
3:     MaxValue ← 0
4:     while NotEmpty(Todo) do
5:         x ← First(Todo)
6:         N ← Neighbors(x)
7:         for all n in N do
8:             if Eval(n) ≥ Eval(x) and NotIn(Processed,n) then
9:                 Insert(Todo,n)
10:            end if
11:            if Eval(x) ≥ maxValue then
12:                maxValue ← Eval(x)
13:            end if
14:            Delete(Todo,x)
15:            Insert(Processed,x)
16:        end for
17:    end while
18: end procedure
```

Ansatz je nach den verwendeten Punktkorrespondenzen die korrekte Lösung finden lässt, wenn der Startpunkt um ca. 10° vom korrekten Ziel abweicht. Dies erfordert jedoch eine zu strikte Eingrenzung des Suchraums.

Eine weniger strikte Eingrenzung des initialen Suchbereichs lässt sich erreichen, wenn zuvor in regelmäßigen Abständen Stichpunkte berechnet werden. Anhand dieser Stichpunkte lässt sich grob abschätzen, wo sich das gesuchte Maximum befindet. Hierfür wird ein 10° Raster verwendet. Leider ist auch hier der Rechenaufwand zu groß, um den ganzen Suchraum mit diesem Raster zu berechnen. Aus diesem Grund wird der Suchbereich schwach eingegrenzt. Die Eingrenzung gibt dabei ungefähr an, in welchem Bereich sich Kamera 2 von Kamera 1 aus gesehen befindet. Der Bereich kann dabei pro Parameter $\pm 45°$ umfassen. Beispielsweise $\theta \in \left[0 : \frac{1}{2}\pi\right]$, $\varphi \in \left[-\frac{1}{4}\pi : \frac{1}{4}\pi\right]$ bedeutet, dass Kamera 2 sich im Bereich rechts oben von Kamera 1 aus gesehen befindet und $\gamma \in \left[-\frac{1}{4}\pi : \frac{1}{4}\pi\right]$ bedeutet, dass Kamera 2 ungefähr aufrecht ausgerichtet ist. Diese Eingrenzung kann leicht angegeben werden, da nur grob die Position abgeschätzt werden muss. Der Aufwand der Berechnungen verringert sich jedoch enorm.

Aus den ermittelten 729 Stichpunkten (90° Suchbereich in 10° Schritten abgetastet für alle drei Dimensionen) werden alle diejenigen, deren Score 90% des Maximums überschreitet, als Startpunkte für die Suche ausgewählt. Die Suche wird mit oben genanntem Verfahren gesondert für jeden ermittelten Startpunkt durchgeführt. Dies entspricht dem oben erwähnten Shotgun Hill Climbing Algorithmus. Falls mehrere Ergebnisse zurückgeliefert werden, bspw. wenn die Lösung auf einem Plateau liegt und somit mehrere Punkte den gleichen Wert haben, wird der Mittelwert aller Lösungen benutzt.

Mit den ermittelten Zielparametern θ, φ und γ lassen sich die fehlenden Rotationsparameter α und β mit einer Houghtranformation bestimmen und somit die Translation $\mathcal{T}$ und die Rotation $\mathcal{R}$ berechnen.

Auf diese Weise werden die extrinsischen Kalibrierungen für alle Kamerakombinationen berechnet. Hierbei werden bewusst mehr Kalibrierungen berechnet, als theoretisch notwendig sind, um das Kameranetzwerk aufzuspannen. Dadurch kann später eine schlechtere Kalibrierung verworfen und stattdessen eine bessere benutzt werden, wodurch sich das Gesamtergebnis verbessert.

4.3 Aufspannen des Kameranetzwerkes

Nachdem die relativen Kalibrierungen zwischen allen Kameras berechnet wurden, wird folgend das Kameranetzwerk aufgespannt. Hierfür werden die paarweisen Kalibrierungen um ihre Inverse erweitert. Anschließend werden aus diesen Kalibrierungen alle möglichen Kameranetzwerke aufgebaut, indem die einzelnen Kamerapositionen mit Hilfe der Kalibrierungen trianguliert werden, und bewertet. Das beste Kameranetzwerk wird schließlich als Lösung ausgewählt. Dieses ist jedoch noch relativ zur ersten Kamera, mit der das Netzwerk aufgebaut wurde, dargestellt. Für die Skalierung und Transformation in das Weltkoordinatensystem sind jedoch der Abstand zwischen zwei beliebigen Kameras und die Position und Ausrichtung einer beliebigen Kamera notwendig. Sofern diese vorhanden sind, wird das Netzwerk in das Weltkoordinatensystem transformiert und auf die richtige Größe skaliert.

4.3.1 Ausnutzung der Redundanz bei paarweisen Kalibrierungen

Nach der automatischen Kalibrierung liegen für jedes Kamerapaar zwei Kalibrierungen vor, beispielsweise von Kamera K_1 nach Kamera K_2 und von Kamera K_2 nach Kamera K_1. Hierbei handelt es sich um eine Redundanz, da sich eine Kalibrierung invertieren lässt. Für den allgemeinen Fall von Kamera K_i und Kamera K_j ist:

$$\mathcal{R}_{K_j \to K_i} = inv\left(\mathcal{R}_{K_i \to K_j}\right) = \mathcal{R}^T_{K_i \to K_j} \tag{4.17}$$

$$\mathcal{T}_{K_j \to K_i} = inv\left(\mathcal{T}_{K_i \to K_j}\right) = -1 \cdot \mathcal{R}^T_{K_i \to K_j} \cdot \mathcal{T}_{K_i \to K_j} \tag{4.18}$$

Eine invertierte Kalibrierung entspricht der Kalibrierung für die entgegengesetzte Richtung. Da die Kalibrierungen vom vorherigen Schritt jedoch fehlerbehaftet sind, entspricht im Allgemeinen die Kalibrierung von Kamera K_1 nach Kamera K_2 nicht der invertierten Kalibrierung von Kamera K_2 nach Kamera K_1. Obwohl für beide Kalibrierungen die gleichen Punktkorrespondenzen verwendet wurden, weichen die Ergebnisse voneinander ab, bedingt durch die initiale Rasterung des Suchraums, der resultierenden Startwerte oder Fehler der Suche. Die Redundanz wird in diesem Schritt bewusst erzeugt, um dem folgenden Aufbau der Kameranetzwerke mehr Möglichkeiten zu bieten.

Auf diese Weise wird für jede Kalibrierung ihre Inverse berechnet und in die Menge der vorhandenen Kalibrierungen hinzugefügt. Die somit erzeugten zusätzlichen zwei Kalibrierungen pro Kamerapaar werden an den nächsten Schritt weitergegeben.

4.3.2 Aufbau des Kameranetzwerks

Der folgende Abschnitt beschreibt an einem Beispiel das Vorgehen zum Aufbau eines Kameranetzwerkes aus paarweisen Kalibrierungen. Das Netzwerk wird dabei iterativ aufgebaut, indem die einzelnen Positionen der Kameras trianguliert werden.

Die erste Kamera (K_1), die in das Netzwerk hinzugefügt wird, wird im Ursprung des Weltkoordinatensystems positioniert und entlang der Z-Achse ausgerichtet (Abb. 4.11 (a)).

$$\mathcal{T}_{K_1} = (0,0,0)^T \tag{4.19}$$

$$\mathcal{R}_{K_1} = I \tag{4.20}$$

Als Nächstes wird Kamera K_2 hinzugefügt. Anhand der Kalibrierung von K_1 zu K_2 steht die Richtung $\mathcal{T}_{K_1 \to K_2}$ fest, in der sich K_2 von K_1 aus gesehen befindet, jedoch nicht der Abstand. Da dieser aber nicht bestimmt werden kann, wird K_2 mit Abstand eins zu K_1 positioniert. Die Ausrichtung von K_2 entspricht $\mathcal{R}_{K_1 \to K_2}$ aus der Kalibrierung von K_1 zu K_2 (Abb. 4.11 (b)).

$$\mathcal{T}_{K_2} = \mathcal{T}_{K_1 \to K_2} \tag{4.21}$$

$$\mathcal{R}_{K_2} = \mathcal{R}_{K_1 \to K_2} \tag{4.22}$$

Die Position jeder weiteren Kamera kann trianguliert werden. In diesem Beispiel wird K_3 mithilfe von K_1 und K_2 trianguliert. Hierfür ist die Position von K_1 und K_2 und die Richtung, in der sich K_3, aus der Sicht von K_1 und K_2, befindet, notwendig. Die Position von K_1 und K_2 im Weltkoordinatensystem sind bekannt. Der Richtungsvektor von K_1 zu K_3 entspricht $\mathcal{T}_{K_1 \to K_3}$, da das System von Kamera K_1 dem Weltkoordinatensystem entspricht. Der Richtungsvektor von K_2 zu K_3 im Weltkoordinatensystem wird mit Gleichung (4.23) berechnet werden, da $\mathcal{T}_{K_2 \to K_3}$ im Koordinatensystem von K_2 vorliegt.

$$\vec{v} = \mathcal{R}_{K2} \cdot \mathcal{T}_{K2 \to K3} \tag{4.23}$$

Mit diesem Wissen lassen sich die folgenden Geradengleichungen für die Triangulation von Kamera K_x aus Kamera K_i und Kamera K_j definieren (Abb. 4.11 (c)).

$$L_i = \mathcal{T}_{K_i} + s \cdot (\mathcal{R}_{K_i} \cdot \mathcal{T}_{K_i \to K_x}) \tag{4.24}$$

$$L_j = \mathcal{T}_{K_j} + t \cdot (\mathcal{R}_{K_j} \cdot \mathcal{T}_{K_j \to K_x}) \tag{4.25}$$

Aus diesen beiden Geraden lässt sich die Position von Kamera K_x triangulieren. Im Idealfall schneiden sich die beiden Geraden an der Position von Kamera K_x. Da die Kalibrierungen jedoch fehlerbehaftet sind, wird die Position von K_x durch den Punkt x' approximiert, der den kleinsten Abstand zu beiden Triangulationsgeraden hat (Abb. 4.11 (d)).

Die Ausrichtung von Kamera K_x kann abschließend auf zwei Arten berechnet werden: über Kamera K_i oder über Kamera K_j. Die Gleichung (4.26) ist jedoch für beide Fälle einsetzbar.

$$\mathcal{R}_{K_x} = \mathcal{R}_{K_i} \cdot \mathcal{R}_{K_i \to K_x} \tag{4.26}$$

Auf diese Weise lassen sich beliebig viele Kameras in ein Kameranetzwerk einfügen.

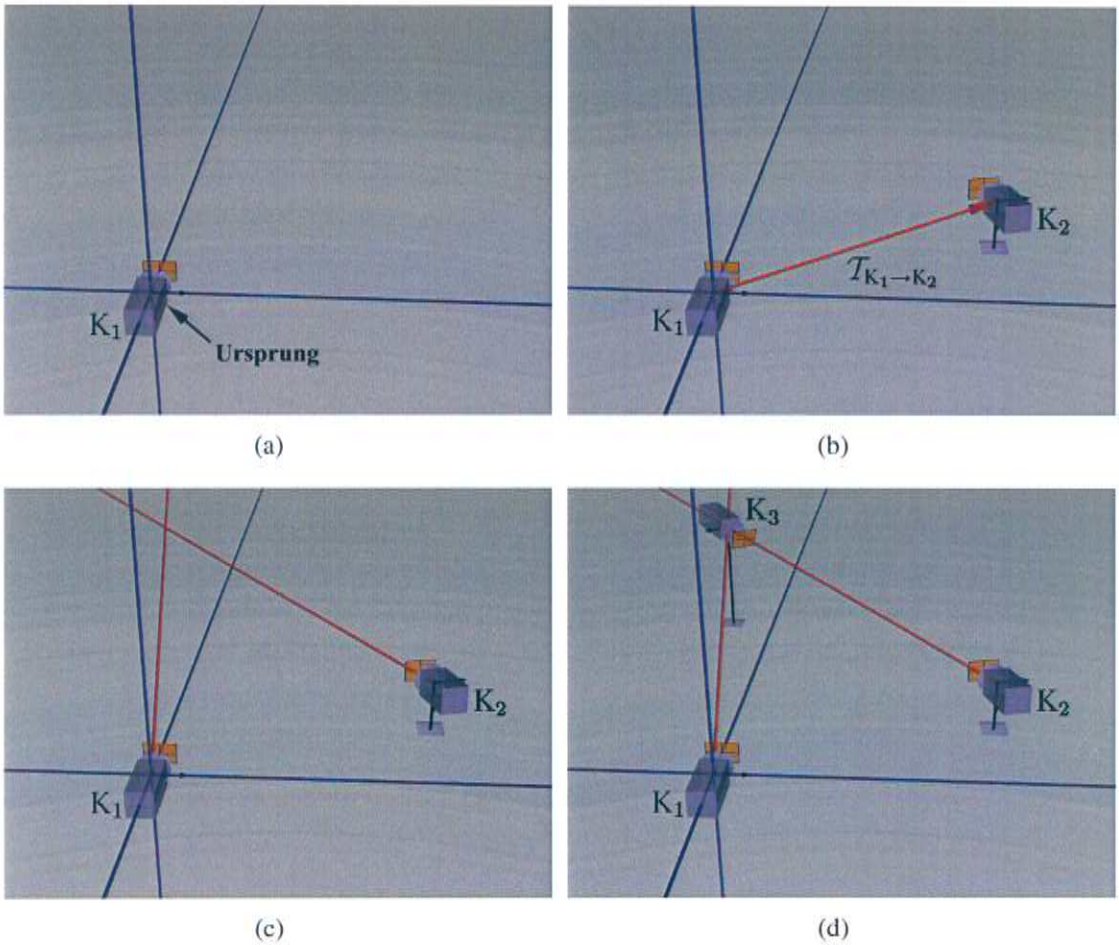


Abbildung 4.11: Triangulation der Kamerapositionen: (a) K_1 liegt im Ursprung, (b) die Position K_2 wird mit $\mathcal{T}_{K_1 \to K_2}$ berechnet, (c) die Triangulationsgeraden zur Berechnung der Position von K_3, (d) die Position von K_3.

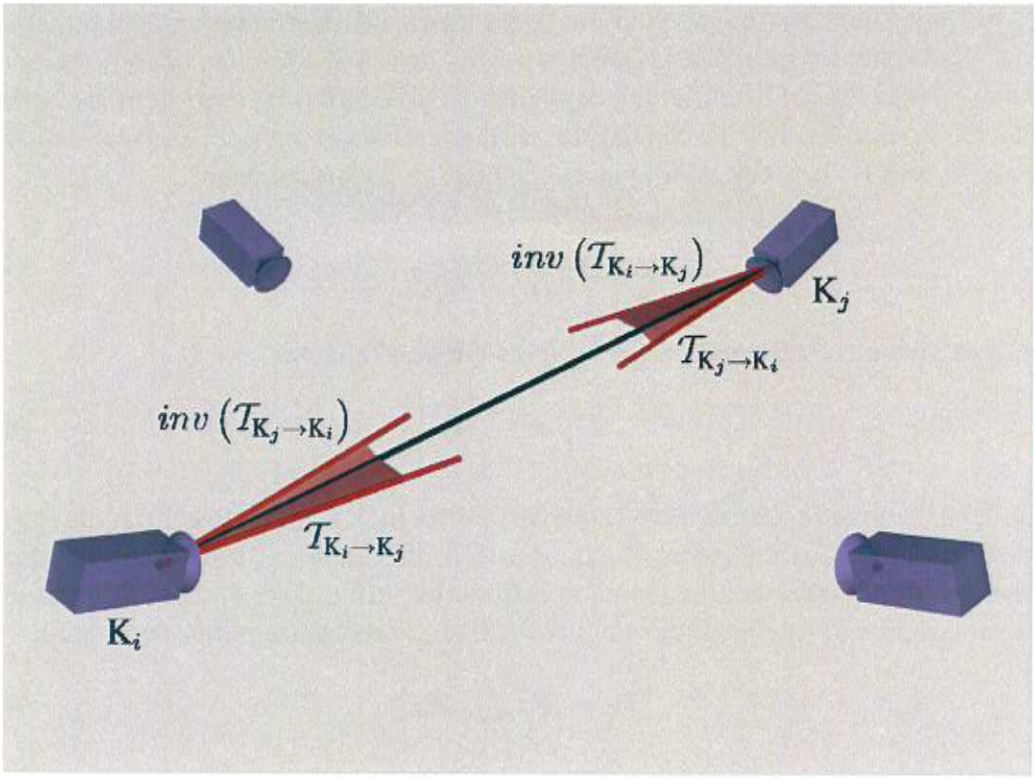


Abbildung 4.12: Der Fehler zwischen der Verbindungsgerade zwischen K_i und K_j und den Triangulationsgeraden aus $\mathcal{T}_{K_i \to K_j}$, $inv\left(\mathcal{T}_{K_i \to K_j}\right)$, $\mathcal{T}_{K_j \to K_i}$ und $inv\left(\mathcal{T}_{K_j \to K_i}\right)$.

4.3.3 Bewertung des Kameranetzwerks

Die Problematik beim Aufbau des Kameranetzwerkes liegt darin, dass ein Fehler am Anfang des Kameranetzwerkaufbaus sich auf nachfolgende Berechnungen unter Umständen stark negativ auswirkt. Aus diesem Grund wird die endliche Menge aller Möglichkeiten zum Aufbau des Kameranetzwerkes betrachtet und jedes resultierende Netz bewertet. Anhand der Bewertung ist es möglich das Kameranetzwerk auszuwählen, das am wahrscheinlichsten der realen Kamerakonfiguration entspricht.

Für die Bewertung eines Kameranetzwerkes werden jeweils zwei Kameras K_i und K_j betrachtet. Anhand der paarweisen Kalibrierungen können jeweils zwei Triangulationsgeraden (s. Gleichung 4.24) von einer zur anderen Kamera berechnet werden. Einmal mit der Kalibrierung von Kamera K_i nach K_j und einmal mit der inversen Kalibrierung von K_j nach K_i. Abbildung 4.12 verdeutlicht dieses Vorgehen. Zusammen mit der Verbindungsgeraden zwischen den Positionen von Kamera K_i und Kamera K_j lässt sich ein Winkelfehler berechnen. Hierfür wird der Winkel zwischen der Verbindungsgerade und jeder der beiden Triangulationsgeraden ermittelt. Auf diese Art und Weise werden alle Kamerakombinationen bewertet und der Fehler aufsummiert. Das Kameranetzwerk mit dem kleinsten Fehler wird als endgültige Lösung bestimmt.

4.3.4 Skalierung und Transformation ins Weltkoordinatensystem

Das Kameranetzwerk ist noch relativ zu Kamera K_1 aufgebaut und ist auch noch nicht korrekt skaliert. Der Skalierungsfaktor und die Transformation ins Weltkoordinatensystem ist

jedoch nur mit Zusatzwissen möglich. Im Folgenden wird die Skalierung und transformation ins Weltsystem beschrieben. Dabei wird für den skalierten Translationsvektor die Notation $\mathcal{T}''$ und für die Kalibrierung bezüglich des Weltkoordinatensystem die Notation $\mathcal{T}^*$ und $\mathcal{R}^*$ verwendet. Für die Skalierung wird der Abstand d zwischen zwei beliebigen Kameras K_i und K_j benötigt. Der Skalierungsfaktor s entspricht dann:

$$s = \frac{d}{|\mathcal{T}_{\mathrm{K}_i} - \mathcal{T}_{\mathrm{K}_j}|} \tag{4.27}$$

Damit lässt sich das Netzwerk auf die korrekte Größe skalieren.

$$\mathcal{T}''_{\mathrm{K}_i} = s \cdot \mathcal{T}_{\mathrm{K}_i} \tag{4.28}$$

Für die Transformation des Kameranetzwerkes ins Weltkoordinatensystem ist die Position und die Ausrichtung der Kamera $\mathrm{K}_{\mathrm{origin}}$, die sich im Ursprung des Kameranetzwerkes befindet, im Weltkoordinatensystem nötig. Mit diesen Informationen lässt sich das ganze Kameranetzwerk mit folgender Formel ins Weltkoordinatensystem transformieren:

$$\mathcal{R}^*_{\mathrm{K}_i} = \mathcal{R}_{\mathrm{K}_{\mathrm{origin}}} \cdot \mathcal{R}_{\mathcal{K}_i} \tag{4.29}$$

$$\mathcal{T}^*_{\mathrm{K}_i} = \left(\mathcal{R}_{\mathrm{K}_{\mathrm{origin}}} \cdot \mathcal{T}''_{\mathrm{K}_i}\right) + \mathcal{T}_{\mathrm{K}_{\mathrm{origin}}} \tag{4.30}$$

Falls von der Kamera im Ursprung die Position und Ausrichtung nicht zur Verfügung stehen, muss das Kameranetzwerk so transformieren werden, dass eine andere Kamera sich im Ursprung befindet, von der die Position und Ausrichtung bekannt ist. Danach lässt sich mit Gleichung (4.29) und (4.30) das Kameranetzwerk ins Weltkoordinatensystem transformieren. Mit den folgenden Formeln lässt sich ein bestehendes Kameranetzwerk so transformieren, dass Kamera K_x sich im Ursprung befindet.

$$\mathcal{R}_{\mathrm{K}_i} = \mathcal{R}^T_{\mathrm{K}_x} \cdot \mathcal{R}_{\mathrm{K}_i} \tag{4.31}$$

$$\mathcal{T}_{\mathrm{K}_i} = \mathcal{R}^T_{\mathrm{K}_x} \cdot (\mathcal{T}_{\mathrm{K}_i} - \mathcal{T}_{\mathrm{K}_x}) \tag{4.32}$$

Auf diese Weise lässt sich das Kameranetzwerk so transformieren, dass es den realen Gegebenheiten entspricht.

5 Experimente und Validierung

Im Folgenden wird das in dieser Arbeit entwickelte Kalibrierungssystem anhand von Experimenten evaluiert. Hierfür wurden drei Testaufnahmen im Smartroom aufgezeichnet, mit denen die Kalibrierung der Kameras berechnet wurde. Zuerst werden die paarweisen Kalibrierungen betrachtet und mit den manuell ermittelten Parametern verglichen. Anschließend wird das daraus generierte Kameranetzwerk untersucht und der Triangulations- und Projektionsfehler bestimmt. Die Ergebnisse werden dabei immer mit einer manuellen Kalibrierung verglichen.

Abbildung 5.1 zeigt eine schematische Darstellung des Smartrooms und die Position der zu kalibrierenden vier Kameras. Die Kameras sind dabei an den Ecken des Raumes auf einer Höhe von ca. 2,6 Meter montiert und größtenteils überlappend ausgerichtet. Jede Kamera ist dabei über einen Firewire-Bus an einen eigenen Rechner (Pentium 4, 3GHz, 1GB Ram) angeschlossen.

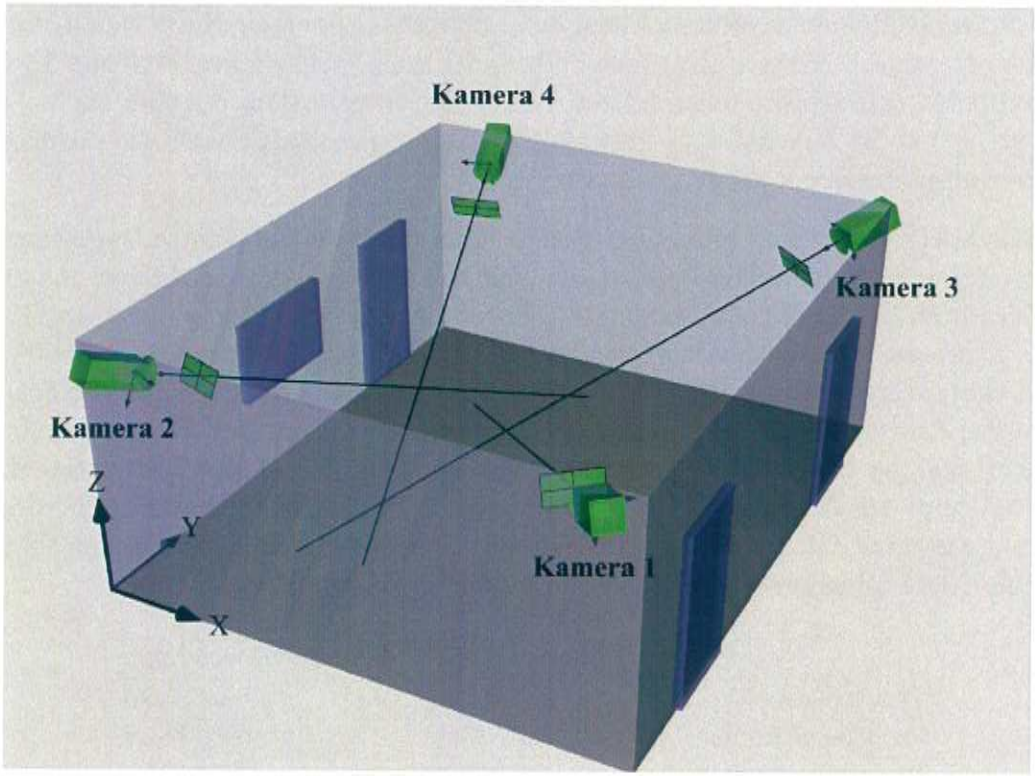


Abbildung 5.1: Die schematische Darstellung des Smartrooms und der vier Kameras.

Um die Ergebnisse bewerten zu können, werden diese immer mit einer manuellen Kalibrierung verglichen. Hierfür wurde im Vorfeld sowohl die intrinsische als auch die extrinsische Kalibrierung mittels traditioneller Verfahren vorgenommen und überprüft. Diese

(a) (b)

Abbildung 5.2: Beispielbilder der Aufnahmen für die intrinsische (a) und extrinsische (b) Kalibrierung für die Grundwahrheit.

Daten werden als Grundwahrheit betrachtet und dienen als Vergleichswerte zu den ermittelten Ergebnissen der Experimente. Die intrinsische Kalibrierung wurde mit Hilfe eines Schachbretts der Größe von 84cm×54cm vorgenommen. Es wurden dabei für jede Kamera gesondert etwa zwölf Bilder aufgezeichnet, in denen das Schachbrett jeweils aus einer anderen Perspektive zu sehen ist (Abb. 5.2 (a)). Anhand dieser Bilder konnte die intrinsische Kalibrierung für alle vier Kameras durchgeführt werden. Anschließend wurde mit einem 3m×3m großen Schachbretteppich die extrinsische Kalibrierung durchgeführt (Abb. 5.2 (b)). Der Nullpunkt des Raumes liegt dabei direkt in der unteren Nord-West-Ecke und die Koordinatenachsen orientieren sich entlang der Raumgrenzen. In Abbildung 5.1 liegt der Nullpunkt in der linken unteren Ecke. Die X-Achse verläuft nach rechts, die Y-Achse nach vorne und die Z-Achse nach oben. Sowohl die intrinsische als auch die extrinsische Kalibrierung wurden mit [5] durchgeführt.

Zusätzlich wurden im Smartroom drei Tische aufgestellt an deren Kanten insgesamt acht Marker befestigt wurden. Die Position der Marker wurde manuell vermessen und damit die Grundwahrheit verifiziert. Anhand dieser Marker wird die Güte eines Kameranetzwerks gemessen. Zum einen wird anhand der Markierungen im Kamerabild die Positionen der Marker im dreidimensionalen Raum trianguliert und die Differenz zur realen Position betrachtet. Zum anderen werden die Positionen der Marker in die einzelnen Kamerabilder projiziert und der Unterschied in Pixel zur tatsächlichen Bildposition gemessen. Abbildung 5.3 zeigt den Aufbau. Mit diesen Daten wurde zuerst der Fehler der Referenzkalibrierung gemessen. Die Ergebnisse sind Tabelle 5.1 zu entnehmen. In Abbildung 5.4 wird zusätzlich der Projektionsfehler der Referenzkalibrierung angezeigt.

	Mittlerer Fehler	Standardabweichung
Triangulation	11,53 mm	3,18 mm
Projektion Kamera 1	1,22 Pixel	0,60 Pixel
Projektion Kamera 2	1,84 Pixel	0,58 Pixel
Projektion Kamera 3	2,70 Pixel	0,55 Pixel
Projektion Kamera 4	1,56 Pixel	0,54 Pixel

Tabelle 5.1: Der Triangulations- und Projektionsfehler der Referenzkalibrierung.

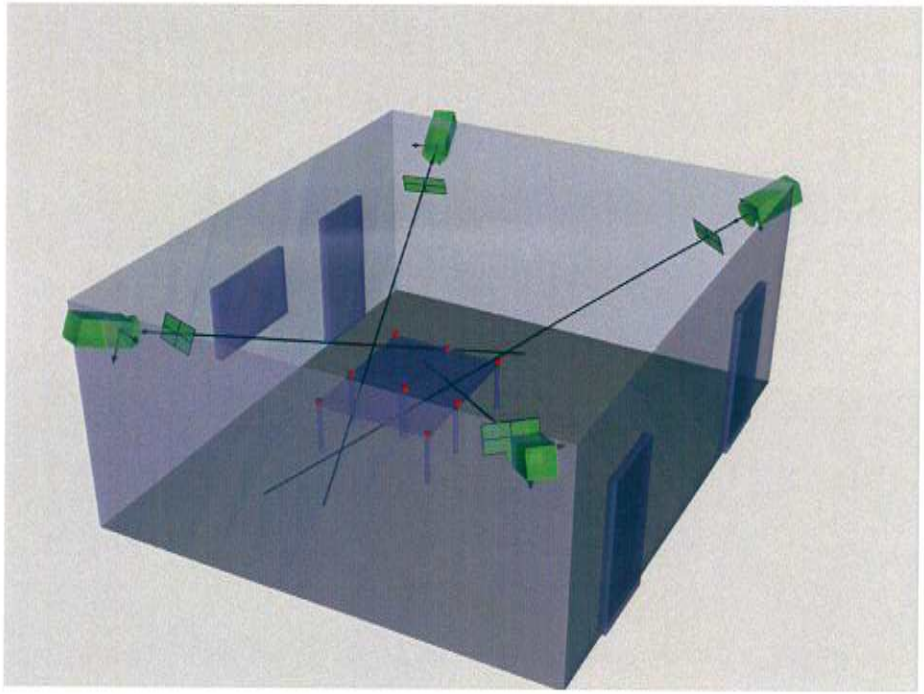


Abbildung 5.3: Die schematische Darstellung der zur Messung des Triangulations- und Projektionsfehlers benötigten Markierungen (rot).

Die Grundwahrheit wird außerdem dazu verwendet, die aufgebauten Netzwerke aus den Experimenten in das Weltkoordinatensystem zu transformieren.

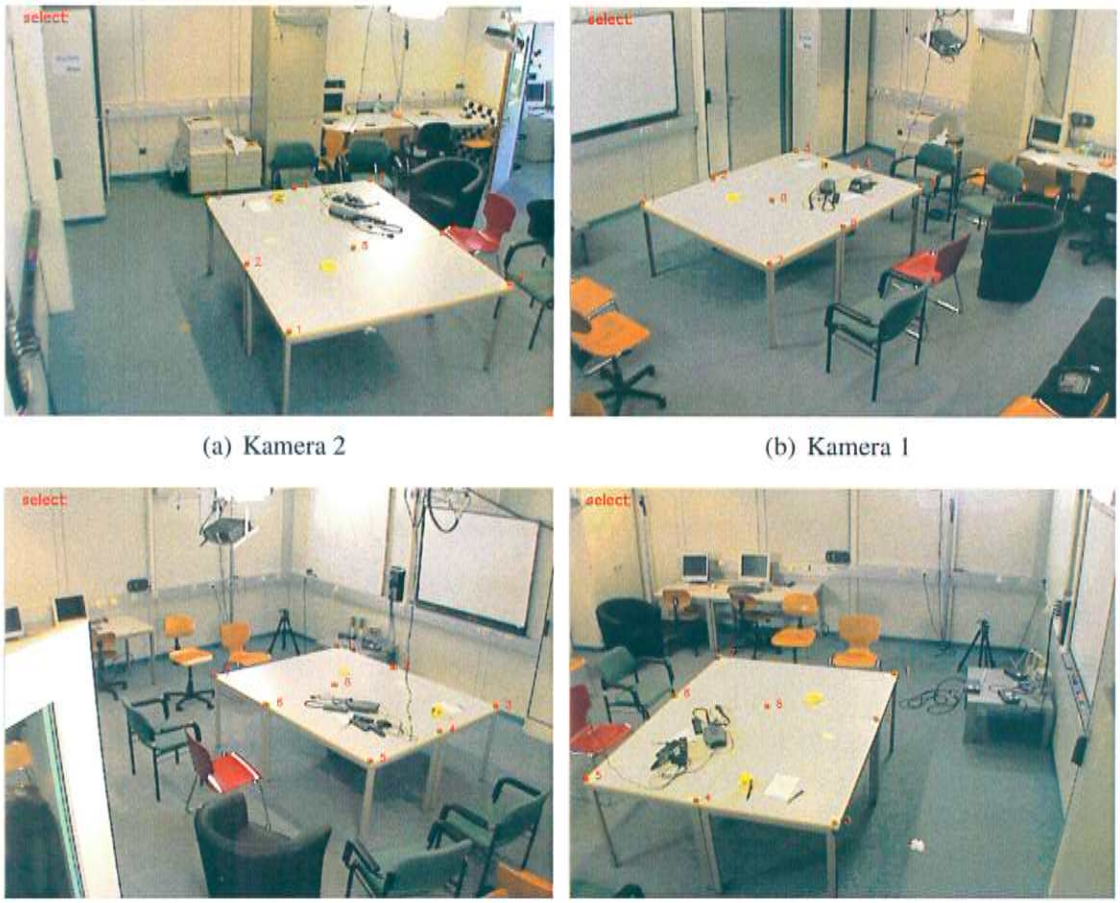
Um die Kalibrierung zu testen, wurden drei unterschiedliche Aufnahmen im Smartroom aufgezeichnet, jeweils mit einer, zwei und drei Personen. Dabei hatten die Personen die Vorgabe, sich im Raum zu bewegen, und sich möglichst im überlappenden Sichtbereich der Kameras aufzuhalten, damit genügend Daten für die Kalibrierung zur Verfügung stehen. Da das Verfahren auf Bewegung achtet, werden Personen, die stehen bleiben, ignoriert und liefern keine Daten für die Kalibrierung. Aus diesem Grund wurden alle Personen angewiesen, nicht zu lange stehen zu bleiben. Alle Aufnahmen wurden mit 15 Bildern pro Sekunde aufgezeichnet und haben die folgenden Längen:

- Aufnahme mit einer Person: 2604 Frames (ca. 2 Minuten 54 Sekunden)
- Aufnahme mit zwei Personen: 2829 Frames (ca. 3 Minuten 9 Sekunden)
- Aufnahme mit drei Personen: 4051 Frames (ca. 4 Minuten 30 Sekunden)

5.1 Experiment 1: Eine Person

Zuerst wird die Aufnahme mit einer Person betrachtet. Abbildung 5.5 zeigt synchrone Beispielbilder dieser Aufnahme. Hierbei betritt eine Person den Raum und bleibt in Bewegung, um ständig Daten zu generieren. Dabei wurde versucht, den Bildbereich aller Kameras vollständig auszunutzen.

Abbildung 5.6 zeigt die Ergebnisse der Suche nach den besten externen Parametern θ, φ, α, β und γ. Es wird die absolute Differenz zur Grundwahrheit angezeigt. Hierbei fällt



(a) Kamera 2 (b) Kamera 1

(c) Kamera 3 (d) Kamera 4

Abbildung 5.4: Der Projektionsfehler der Referenzkalibrierung. Die korrekten Bildpositionen sind grün, die Projektionen der Positionen im dreidimensionalen Raum sind rot markiert. Da die Fehler jedoch minimal sind, verdecken die roten Markierungen die grünen.

auf, dass die Kalibrierungen von Kamera 2 und Kamera 3 zu Kamera 4 deutlich schlechter ausfallen, als der Durchschnitt. Auch die Kalibrierungen von Kamera 4 und Kamera 2 zu Kamera 3 sind schlechter als die restlichen. Dies liegt vermutlich daran, dass der komplette rechte Bildbereich von Kamera 4 nicht ausgenutzt wurde, da in diesem Bereich mehrere Gegenstände standen (s. Abb. 5.5 (d)). Weiterhin scheinen bei dieser Aufnahme sehr wenig Daten für die Kalibrierung vorhanden gewesen zu sein. Bei den Aufnahmen mit einer Person wurden in jedem Bild nur ein Merkmal (Kopf der Person) gefunden. In den folgenden Experimenten zeigt sich, dass sich die Kalibrierung mit der Anzahl der verwerteten Merkmale verbessert.

Das anhand dieser Kalibrierungen erzeugte Kameranetzwerk wird in Abbildung 5.7 gezeigt. Wie sich aus den Kalibrierungsdaten vermuten lässt, ist auch das Kameranetzwerk fehlerbehaftet. Jedoch ist nur Kamera 3 falsch positioniert, Kamera 4 konnte trotz der schlechten Kalibrierungen korrekt positioniert werden.

Die Güte des Kameranetzwerks zeigt sich bei Betrachtung des Triangulations- und Projektionsfehlers. Hierfür wurde anhand der Kalibrierung und der aufgezeichneten Bilder die Position der Marker berechnet. Zusätzlich wurden die bekannten Positionen im dreidimensionalen Raum der Merker die Punkte in die Bilder der Kameras projiziert und der Fehler gemessen. Abbildung 5.8 zeigt den Projektionsfehler. Die beobachteten Fehler zeigt Tabelle 5.2.

	Mittlerer Fehler	Standardabweichung
Triangulation	176,71 mm	13,81 mm
Projektion Kamera 1	10,12 Pixel	3,31 Pixel
Projektion Kamera 2	1,84 Pixel	0,58 Pixel
Projektion Kamera 3	81,66 Pixel	18,10 Pixel
Projektion Kamera 4	5,08 Pixel	1,66 Pixel

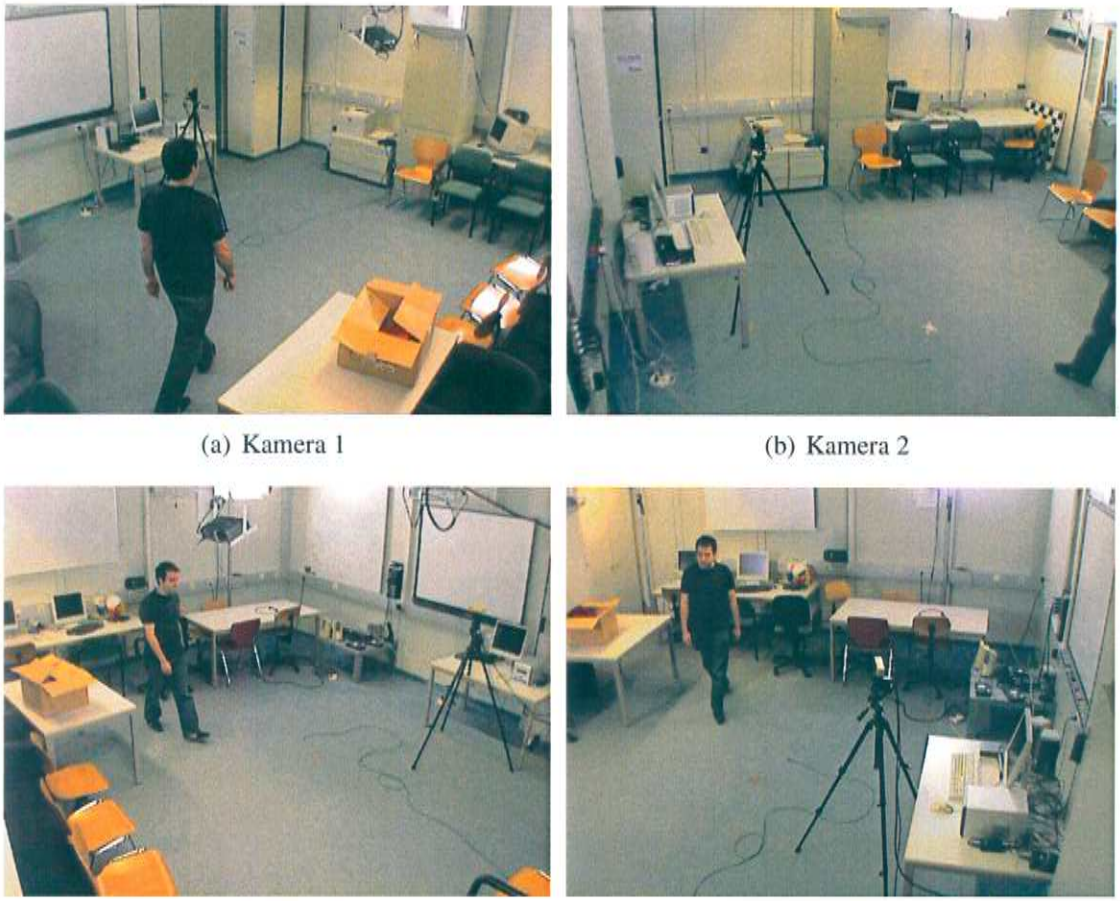
Tabelle 5.2: Der Triangulations- und Projektionsfehler der berechneten Kalibrierung aus Experiment 1.

Der große Triangulationsfehler ist hauptsächlich auf die falsche Positionierung von Kamera 3 zurückzuführen. Bei einer Triangulation der Markerpositionen ohne Kamera 3 ergibt sich ein mittlerer Triangulationsfehler von 32,81 mm mit einer Standardabweichung von 7,56 mm.

Eine paarweise Kalibrierung von zwei Kameras dauerte im Schnitt ca. 23 Minuten. Die gesamte Kalibrierung wurde auf alle vier Kamerarechner verteilt und dauerte insgesamt ca. 70 Minuten. Durchschnittlich wurden pro paarweise Kalibrierung 221 Korrespondenzen verwendet. Die kleinste Anzahl an Korrespondenzen hatten dabei Kamera 2 $\Leftrightarrow$ Kamera 3 (131 Korrespondenzen) und Kamera 3 $\Leftrightarrow$ Kamera 4 (201 Korrespondenzen).

5.2 Experiment 2: Zwei Personen

Die Aufnahmen von Experiment 2 entsprechen größtenteils dem Szenario von Experiment 1, jedoch befinden sich nun zwei Personen im Raum. Hiermit soll die Fähigkeit des Kalibrierungsalgorithmus, mehrere Korrespondenzen richtig aufzulösen, getestet werden.



(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 5.5: Beispielbilder der Aufnahmen für Experiment 1.

	θ	φ	γ	α	β
Kamera 1 -> Kamera 2	0,45680	0,48030	0,13990	1,41140	1,27620
Kamera 1 -> Kamera 3	1,27160	0,54760	1,32440	5,73140	1,60360
Kamera 1 -> Kamera 4	0,01740	0,11580	2,02810	0,75200	0,04180
Kamera 2 -> Kamera 1	0,03380	0,10028	0,29160	1,96630	0,21450
Kamera 2 -> Kamera 3	1,69200	0,32870	10,09840	4,48600	0,56480
Kamera 2 -> Kamera 4	7,37490	3,03810	5,70400	14,10500	2,54800
Kamera 3 -> Kamera 1	1,62129	0,02370	0,65170	3,14660	1,13780
Kamera 3 -> Kamera 2	0,59290	0,22560	3,08327	2,01500	0,31290
Kamera 3 -> Kamera 4	12,09580	1,23790	0,13920	4,28830	1,67260
Kamera 4 -> Kamera 1	0,28480	0,34670	0,28040	0,41400	0,85200
Kamera 4 -> Kamera 2	0,17240	0,24310	1,21880	0,65100	0,53070
Kamera 4 -> Kamera 3	14,94910	2,20867	3,55770	18,28590	3,23720

Abbildung 5.6: Der Absolute Fehler der Kalibrierung aus Experiment 1 zur Grundwahrheit in Grad.

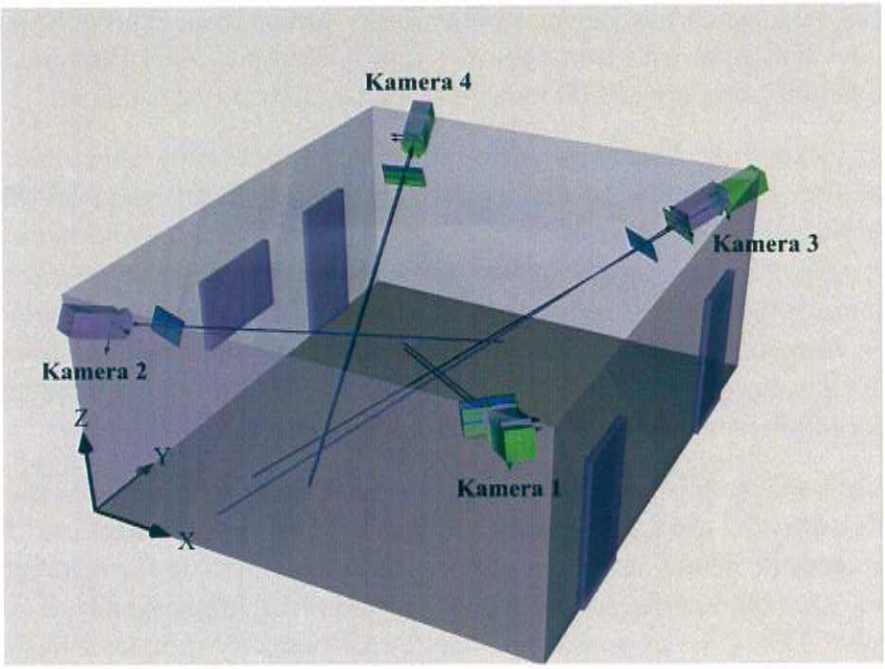
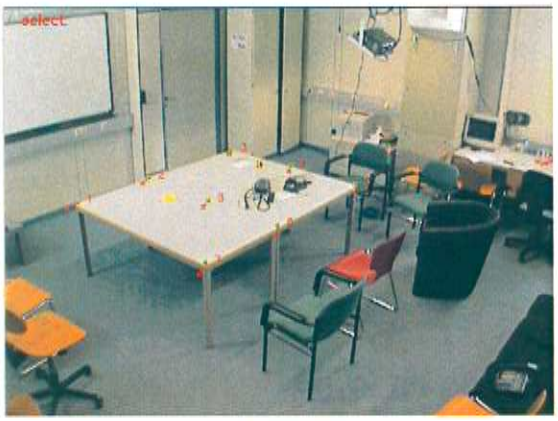


Abbildung 5.7: Die schematische Darstellung der Kalibrierungsergebnisse aus Experiment 1 (blau). Zum Vergleich wurde grün die Grundwahrheit eingeblendet.



(a) Kamera 1

(b) Kamera 2

(c) Kamera 3

(d) Kamera 4

Abbildung 5.8: Der Projektionsfehler aus der Kalibrierung von Experiment 1. Die korrekten Bildpositionen sind grün, die Projektionen der Positionen im dreidimensionalen Raum sind rot markiert.

Abbildung 5.9 zeigt synchrone Beispielsbilder dieser Aufnahmen. Hierbei ist auch gut zu sehen, dass der Fall auftreten kann, dass in einem Kamerabild zwei Personen, in einem anderen Kamerabild aber nur eine Person zu sehen ist (Abb. 5.9 (c) und (d)).

Abbildung 5.10 zeigt die Ergebnisse der paarweisen Kalibrierung. Auch hier wird die absolute Differenz zur Grundwahrheit angezeigt. Es zeigt sich, dass der Algorithmus keine Probleme hat, die paarweisen Kalibrierungen zu berechnen, obwohl keine eindeutigen Punktkorrespondenzen vorhanden sind. Generell hat sich die paarweise Kalibrierung im Vergleich zu Experiment 1 verbessert. Es ist nur ein großer Ausreißer bei der Kalibrierung von Kamera 2 zu Kamera 3 zu beobachten. Die generelle Verbesserung wird hauptsächlich auf die größere Anzahl an Daten für die Kalibrierung zurückgeführt. Die Merkmalsextraktion kann nun größtenteils zwei Merkmale pro Bild liefern.

Das hieraus aufgebaute Kameranetzwerk wird in Abbildung 5.11 schematisch dargestellt. Es zeigt sich, dass trotz der falsch ermittelten Kalibrierung von Kamera 2 auf Kamera 3, das Kameranetzwerk richtig aufgebaut wurde. Dies ist möglich da für den Aufbau eines Kameranetzwerkes mit vier Kameras nur fünf paarweise Kalibrierungen benötigt werden, tatsächlich aber $12 \times 2 = 24$ vorliegen. Dadurch kann das System aus der Menge aller Möglichkeiten das Kameranetzwerk aufzubauen, die beste auswählen, und somit einzelne Ausreißer unterdrücken. Experiment 1 zeigte jedoch, dass dies nur in bestimmten Maße möglich ist. Dort konnten nicht alle Ausreißer unterdrückt werden.

Auch hier wird das Kameranetzwerk anhand des Triangulations- und Projektionsfehlers (Abb. 5.12) bewertet. Die gemessenen Fehler zeigt Tabelle 5.3.

	Mittlerer Fehler	Standardabweichung
Triangulation	44,47 mm	3,16 mm
Projektion Kamera 1	1,22 Pixel	0,60 Pixel
Projektion Kamera 2	7,78 Pixel	0,59 Pixel
Projektion Kamera 3	9,71 Pixel	2,37 Pixel
Projektion Kamera 4	6,32 Pixel	1,10 Pixel

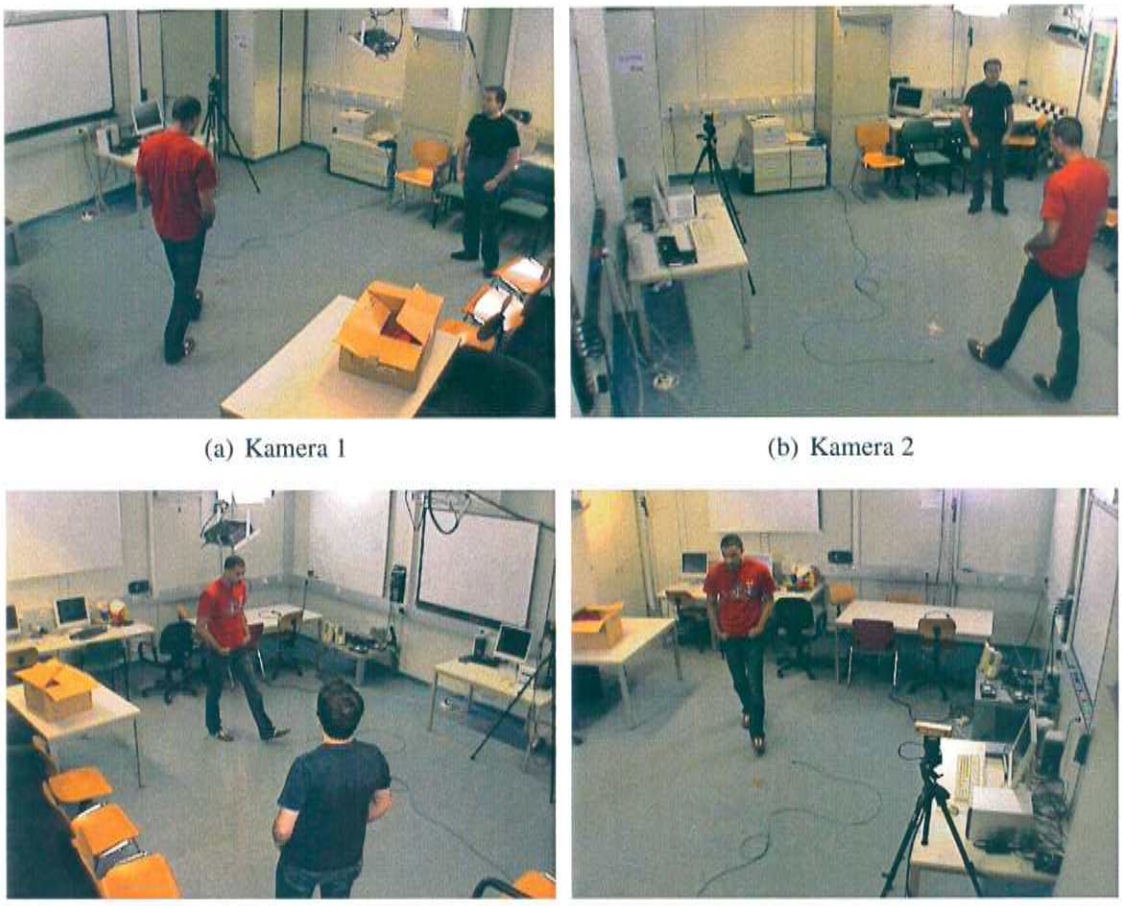
Tabelle 5.3: Der Triangulations- und Projektionsfehler der berechneten Kalibrierung aus Experiment 2.

Erwartungsgemäß sind hier sowohl der Triangulations- als auch die Projektionsfehler klein.

In diesem Experiment dauerte eine paarweise Kalibrierung ca. 45 Minuten. Die gesamte Kalibrierung auf vier Rechner verteilt nahm ca. 2 Stunden und 15 Minuten in Anspruch. Für eine paarweise Kalibrierung wurden durchschnittlich 676 hypothetische Korrespondenzen verwendet.

5.3 Experiment 3: Drei Personen

Die Aufnahmen zu Experiment drei unterscheiden sich stärker zu den vorherigen Aufnahmen. In diesem Szenario befinden sich drei Personen im Raum. Weiterhin ist es ihnen freigestellt, wie sich bewegen. Die Personen dürfen nach eigener Entscheidung stehen



(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 5.9: Beispielbilder der Aufnahmen für Experiment 2.

	θ	φ	γ	α	β
Kamera 1 -> Kamera 2	0,88650	0,16780	0,29620	1,08860	0,27620
Kamera 1 -> Kamera 3	0,91590	0,54760	0,43340	1,23140	0,60360
Kamera 1 -> Kamera 4	0,56427	0,86080	2,02810	2,25200	1,45820
Kamera 2 -> Kamera 1	0,20060	0,05598	0,13540	0,03370	0,28550
Kamera 2 -> Kamera 3	23,60610	13,37560	6,23120	33,98600	12,56480
Kamera 2 -> Kamera 4	3,07800	1,47560	3,12590	7,10500	2,04800
Kamera 3 -> Kamera 1	1,42599	0,02370	0,96420	3,64660	1,13780
Kamera 3 -> Kamera 2	3,15710	2,27440	3,32298	6,48500	3,18710
Kamera 3 -> Kamera 4	5,29890	0,30040	0,22540	4,28830	0,17260
Kamera 4 -> Kamera 1	0,30040	0,36230	0,10850	0,41400	0,85200
Kamera 4 -> Kamera 2	0,64110	0,24310	0,57810	1,34900	0,53070
Kamera 4 -> Kamera 3	1,27720	0,56805	0,51080	3,28590	1,23720

Abbildung 5.10: Der Absolute Fehler der Kalibrierung aus Experiment 2 zur Grundwahrheit in Grad.

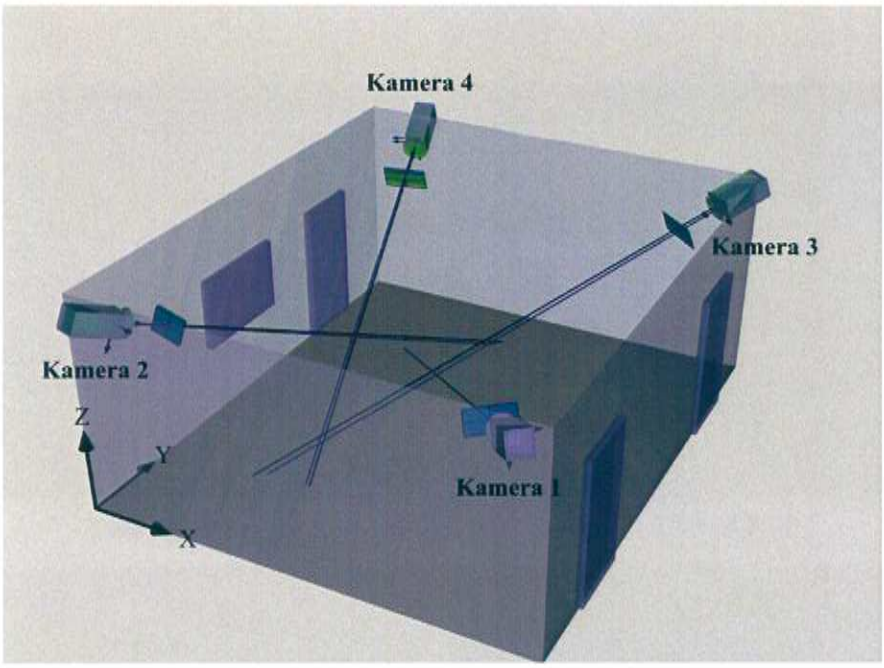
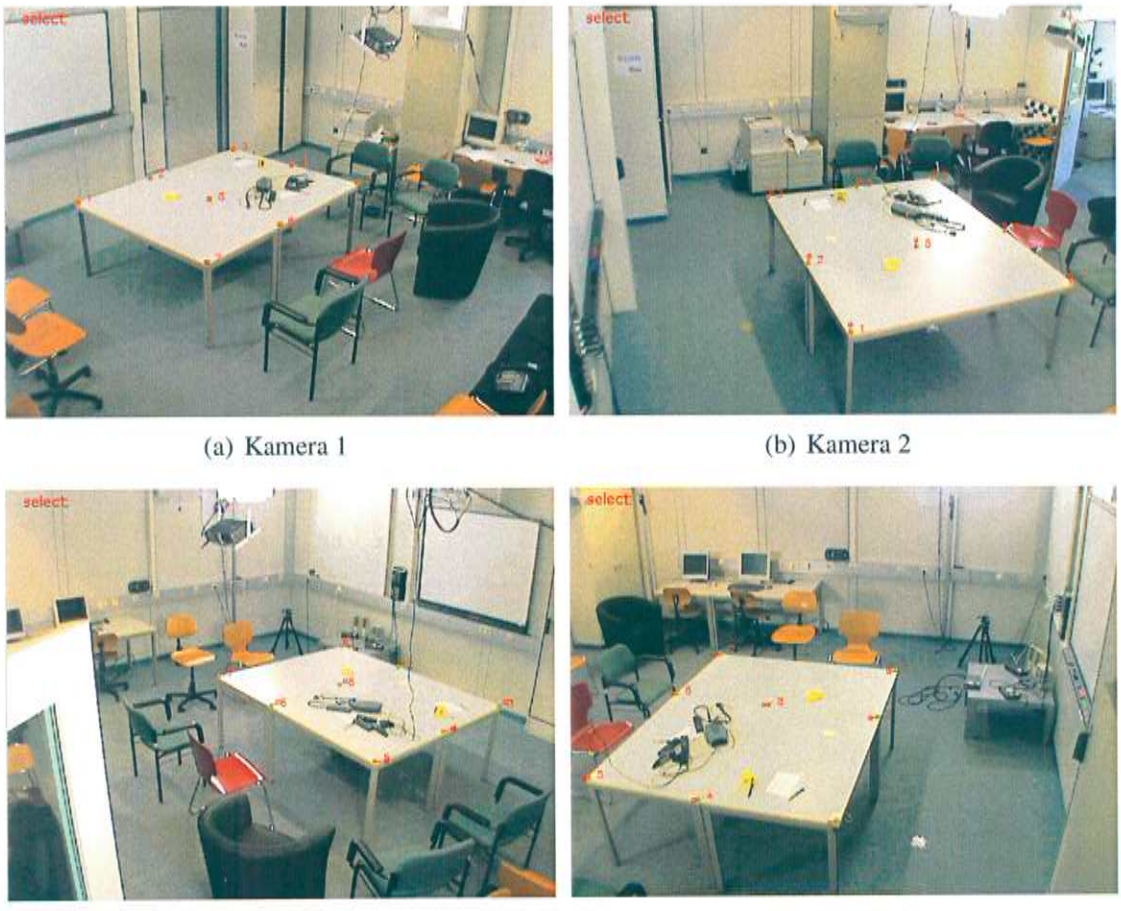


Abbildung 5.11: Die schematische Darstellung der Kalibrierungsergebnisse aus Experiment 2 (blau). Zum Vergleich wurde grün die Grundwahrheit eingeblendet.



(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 5.12: Der Projektionsfehler aus der Kalibrierung von Experiment 2. Die korrekten Bildpositionen sind grün, die Projektionen der Positionen im dreidimensionalen Raum sind rot markiert.

bleiben, sich setzen oder miteinander reden. Synchrone Beispielbilder werden in Abbildung 5.13 gezeigt. Mit diesen Aufnahmen soll die generelle Nutzbarkeit des Systems in allgemeinen Szenarien getestet werden.

Die absolute Differenz zwischen der Grundwahrheit und den Ergebnissen der paarweisen Kalibrierungen werden in Abbildung 5.14 gezeigt. Hierbei ist zu beobachten, dass es keine großen Ausreißer gibt. Das System konnte alle paarweisen Kalibrierungen mit relativ kleinem Fehler bestimmen. Auch hier zeigt sich, dass sich das System mit steigender Anzahl an Kalibrierungsdaten verbessert. Die Merkmalsextraktion konnte hier im Schnitt zwei bis drei Merkmale pro Bild erzeugen.

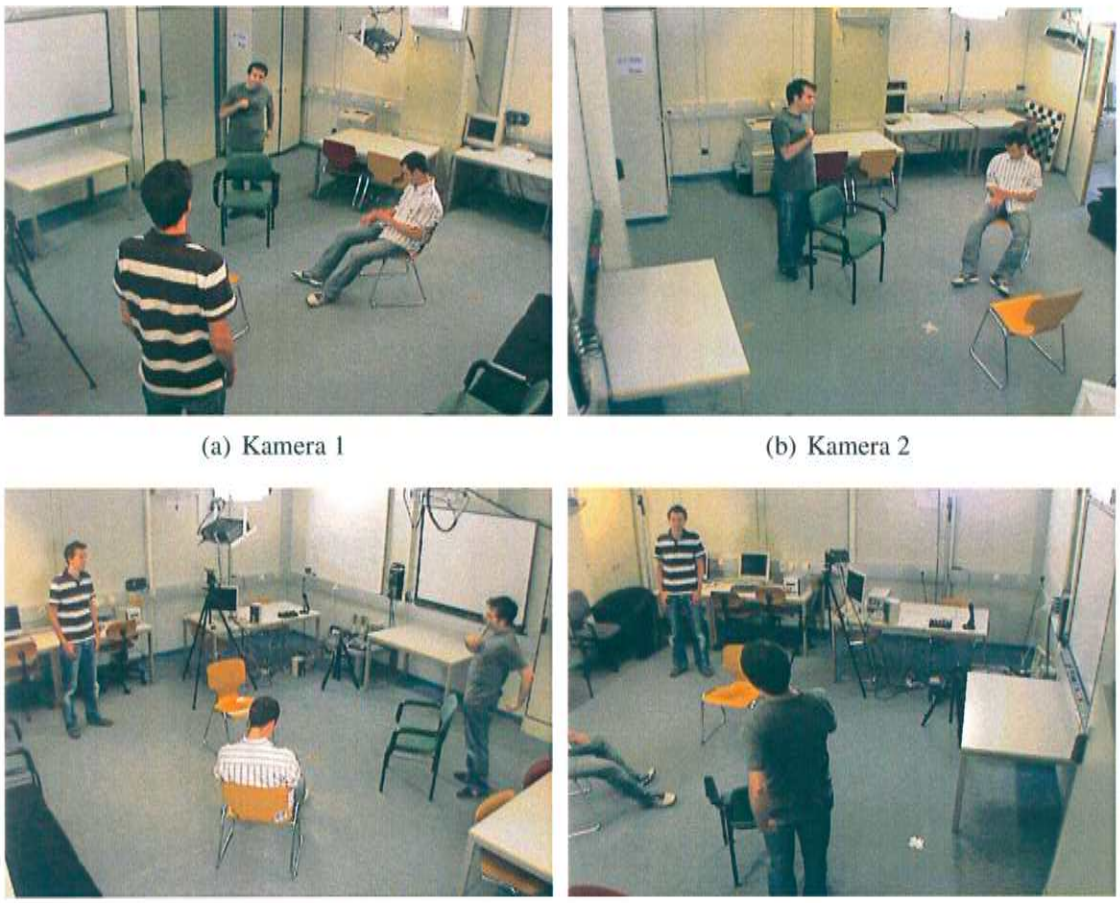
Entsprechend diesen Daten sind auch keine großen Fehler im aufgebauten Kameranetzwerk zu erwarten. Abbildung 5.15 zeigt das resultierende Kameranetzwerk.

Die gemessenen Triangulations- und Projektionsfehler werden in Tabelle 5.4 aufgeführt und sind in Abbildung 5.16 dargestellt.

	Mittlerer Fehler	Standardabweichung
Triangulation	49,98 mm	16,37 mm
Projektion Kamera 1	11,51 Pixel	1,32 Pixel
Projektion Kamera 2	6,06 Pixel	0,57 Pixel
Projektion Kamera 3	2,70 Pixel	0,55 Pixel
Projektion Kamera 4	5,85 Pixel	4,52 Pixel

Tabelle 5.4: Der Triangulations- und Projektionsfehler der berechneten Kalibrierung aus Experiment 3.

Eine paarweise Kalibrierung benötigte für dieses Experiment ca. 70 Minuten. Auf vier Rechner verteilt, ergibt sich daraus eine Gesamtlaufzeit von ca. 3 Stunden und 30 Minuten. Es wurden dabei für die paarweisen Kalibrierungen etwa 1122 hypothetische Korrespondenzen verwendet. Durchschnittlich handelt es sich dabei um ca. $\frac{1}{3}$ korrekte und $\frac{2}{3}$ falsche Korrespondenzen.



(a) Kamera 1 (b) Kamera 2

(c) Kamera 3 (d) Kamera 4

Abbildung 5.13: Beispielbilder der Aufnahmen für Experiment 3.

	θ	φ	γ	α	β
Kamera 1 -> Kamera 2	0,02710	0,24590	0,13990	0,41140	0,77620
Kamera 1 -> Kamera 3	0,56850	0,23510	0,08190	0,76860	0,60360
Kamera 1 -> Kamera 4	0,48615	0,54830	1,63750	1,75200	0,95820
Kamera 2 -> Kamera 1	1,04940	0,75910	0,29160	2,53370	1,28550
Kamera 2 -> Kamera 3	0,16860	0,56310	1,81720	1,48600	1,06480
Kamera 2 -> Kamera 4	1,74990	0,53810	2,57900	4,60500	1,04800
Kamera 3 -> Kamera 1	0,37129	0,02370	0,02670	0,64660	0,63780
Kamera 3 -> Kamera 2	0,26648	0,24320	0,23704	0,48500	0,68710
Kamera 3 -> Kamera 4	2,30530	0,04000	0,71210	4,71170	1,17260
Kamera 4 -> Kamera 1	0,90980	0,97170	1,68660	0,41400	1,35200
Kamera 4 -> Kamera 2	0,80420	0,59470	1,25780	2,15100	0,53070
Kamera 4 -> Kamera 3	1,66790	1,34930	0,43270	4,78590	2,23720

Abbildung 5.14: Der Absolute Fehler der Kalibrierung aus Experiment 3 zur Grundwahrheit in Grad.

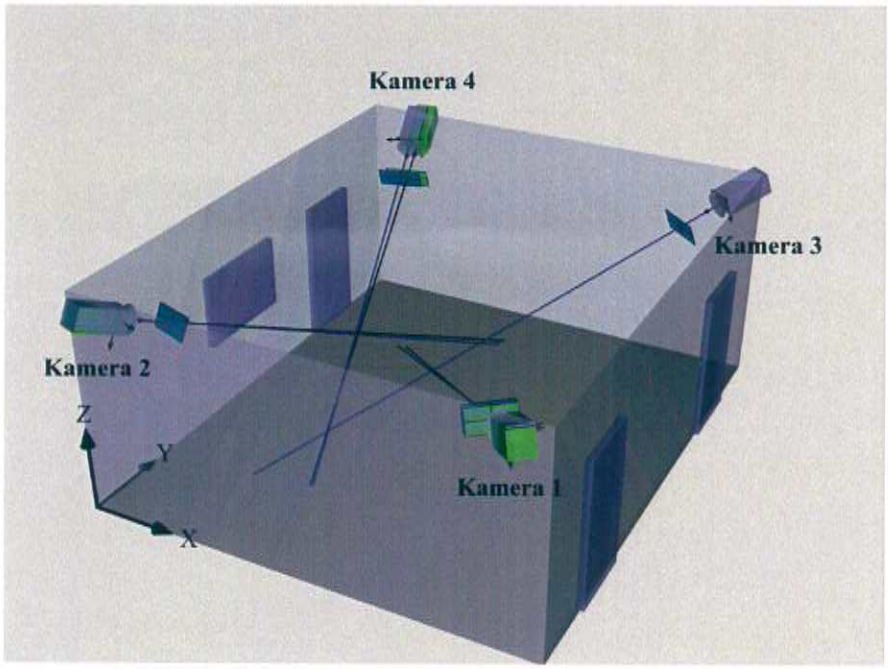
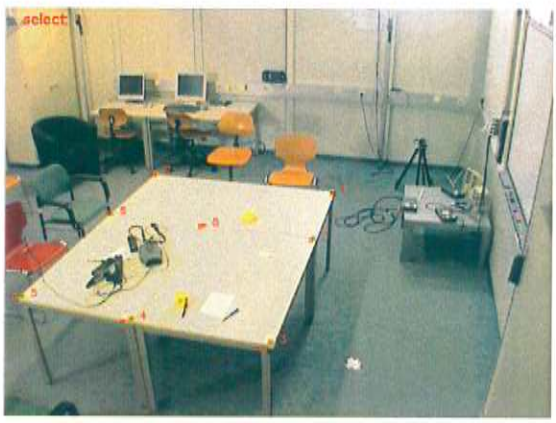


Abbildung 5.15: Die schematische Darstellung der Kalibrierungsergebnisse aus Experiment 3 (blau). Zum Vergleich wurde grün die Grundwahrheit eingeblendet.

(a) Kamera 1

(b) Kamera 2

(c) Kamera 3



(d) Kamera 4

Abbildung 5.16: Der Projektionsfehler aus der Kalibrierung von Experiment 3. Die korrekten Bildpositionen sind grün, die Projektionen der Positionen im dreidimensionalen Raum sind rot markiert.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein automatisches Kalibrierungssystem entwickelt, dass durch das Beobachten der Bewegung in einem Raum die extrinsische Kalibrierung eines Kameranetzwerkes berechnet. Hierfür wurde zuerst ein Filter ausgearbeitet, der anhand der aufgezeichneten Bilder Merkmale extrahiert, die für die Kalibrierung benutzt werden können. Um die Ergebnisse des Filters zu verbessern, wurde Kontextwissen über das zu Grunde liegende Szenario verwendet. Das Szenario entspricht dabei einem intelligenten Raum, der mit mehreren Videokameras ausgestattet ist. Innerhalb des Raums befinden und bewegen sich mehrere Menschen, die durch den intelligenten Raum beobachtet werden sollen. Dabei wurde ausgenutzt, dass alle Kameras ungefähr aufrecht ausgerichtet sind. Anhand dieses Wissens kann sehr einfach der höchste Punkt eines Objektes bestimmt werden. Außerdem ist der höchste Punkt eines Objekts am unwahrscheinlichsten in anderen Kamerasichten verdeckt. Weiterhin wurden nur bewegte Bildbereiche betrachtet, da das Detektieren von Punktkorrespondenzen mit Farbinformation ein relativ komplexes Problem darstellt, dass sich bei Benutzung von Bewegungsmerkmalen vereinfacht. Anhand von mehreren Kriterien werden dabei aus einer Menge von Merkmalshypothesen die gewünschten Merkmale ausgewählt.

Der Algorithmus zur anschließenden Kalibrierung ist ein automatischer Selbstkalibrierungsansatz, d.h. es wurde nur mit Punktkorrespondenzen aus den einzelnen Aufnahmen gearbeitet und keine zusätzlichen Positionen im dreidimensionalen Raum der sich bewegenden Objekte in den Aufnahmen benötigt. Im Unterschied zu den bekannten Selbstkalibrierungsverfahren werden keine korrekten Punktkorrespondenzen gebraucht. Das System betrachtet für die Kalibrierung die gesamte Menge an möglichen Korrespondenzen zwischen zwei Kamerabildern. Ausgehend vom Ansatz der Minimierung des Triangulationsfehlers wurde ein Hybridverfahren entwickelt. Hierfür wurde der zuerst sehr große Suchraum anhand einer Parameterumformung und einer Houghtransformation verkleinert. Die implementierte Houghtransformation ist in der Lage anhand von drei Parametern für die Kameraposition und Ausrichtung θ, φ, γ die hypothetischen Korrespondenzen zu verifizieren und bietet eine gute Bewertungsfunktion für eine bestimmte Parameterkombination. Mit einem modifizierten Gradientenabstieg (Hill-Climbing) werden dann die besten externen Parameter gesucht, die der realen extrinsischen Kalibrierung entsprechen. Die Kalibrierung wird dabei für alle möglichen Kamerapaare berechnet. Nach Hinzunahme der Inversen der berechneten paarweisen Kalibrierungen stehen für den nachfolgenden Kameranetzwerkaufbau genügend Informationen zur Verfügung. Der Hintergrund dieser zusätzlichen Redundanz ist, dass einzelne Fehler in der paarweisen Kalibrierung so überbrückt werden können. Für die Bestimmung des endgültigen Kameranetzwerkes wurde die Menge aller möglichen Kameranetzwerke betrachtet und das beste automatisch ausgewählt. Für die Bewertung der Güte eines Netzwerks wurde eine Bewertungsfunktion entworfen, die den Fehler zwischen der tatsächlichen Position einer Kamera und der

theoretischen Position entsprechend der paarweisen Kalibrierungen, berechnet. Da das berechnete Kameranetzwerk relativ zu einer Kamera ausgerichtet und nicht maßstabsgetreu aufgebaut wurde, wurde gezeigt, wie das Kameranetzwerk unter Zunahme von zusätzlichen Informationen in das Weltkoordinatensystem transformiert werden kann.

Anschließend wurde das entwickelte System anhand von drei Experimenten validiert. Hierfür wurden Szenarien mit steigendem Schwierigkeitsgrad (eine, zwei und drei Personen) aufgezeichnet. Anhand einer im Voraus manuell ermittelten und verifizierten Grundwahrheit wurden die Ergebnisse verglichen. Dabei zeigte das System, dass es mit Aufnahmen, in denen sich mehrere Personen bewegen, für die Korrespondenzen in einzelnen Bildern aber nicht bekannt sind, funktioniert. Deutlich wurde dabei, dass das System mit der Anzahl der zur Verfügung stehenden Merkmale seine Ergebnisse verbessert. Weiterhin wurde gezeigt, dass einzelne Ausreißer in der paarweisen Kalibrierung beim Aufbau des Kameranetzwerkes ignoriert werden. Anhand der Experimente konnte gezeigt werden, dass das System korrekte Ergebnisse unter realen Bedingungen liefert und somit der Aufwand bei der extrinsischen Kalibrierung deutlich verringert werden kann.

Das vorgestellte Verfahren lässt sich noch in mehrerer Hinsicht erweitern bzw. verbessern.

Der größte Nachteil des Systems liegt in dessen Geschwindigkeit. Eine Kalibrierung eines Kameranetzwerkes mit vier Kameras und Aufnahmen von drei Personen benötigt durchschnittlich ca. drei Stunden und 30 Minuten Rechenzeit. Jedoch kann der Algorithmus an einigen Stellen optimiert und somit beschleunigt werden:

Multi-Threading
Im Zeitalter der Dual-Core und Quad-Core Prozessoren kann der Algorithmus parallel auf allen Prozessoren eines Mehrprozessorsystems ausgeführt werden. Weiterhin kann sogar die Suche parallelisiert werden, da die Berechnungen der einzelnen Houghtransformationen für die Bewertungsfunktion komplett voneinander unabhängig sind. So lässt sich der Aufwand bei der Betrachtung der Nachbarknoten und deren Bewertung auf die einzelnen Prozessoren eines Systems verteilen.

Auswahl der Merkmalspunkte
Da der Aufwand des Algorithmus mit der Anzahl der zu verwendenden Daten steigt, lässt sich durch eine Reduzierung der Daten die Laufzeit des Algorithmus verbessern. Jedoch verbessern sich auch die Ergebnisse mit der Anzahl der Daten. Aus diesem Grund müssen die einzelnen Korrespondenzen genauer betrachtet werden, um nur solche zuzulassen, die bezüglich der Kalibrierung einen hohen Informationsteil besitzen. Beispielsweise enthalten kleine Änderungen zwischen zwei folgenden Bildern weniger Informationen als solche mit einem größeren Unterschied. Beispielsweise kann dafür einfach die Distanz zwischen den einzelnen Merkmalen betrachtet werden.

Gestufte Houghtransformation
Der größte Zeitaufwand in diesem Algorithmus wird durch die Berechnung der Houghtransformationen verbraucht. Aus diesem Grund kann das gesamte System durch eine Optimierung der Houghtransformation beschleunigt werden. Bei der Houghtransformationen wird die Verbindungsgerade aus Gleichung (4.12) mit der feinsten Auflösung berechnet (Abb. 4.9). Jedoch wird nur das Maximum für die Bewertungsfunktion benutzt. Weiterhin kann anhand einer groben Rasterung der ungefähre Bereich, in dem das Maximum liegt, abgeschätzt werden. Dadurch lässt sich erst mit einer groben Rasterung die ungefähre Lage des Maximums ermitteln, um dann folgend in einer feineren Rasterung diesen Bereich genauer zu betrachten.

Weiterhin kann der Algorithmus bezüglich der Ergebnisse optimiert werden:

Suchraum

Um den Aufwand zu verkleinern musste, der Suchraum in diesem Algorithmus grob in einen 90° Bereich eingegrenzt werden. Diese Einschränkung kann jedoch weiter gelockert werden, wenn beispielsweise zuvor der komplette Suchraum mit einer gröberen Rasterung (bspw. 20° oder größer) durchsucht wird. Dadurch könnte grob abgeschätzt werden in welchem 90° Bereich die Lösung liegt. Sobald dieser feststeht, kann der bisherige Algorithmus die Kalibrierung berechnen.

Suchalgorithmus

Unter Umständen kann es vorkommen, dass die Suche die korrekte Lösung verfehlt und dadurch nur an einem lokalen Maximum endet. Durch die vorgegebenen Schrittweiten kann der Algorithmus am Anfang die korrekte Lösung überspringen. Eine feinere initiale Rasterung würde dieses Problem zwar lösen aber die Laufzeit stark verschlechtern.

Globale Optimierung des bestimmten Kameranetzwerkes

Das Ergebnis des Algorithmus ist im Idealfall ein Kameranetzwerk, dass ungefähr dem realen Kameraaufbau entspricht. Ausgehend von dieser Startposition könnte mit einer globalen Optimierung das Ergebnis weiter verbessert werden. Hierfür könnten beispielsweise die aufgezeichneten Punktkorrespondenzen wiederverwendet werden.

Automatische Skalierung und Transformation ins Weltsystem

In dieser Arbeit wurde das bestimmte Kameranetzwerk mit Hilfe der Grundwahrheit bzgl. einer Kameraposition und der Distanz zwischen zwei konkreten Kameras in das Weltkoordinatensystem transformiert. Jedoch lässt sich dieser Schritt durch Zunahme eines kleinen Kalibrierungsobjekts automatisieren. Hierfür wäre die Größe und die Position des Objektes notwendig. Anhand von zwei Kamerabildern könnten diese Daten im relativen Koordinatensystem berechnet werden. Zusammen mit der Größe und Position im Weltkoordinatensystem ist eine Transformation ins Weltkoordinatensystem möglich.

Literaturverzeichnis

[1] Addison-Wesley. *Artificial Intelligence*. Patrick Henry Winston, 1984. 34

[2] A. Azarbayejani and A. P. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *International Conference on Pattern Recognition*, pages III: 627–632, 1996. 6

[3] A. Bartoli and P. Sturm. Nonlinear estimation of the fundamental matrix with minimal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):426–432, 2004. 12

[4] K. Bernardin, H. K. Ekenel, and R. Stiefelhagen. Multimodal identity tracking in a smartroom. In I. Maglogiannis, K. Karpouzis, and M. Bramer, editors, *Artificial Intelligence Applications and Innovations*, volume 204 of *International Federation for Information Processing*, pages 324–336. Springer, 2006. 1

[5] J.-Y. Bouguet. Camera calibration toolbox for matlab `http://www.vision.caltech.edu/bouguetj/calib_doc/`, April 25th 2007. 5, 44

[6] X. Chen, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 520–527, Los Alamitos, June 13–15 2000. IEEE. 6

[7] P. Chippendale and F. Tobia. Collective calibration of active camera groups. In *Advanced Video and Signal Based Surveillance*, pages 456–461, 2005. 7, 8

[8] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–334, 1992. 6

[9] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision*, pages 579–587, 1992. 12

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000. 11, 12

[11] O. Lanz. Automatic lens distortion estimation for an active camera. In *International Conference on Computer Vision and Graphics*, 2004. 7

[12] A. P. Pentland. Smart rooms, smart clothes. In *International Conference on Pattern Recognition*, pages Vol II: 949–953, 1998. 1

[13] B. G. S. Ramesh Jain, Rangachar Kasturi. *Machine Vision*. McGraw-Hill, Inc., 1995. 15

[14] S. N. Sinha and M. Pollefeys. Synchronization and calibration of camera networks from silhouettes. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004.*, pages 116–119, 2004. 7

[15] R. Stiefelhagen, K. Bernardin, H. K. Ekenel, J. McDonough, K. Nickel, M. Voit, and M. Wölfel. Audio-visual perception of a lecturer in a smart seminar room. *Signal Processing*, 86(12):3518–3533, 2006. 1

[16] F. Szenberg, P. C. P. Carvalho, and M. Gattass. Automatic camera calibration for image sequences of a football match. In S. Singh, N. A. Murshed, and W. G. Kropatsch, editors, *Second International Conference of Advances in Pattern Recognition 2001*, volume 2013 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 2001. 7, 8

[17] Z. Szlávik, L. Havasi, and T. Szirányi. Image matching based on co-motion statistics. In *International Symposium on 3D Data Processing, Visualization and Transmission 2004*, pages 584–591. IEEE Computer Society, 2004. 7

[18] Z. Szlávik, T. Szirányi, L. Havasi, and C. Benedek. Optimizing of searching co-motion point-pairs for statistical camera calibration. In *IEEE International Conference on Image Processing, 2005*, pages 1178–1181, 2005. 7

[19] Z. Szlávik, L. Havasi, T. Szirányi, and C. Benedek. Random motion for camera calibration. In *13th European signal processing conference. Antalya, 2005.*, 2005. 7, 8

[20] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Trans. Robotics and Automation*, 3(4):323–344, 1987. 5, 9

[21] M. Voit, K. Nickel, and R. Stiefelhagen. Multi-view head pose estimation using neural networks. In *Computer and Robot Vision*, pages 347–352, 2005. 1

[22] W. Wang and H. T. Tsui. An svd decomposition of essential matrix with eight solutions for the relative positions of two perspective cameras. *15th International Conference on Pattern Recognition 2000*, 01:1362, 2000. 12

[23] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 9, 23