UNIVERSITÄT KOBLENZ · LANDAU

Fachbereich 4 Informatik

Untersuchung von Analyse-durch-Synthese Techniken im markerlosen Tracking

von

Dipl.-Inf. Martin Schumann

Genehmigte Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
Fachbereich 4: Informatik
Universität Koblenz-Landau

Vorsitzender des Promotionsausschusses: Prof. Dr. Jan Jürjens
Vorsitzender der Promotionskommission: Prof. Dr. Klaus Diller
Berichterstatter und Betreuer: Prof. Dr.-Ing. Stefan Müller
Weitere Berichterstatterin: Prof. Gudrun Klinker, Ph.D.

Datum der wissenschaftlichen Aussprache: 06.05.2020

Zusammenfassung

Im Kontext der Erweiterten Realität versteht man unter *Tracking* Methoden zur Bestimmung von Position und Orientierung (Pose) eines Betrachters, die es ermöglichen, grafische Informationen mittels verschiedenster Displaytechniken lagerichtig in dessen Sichtfeld einzublenden. Die präzisesten Tracking-Ergebnisse liefern Methoden der Bildverarbeitung, welche in der Regel nur die Pixel des Kamerabildes zur Informationsgewinnung heranziehen. Der Bildentstehungsprozess wird bei diesen Verfahren jedoch nur bedingt oder sehr vereinfacht miteinbezogen. Bei modellbasierten Verfahren hingegen, werden auf Basis von 3D-Modelldaten Merkmale identifiziert, ihre Entsprechungen im Kamerabild gefunden und aus diesen Merkmalskorrespondenzen die Kamerapose berechnet. Einen interessanten Ansatz bilden die Strategien der *Analyse-durch-Synthese*, welche das Modellwissen um Informationen aus der computergrafischen Bildsynthese und weitere Umgebungsvariablen ergänzen.

Im Rahmen dieser Arbeit wird unter Anwendung der *Analyse-durch-Synthese* untersucht, wie die Informationen aus dem Modell, dem Renderingprozess und der Umgebung in die einzelnen Komponenten des Trackingsystems einfließen können. Das Ziel ist es, das Tracking, insbesondere die Merkmalssynthese und Korrespondenzfindung, zu verbessern. Im Vordergrund steht dabei die Gewinnung von visuell eindeutigen Merkmalen, die anhand des Wissens über topologische Informationen, Beleuchtung oder perspektivische Darstellung hinsichtlich ihrer Eignung für stabiles Tracking der Kamerapose vorhergesagt und bewertet werden können.

Abstract

In the context of augmented reality we define *tracking* as a collection of methods to obtain the position and orientation (pose) of a user. By means of various displaying techniques, this ensures a correct visual overlay of graphical information onto the reality perceived. Precise results for calculation of the camera pose are gained by methods of image processing, usually analyzing the pixels of an image and extracing features, which can be recognized over the image sequence. However, these methods do not regard the process of image synthesis or at least in a very simplyfied way. In contrast, the class of model-based methods assumes a given 3D model of the observed scene. Based on the model data features can be identified to establish correspondences in the camera image. From these feature correspondences the camera pose is calculated. An interesting approach is the strategy of analysis-by-synthesis, regarding the computer graphics rendering process for extending the knowledge about the model by information from image synthesis and other environment variables.

In this thesis the components of a tracking system are identified and further it is analyzed, to what extend information about the model, the rendering process and the environment can contribute to the components for improvement of the tracking process using analysis-by-synthesis. In particular, by using knowledge as topological information, lighting or perspective, the feature synthesis and correspondence finding should lead to visually unambiguous features that can be predicted and evaluated to be suitable for stable tracking of the camera pose.

Danke

Prof. Dr. Stefan Müller für Rat und Tat, das Fördern und Fordern, Ansporn und Motivation, Diskussion und offene Ohren.

Den Gutachter:innen für Lob und Kritik.

Meinen Kolleg:innen und Freund:innen der Universität Koblenz für die gute Zeit, den Spaß und eure Unterstützung.

Ganz besonders Dominik Grüntjens, Diana Röttger und Brigitte Jung.

Allen Studierenden und HiWis für ihre Beiträge und Anregungen durch Abschlussarbeiten, Seminare und Praktika.

Ganz besonders Jan Hoppenheit, Bernhard Reinert und Kati Hebborn.

Meinen Eltern - einfach für alles.

Ermöglicht durch eine Forschungsförderung der Deutschen Forschungsgemeinschaft (DFG), Kennzeichen MU2783/3-1.

Einige 3D Modelle des Campus Koblenz wurden von den Arbeitsgruppen Aktives Sehen und Labor Bilderkennen bereitgestellt.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Im Kontext der Augmented Reality werden mit Methoden des sogenannten Trackings die Position und Orientierung (Pose) eines Betrachters erfasst, um grafische Informationen lagerichtig in dessen Sichtfeld einzublenden. Übliche Verfahren realisieren dies in der Regel, indem Unterschiede und Auffälligkeiten in den Pixelintensitäten eines Kamerabildes zur Erkennung und Beschreibung von Merkmalen (Features) herangezogen werden und die Pose aus einer Menge von Korrespondenzen über eine Bildsequenz berechnet wird. Bei diesem Vorgehen kann im Voraus keine Aussage über die zu erwartende Qualität und Zuverlässigkeit getroffen werden, mit der solch ein erkanntes Merkmal in den folgenden Bildern der Sequenz einer Korrespondenz zugeordnet werden kann (Matching) und sich damit zur robusten Poseschätzung eignet. Der Ansatz der *Analyse-durch-Synthese* geht von der Verfügbarkeit eines 3D-Modells der zu trackenden Szene aus. Zu jedem Kamerabild kann so anhand der letzten bekannten Pose eine synthetische Referenz des Modells erzeugt werden. Besondere Vorteile sind die Vermeidung des Aufsummierens von Fehlern über die Zeit, was zu einer Drift der Pose führt, sowie das Setzen einer absoluten Initialreferenz, während beim sequentiellen Tracking immer nur die relative Bewegung zum letzten Kamerabild bekannt ist.

Im Rahmen dieser Arbeit werden Methoden entwickelt, welche Informationen aus dem Modell, dem Renderingprozess und der Umgebung heranziehen, um das Tracking zu verbessern. Die untersuchten Schwerpunkte liegen dabei auf der Synthese von Merkmalen, welche besonders gut für das Tracking geeignet sind, sowie auf der Korrespondenzfindung

im Kamerabild. Weitere Aspekte betreffen die Durchführung einer GPS-gestützten Initialisierung des Trackingsystems in einem Outdoor-Szenario und die Analyse der optimalen Parameter für die Poseberechnung.

Es wird ein intelligentes Merkmalsmanagement entwickelt, das eine Bewertung der Merkmale anhand der gesammelten Informationen über topologische Anordnung der Modellgeometrie, Renderingparameter wie Beleuchtung und Perspektive, Korrespondenzqualität und die zeitliche Entwicklung der Merkmalsqualität durchführt. Somit kann eine möglichst minimale, jedoch qualitative Menge an Merkmalen ausgewählt werden, welche für die jeweilige Situation am besten geeignet erscheint. Die Vorhersage der Zuverlässigkeit von Merkmalen, die mit einer sehr großen Wahrscheinlichkeit im Kamerabild eindeutig wiedergefunden werden können, erhöht die Präzision und Stabilität des Trackingergebnisses. Gleichzeitig wird durch die Priorisierung und Selektion auf der Menge der Merkmale die Datenlast reduziert und die Performanz erhöht, was sich besonders günstig auf den Einsatz mobiler Anwendungen und Geräte auswirkt.

1.2 Beitrag

Voruntersuchungen

Die Voruntersuchungen zu den grundsätzlichen Einsatzmöglichkeiten des Trackingansatzes der *Analyse-durch-Synthese* werden im Kapitel **Analyse-durch-Synthese** dargestellt, deren Ergebnisse unter dem Titel „Analysis by Synthesis Techniques for Markerless Tracking“ auf dem *6. Workshop Virtuelle Realität und Augmented Reality* der GI Fachgruppe VR/AR präsentiert wurden [SAM09]. Sie befassen sich insbesondere mit den Fragestellungen bezüglich der anzuwendenden Optimierungsstrategie für die Berechnung der Pose, der eingesetzten Vergleichsmaße zwischen Merkmalen und zwi-

schen Bildern, sowie den Anforderungen an Modell und Rendering.

Es werden zwei denkbare Vorgehensweisen vorgestellt und untersucht. Einerseits kann zu jedem Kamerabild genau ein synthetisches Bild von der letzten bekannten Pose aus gerendert und mit dem Kamerabild verglichen werden, wobei sich die neue Pose durch den Einsatz eines Optimierungsalgorithmus aus der Minimierung des Fehlers zwischen den Merkmalen beider Bilder ergibt. Ein weiterer Weg ist das Streuen einer Anzahl neuer Posen um die zuletzt gültige, wobei zu jeder neuen Pose ein Bild gerendert wird und alle synthetischen Bilder mit dem Kamerabild verglichen werden. Über das Bild mit der geringsten Abweichung kann auf die optimierte Pose geschlossen werden. Als Maß für die Ähnlichkeit der Bilder stehen ein direkter, pixelbasierter Bildvergleich und der Einsatz gängiger Bildmerkmalsdetektoren zur Diskussion.

Die Ergebnisse der Untersuchungen zeigen deutlich, dass der Vergleich eines synthetischen Bildes mit dem Kamerabild auch unter korrekt simulierter Beleuchtung nicht hinreichend genau ist, um durchgängig stabiles und präzises Tracking zu ermöglichen. Zwischen einem gerenderten Bild und dem realen Kamerabild können nur unzureichende Korrespondenzen für eine Fehlerminimierung erstellt werden, da die Deskriptoren der Merkmale auf beiden Bildern nur in geringer Zahl übereinstimmende Ergebnisse liefern. Die Tests wurden mit bekannten Verfahren zur Detektion und Deskription von Bildmerkmalen sowohl unter approximierter diffuser Beleuchtung, als auch unter exakter Rekonstruktion der Lichtquelle mit einer High Dynamic Range Kamera durchgeführt. Als Folge wird vorgeschlagen, die Merkmale direkt aus der Geometrie des Modells zu gewinnen. Dadurch entfällt auch der doppelte Aufwand für Detektion und Deskription von Merkmalen auf zwei Bildern.

Unter den Methoden der direkten Bildvergleiche auf Basis der reinen Pixelintensitäten zeigte die Anwendung vorverarbeiteter Kantenbilder die höchste Präzision. Schlussfolgernd lässt sich sagen, dass die Extraktion von Merkmalen aus einem gerenderten Bild nicht zielführend ist, da nicht dieselben Merkmale wie im Kamerabild gefunden werden. Hingegen scheint der direkte Bildvergleich zumindest für eine nicht-kontinuierliche, grobe Bestimmung der Pose geeignet zu sein, wie sie bei der Initialisierung des Trackingprozesses notwendig ist.

Initialisierung

In Kapitel **Initialisierung** wird eine Lösung zur Initialisierung eines Trackingsystems vorgeschlagen, die unter dem Titel „Initialization of Model-Based Camera Tracking with Analysis-by-Synthesis“ auf dem *8th International Symposium on Visual Computing (ISVC)* vorgestellt wurde [SKM12].

Werden im Kontext des Trackings nichtlineare Optimierungsalgorithmen zur Berechnung der Pose eingesetzt, muss eine grobe Kamerapose beim Start des Trackingsystems vorgegeben sein, damit die Berechnung konvergiert. Die Umsetzung der Initialisierung wird ausgehend von den durchgeführten Voruntersuchungen mit einem ähnlichkeitsbasierten Bildvergleich auf Basis von Kantenbildern realisiert. In einem Outdoor-Trackingszenario mit vorhandenen Gebäudemodellen wird mit Hilfe von GPS und Kompassdaten, welche am Standort der realen Kamera gemessen werden, automatisch das für den Benutzer sichtbare Modell bestimmt und ausgerichtet, sodass ein kontinuierliches Posetracking gestartet werden kann.

Als erster Schritt wird eine Datenbasis aufgebaut, die zu jedem Modell die GPS Position und seine Ausrichtung bezüglich der Himmelsrichtung hält. Anhand der Abweichung zwischen aktuell gemessener Blickrichtung der realen Kamera und dem in der Datenbank eingetragenen Rotationswin-

kels des Modells, wird die Kamera entsprechend im Trackingkoordinatensystem ausgerichtet. Um die Abfrage in der Datenbank zu ermöglichen, ist zunächst das sichtbare Modell zu bestimmen.

Da die bereits durch GPS und Kompass erlangte Pose meist noch zu grob ist, um das kontinuierliche Tracking korrekt zu starten, muss ein Verfeinerungsschritt vorgenommen werden. Dazu werden synthetische Referenzbilder um die grobe Pose verteilt erstellt. Die so gerenderten Bilder werden mit dem Kamerabild verglichen, indem auf beiden Bildern eine Kantendetektion durchgeführt und die Anzahl der übereinstimmenden Kantenpixel gemäß der Stärke und Richtung ihres Gradienten ermittelt wird. Das über dieses Maß zurückgegebene Bild mit der größten Ähnlichkeit entspricht der Initialpose. In 75% der Testfälle konnte die mit dieser Methode vorgeschlagene Pose ohne manuelle Korrektur zum Start des Trackings verwendet werden. Bei dieser Form der Ähnlichkeitsbestimmung zwischen gerendertem und realem Kamerabild ist zu beachten, dass bei sehr detaillierten Modellen eine fehlerhafte Modellierung zu starken Abweichungen der Initialpose führen kann.

Poseberechnung

Mit der Problematik der Poseberechnung und ihren Algorithmen befasst sich das Kapitel **Pose**. Die beste Wahl der Parameter bei einer nichtlinearen Poseschätzung mit Punkt- und/oder Kantenmerkmalen wurde analysiert und unter dem Titel „Parameter and Configuration Analysis for Non-Linear Pose Estimation with Points and Lines“ auf der *7th International Conference on Computer Vision Theory and Applications (VISAPP)* vorgestellt [RSM12b].

Die erste zu klärende Frage betrifft die Wahl des Optimierungsschemas zur Posebestimmung. Da eine Minimierung des Fehlers über Merkmalskorrespondenzen aus Modell und Kamerabild durchgeführt werden soll, eig-

nen sich nichtlineare Optimierungsalgorithmen aufgrund ihrer Robustheit gegenüber fehlerhaften Korrespondenzen besonders. Bei der Realisierung stehen verschiedene Fehlermaße zwischen den Korrespondenzen, sowie verschiedene Parametrisierungen der zu optimierenden Poseparameter zur Auswahl. Es gilt, diejenige Parameterkombination zu finden, welche die präziseste Optimierung der Pose liefert. In der Diskussion sind mögliche Abstandsmaße im Bild- oder Objektraum. Bei Kantenmerkmalen kann der Fehler auch im Hough-Parameterraum angegeben werden. Die Parametrisierung der Rotation betreffend, muss eine geeignete Repräsentation gefunden und festgestellt werden, ob sie die Eigenschaften einer gültigen Rotation sicherstellt, oder ob diese durch zusätzlichen Aufwand erzwungen werden müssen. Mögliche Parametrisierungen der Rotationsparameter sind durch eine Rotationsmatrix, die Euler-Winkel-Darstellung, Quaternionen oder die Rodrigues-Formel gegeben. Des Weiteren ist die Frage zu beantworten, ob die Verwendung von Punkten oder Kanten zum Tracking erfolgversprechender ist, oder ob sich etwa eine Kombination aus beiden als sinnvoll erweist.

In Tests werden der Einfluss des Merkmalstyps, der Merkmalsanzahl, sowie der fehlerhafter Korrespondenzen auf die Robustheit der Pose bestimmt. Dazu werden unterschiedliche Mengen idealer und künstlich verrauschter Korrespondenzen als Eingabe für die Poseschätzung generiert. Es wird eine Kombination aus dem Levenberg-Marquardt Algorithmus, einer Rotationsparametrisierung nach Rodrigues und der Verwendung eines pixelbasierten Abstandsmaßes sowohl für Punktmerkmale als auch für Geradenmerkmale vorgeschlagen, da diese zu den besten Resultaten führt.

Das Ergebnis ist ein Poseschätzer, der die vorgeschlagene Kombination aus Optimierung, Abstandsmaß und Parametrisierung verwendet und als

Eingabemenge eine minimale, aber beliebig kombinierte Anzahl aus Punkt- und/oder Kantenmerkmalen akzeptiert. Zusätzlich werden die Eingabekorrespondenzen anhand des Modellwissens auf kritische Konfigurationen hin analysiert. Bei der Analyse stellte sich heraus, dass sich Kanten besser für stabiles Tracking eignen. Eine durch Punktmerkmale berechnete Pose kann durch die Hinzunahme von Kantenmerkmalen hinsichtlich ihrer Präzision verbessert werden, was im umgekehrten Fall nicht eintritt.

Merkmalsmanagement

Die Möglichkeiten der Bewertung von Merkmalen im Kontext des modellbasierten Trackings werden in Kapitel **Merkmalsmanagement** erfasst und ein entsprechendes Merkmalsmanagement entworfen und getestet. Dieses wurde unter dem Titel „Intelligent Feature Management for 3D Model-Based Camera Pose Tracking“ auf der *9th International Conference on Computer Vision Theory and Applications (VISAPP)* veröffentlicht [SHM14].

Da es während des Trackingvorgangs zu Problemen bei der Korrespondenzsuche kommen kann, wenn die Merkmale im nächsten Bild nicht eindeutig wiederzuerkennen sind, soll das Wissen über Modell und Rendering auf das Matching und die Bewertung der Merkmale angewandt werden. Üblicherweise kann das Problem reduziert werden, indem eine möglichst große Anzahl an Merkmalen eingesetzt wird. Die Annahme, dass bei einem rein quantitativen Vorgehen auch eine ausreichende Anzahl zufällig korrekter und zum Tracking geeigneter Merkmale gefunden wird, steht jedoch im Widerspruch zur Bestrebung, Echtzeitfähigkeit zu erreichen. Daher soll ein intelligentes Merkmalsmanagement das planvolle Erzeugen von nur wenigen, aber dafür qualitativ besonders gut geeigneten Merkmalen ermöglichen.

Aus einem vorhandenen Modell werden Informationen über die Beschaf-

fenheit und topologische Anordnung gesammelt, sowie beeinflussende Renderingparameter wie Beleuchtung und Perspektive oder die Verdeckungswahrscheinlichkeit berücksichtigt, um ein Qualitätsmaß zu definieren. Anhand dessen können diejenigen Merkmale selektiert und priorisiert werden, die für die Detektion im Bild als sehr gut erkennbar gelten und somit für das Tracking besondere Stabilität versprechen. Grundlage ist eine Liste aus Kantenmerkmalen, welche mit einem Deskriptor-Vektor annotiert werden, der Einträge für die Qualitätskriterien enthält. Auf diesen wird das Qualitätsmaß der Merkmale berechnet. Nur Merkmale, die eine Mindestqualität erfüllen, werden zur Korrespondenzsuche herangezogen.

Die betrachteten Kriterien im Deskriptor sind Länge, Distanz, Silhouette, Richtung, Beleuchtung und der Erfolg der Korrespondenzsuche, sowie der zeitliche Verlauf der Merkmalsqualität. Zum Beispiel ist die Distanz zwischen Kanten entscheidend, da ein geringer Abstand zu einer Mehrdeutigkeit bei der Korrespondenzsuche führen kann. Auch die Lichtsituation kann großen Einfluss auf die Erkennbarkeit der Kanten im Bild haben. Eine Kante zwischen zwei angrenzenden Flächen, die etwa gleich stark beleuchtet sind, wird schwer zu erkennen sein. Es wird daher getestet, wie groß der Beleuchtungsunterschied auf den benachbarten Flächen einer Kante ist.

Es gilt herauszufinden, welche der Qualitätskriterien für das Tracking von besonderer Bedeutung sind und wie sie daher zu gewichten sind. In Tests wurde der Einfluss der Kriterien im Merkmalsmanagement mit variierten Gewichtungen getestet und die Präzision der berechneten Pose aufgezeichnet. Im Vergleich zum Tracking ohne Merkmalsmanagement, also der Verwendung aller Modellmerkmale ohne Filterung, konnte das Ergebnis der Pose verbessert werden. Es tritt weniger Rauschen auf und die Pose konnte auch in schwierigen Szenen erfolgreich getrackt werden. Die Er-

gebnisse zeigen, dass die Einbeziehung von Wissen über Modell, Rendering und Umgebung in die einzelnen Komponenten des Trackingprozesses und eine Evaluierung der Merkmale anhand der vorgeschlagenen Qualitätskriterien die Verwendung von beliebig komplexen Modellen ermöglicht, welche nicht speziell für den Einsatz in einem Trackingszenario erstellt wurden.

Korrespondenzsuche

Ein neuer Ansatz zur Korrespondenzsuche zwischen Modellkantenmerkmalen und Geraden im Bild unter Nutzung der Grafikhardware wird in Kapitel **Korrespondenzsuche** vorgeschlagen, welcher unter dem Titel „A Matching Shader Technique for Model-Based Tracking“ auf der *20th International Conference on Computer Graphics, Visualization and Computer Vision* publiziert wurde [SHM12].

Gängige Verfahren zur Bestimmung von Korrespondenzen unter Kantenmerkmalen, wie etwa der Moving Edges Algorithmus und seine Weiterentwicklungen, tasten das Bild auf orthogonalen Suchlinien ab, die entlang der projizierten Modellkante gesampelt werden. Diese sind daher nicht in der Lage, eine Rückmeldung über die genaue Pixellänge der Kante im Bild zu geben. Für die Bewertung der Korrespondenzqualität muss der Matcher jedoch für jedes Modellmerkmal die Anzahl der im Bild gefundenen Pixel zurückgeben können. Unter Einbeziehung des Modellwissens ist die zu erwartende Länge eines Modellmerkmals nach der Projektion bekannt und das Verhältnis zur tatsächlich gefundenen Länge im Bild kann dann als Kriterium angegeben werden.

Um dies zu realisieren, wurde ein shaderbasierter Kantenmatcher entwickelt. Auf einem Canny-Kantenbild der Kamera wird ein Shader für jedes projizierte Modellmerkmal aufgerufen und simuliert mögliche, durch Bewegung verursachte, Verschiebungen der Start- und Endpunkte der Kan-

tenmerkmale im Bild. Zwischen allen erzeugten Punkten werden ihre verbindenden Bildlinien berechnet und für jedes Pixel der resultierenden Linien überprüft, ob ein entsprechendes Kantenpixel auf dem Eingabebild zu finden ist. Die Summe der gezählten Pixel wird über eine Rückgabetextur effizient ausgelesen. Jedes Pixel dieser Rückgabetextur speichert die Ergebnissumme genau einer Kante, wobei anhand der Pixelkoordinaten die jeweilige Kante eindeutig adressiert wird. Über das Texturmaximum kann anschließend auf die Kantenkorrespondenz im Bild geschlossen werden.

2 Tracking

In der Augmented Reality (AR) wird die visuell wahrgenommene Realität durch computergraphische Einblendungen erweitert, um den Benutzer einer AR-Anwendung durch zusätzliche Informationen zu unterstützen. Die über eine Kamera aufgenommene reale Umgebung soll dabei derart mit virtuellen, gerenderten Objekten überlagert werden, sodass für den Betrachter der Eindruck visueller Kohärenz zwischen virtueller und realer Welt entsteht. Um eine exakte Überblendung der Realität vom Kamera- bzw. Betrachterstandpunkt aus zu ermöglichen, ist es erforderlich, die genaue Lage und Orientierung (Pose) der bildgebenden Kamera im Raum zu bestimmen. Dies wird durch sogenannte Trackingverfahren gewährleistet, die daher wesentlicher Teilaspekt der Forschung im Bereich der AR sind. In der Definition [Azu97] von AR werden als Kriterien für AR-Systeme neben der Kombination der realen Umgebung mit virtuellen Objekten und der Ausrichtung der virtuellen Objekte relativ zu den realen Objekten (Registrierung) auch die Interaktivität und Echtzeitfähigkeit des Systems genannt. Daraus lassen sich als wichtigste Eigenschaften Exaktheit, Robustheit und Schnelligkeit ableiten, welche Trackingverfahren erfüllen müssen, um für den Einsatz im Bereich der AR geeignet zu sein.

Realisiert werden kann das Tracking durch viele verschiedenartige Systeme, die sich nach [ZDHB08] in sensorbasierte und bildbasierte Trackingmethoden gliedern. Zu ersteren zählen magnetische, akustische, mechanische und Trägheitsverfahren, sowie Systeme, die auf Funkbasis (WLAN / GPS) operieren oder optische Sensoren für Infrarot-LEDs besitzen. Bildbasiertes Tracking, welches auf Eingangsbildern einer Kamera arbeitet, nimmt eine Sonderstellung gegenüber den anderen Verfahren ein, da es durch Rückkopplung mit der realen Umgebung dynamische Fehlerkorrektur auf den

Bilddaten durchführen kann (sog. closed-loop Systeme). Mit Hilfe von Methoden der Bildverarbeitung wird durch Minimierung des Fehlers zwischen dem aktuellen *Ist*-Zustand und dem definierten *Soll*-Zustand die Korrektur der Kamerabewegung abgeleitet. Eine Kombination aus mehreren der genannten Verfahren ist ebenfalls möglich (hybrides Tracking). Eine umfassende und tiefgreifendere Übersicht bietet [RDB01].

2.1 Markerbasiert

Robustes und echtzeitfähiges Tracking wurde zunächst durch markerbasierte Verfahren umgesetzt. Dazu werden künstliche Strukturen in Form von Markern in der Umgebung installiert, die durch ihre Geometrie oder Farbe besonders gut mit den Methoden der Bildverarbeitung im Kamerabild erkannt werden können (Abbildung 1). Da die dreidimensionale Konfiguration dieser Marker bekannt ist, kann so die Pose der Kamera relativ zum Marker berechnet werden. Die bekanntesten freien Bibliotheken zur Erstellung markerbasierter AR-Anwendungen sind das ARToolKit [KB99] mit seinen Erweiterungen ARToolKit Plus [WS07a] und AR-Tag [Fia04]. Da der Einsatz von Markern in groß dimensionierten Szenarien, in öffentlichen Umgebungen oder der Natur jedoch nicht immer praktikabel ist, hat die Erforschung des markerlosen Trackings immer mehr an Einfluss gewonnen.

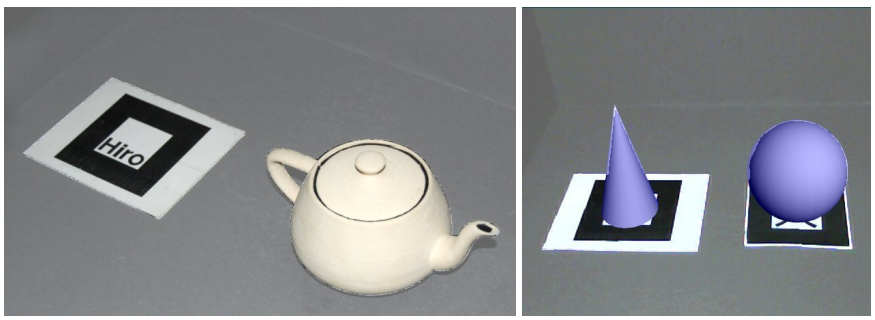


Abbildung 1: Marker und virtuelle Überlagerung

2.2 Markerlos

2.2.1 Sequentiell

In etablierten markerlosen Trackingverfahren, die auf reinen Methoden der Bildverarbeitung basieren, werden natürliche Bildmerkmale (Features) wie Punkte (nulldimensional), Linien (eindimensional), Segmente (zweidimensional) oder höherwertige Deskriptoren in zwei aufeinanderfolgenden Bildern einer Sequenz extrahiert und eine Suche nach entsprechenden Korrespondenzen durchgeführt. Aus der Lageänderung der Merkmale über die Zeit kann dann durch Fehlerminimierung die Kamerapose geschätzt werden. Das Detektieren dieser Merkmale geschieht jedoch nur über die Intensitätsauffälligkeiten im Bild, welches als einzige Informationsquelle bei der Synthese der Merkmale dient. Es fehlt ihnen jede Information über die zugrunde liegende geometrische Struktur, ihren topologischen Zusammenhang und es findet keine Berücksichtigung der beeinflussenden Renderingparameter wie etwa der Beleuchtung statt. Bei der Erzeugung dieser Merkmale kann im Voraus keinerlei Aussage über die zu erwartende Zuverlässigkeit getroffen werden, mit der solch ein erkanntes Merkmal in den folgenden Bildern wieder auffindbar sein wird und sich damit zur robusten Verfolgung eignet.

Daher ist das Erkennen der Merkmale in den beiden zum Vergleich herangezogenen Kamerabildern aufgrund von Bildstörungen und anderen Umgebungseinflüssen nicht völlig zuverlässig. Es kann zu Problemen bei der Korrespondenzsuche während des Trackingvorgangs kommen, wenn die Merkmale im nächsten Bild nicht eindeutig wiederzuerkennen sind. Das Problem kann reduziert werden, indem eine möglichst große Anzahl an Merkmalen generiert wird, um eine ausreichende Anzahl zufällig korrekter Merkmale zu treffen. Der hier vorgestellte Ansatz hingegen soll ein

planvolles Erzeugen von nur wenigen, aber dafür qualitativ besonders gut geeigneten Merkmalen ermöglichen. Dazu wird aus einem vorhandenen Modell eine synthetische Vergleichsreferenz zum Kamerabild erzeugt. Aus dem Wissen über die Eigenschaften des vorhandenen Modells sowie der Berücksichtigung beeinflussender Renderingparameter können so Merkmale abgeleitet werden, die für die Detektion im Bild als sehr gut erkennbar gelten. Gleichzeitig wird die Korrespondenzsuche zusätzlich vereinfacht, da durch die Kenntnis der Lage der Merkmale aus der Synthese eine Einschränkung des Suchraums im Kamerabild vorgenommen werden kann.

Dabei findet eine Gewichtung der Merkmale nach ihrer zu erwartenden Qualität statt, indem aus den in die Bildsynthese einfließenden Erzeugungsparametern eine Vorhersage über ihre Stabilität getroffen wird. In [ST94] wurde bereits der Versuch unternommen, anhand eines Abstandsmaßes die Qualität von Bildmerkmalen zu bestimmen und eine nachträgliche Auswahl zu treffen. Dazu wurde die Ähnlichkeit eines Merkmals im ersten Bild der Trackingsequenz zu seinem Auftreten im aktuellen Bild verglichen. Bei zu starker Veränderung wurde das Merkmal verworfen und für das Tracking nicht weiter beachtet. Doch auch in diesem Ansatz werden nur augenblicklich durch ihre Intensitätsabweichung *interessante* Bereiche im Kamerabild zur Generierung vieler Merkmale herangezogen und im Nachhinein eine Aussortierung durchgeführt. Stattdessen wäre es sinnvoller, von vornherein nur diejenigen Merkmale zu erzeugen, die unter Berücksichtigung aller bekannten Einflussfaktoren die beste Information für die Durchführung stabilen Trackings liefern.

Ein weiterer Nachteil des Trackings über Bildsequenzen ist, dass fehlerhaft erkannte Merkmale und Korrespondenzen Einfluss auf alle folgenden Ergebnisse haben. Die Summierung der Fehler von Bild zu Bild führt zu

einer Drift der Kamerapose. Wünschenswert ist es daher, an Stelle zweier suboptimaler Eingangsbilder, die zur Bestimmung der Pose verglichen werden, einen störungsfreien Prototyp als Referenz für jedes einzelne Kamerabild heranzuziehen. Das Wissen, welches aus einem synthetisch generierten Prototyp hervorgeht, kann zur Verbesserung der Posebestimmung beitragen. Einerseits wird durch den Vergleich jedes Kamerabildes mit der Modellreferenz das Driften der Kamerapose vermieden. Gleichzeitig wird die Initialisierung anhand des Modells ermöglicht, ohne dass zusätzliche Annahmen über die Umgebung getroffen werden müssten. Während bei rekursivem sequentiellem Tracking der Merkmale die Analyse der Pose nur relativ zum ersten Bild erfolgt, kann anhand des Bezugs zum Modell eine globale Posebestimmung der Kamera im Raum vorgenommen werden und so eine absolute Verortung in der Welt stattfinden.

2.2.2 Modellbasiert

Die modellbasierten Methoden konstruieren ihre Modelle zumeist aus Linien und Kantenzügen, zu denen nach der Projektion ins Kamerabild Entsprechungen anhand der Detektion starker Gradienten gesucht werden und sich die Kamerapose aus der Minimierung des Distanzfehlers ergibt. In [CMC03] kommt ein CAD Modell zum Einsatz, das komplexe Strukturen des zu trackenden Objekts wie Linien, Kreise, Zylinder und Kugeln stückweise parametrisch beschreibt und zu deren Beschreibung punktweise Übereinstimmungen im Bild gesucht werden. Aus dieser Arbeit wurde bereits ersichtlich, dass das Wissen über die Szene zur Verbesserung des Trackings beitragen kann, was Stabilität und Geschwindigkeit betrifft.

Ansehnliche Erfolge mit Tracking auf einem Linienmodell in Kombination mit Punktmerkmalen konnten [VLF04] verzeichnen, sowie [RD06], die

Kanten aus einem texturierten Modell der Umgebung gewinnen. In [DC02] und [WS07b] werden CAD Modelle verwendet, wobei letztere die Konturen beim Rendern aus den Daten des Tiefenpuffers extrahieren. Diese Konturen werden von der letzten korrekten Kamerapose aus gerendert und in das aktuelle Kamerabild projiziert. Die Minimierung des Fehlers zwischen projizierten Konturlinien und Gradientenlinien im Bild führt zur Berechnung der neuen Pose. Zumeist wird die Korrespondenzsuche zwischen Modellmerkmal und Bildgradient anhand von orthogonalen Suchlinien realisiert, die entlang des projizierten Modellmerkmals aufgespannt werden. Der sogenannte *Moving Edges Algorithmus* [Bou89] wurde unter anderem von [HS90] und [MBCM99] zum 3D-2D Tracking eingesetzt. Dabei kann die Korrespondenzsuche verbessert werden, indem zu einer Modellkante für jede orthogonale Suchlinie mehrere Gradientenpunkte im Bild als in Frage kommende Korrespondenzen abgewogen werden.

Es werden bei den vorgestellten Vorgehensweisen jedoch keine weiteren Eigenschaften des Modells bei der Erzeugung der Linienmerkmale berücksichtigt, so dass das Modell als ganzheitlich betrachtet wird. Die fehlende Selektion und Gewichtung der Linienmerkmale nach ihrer Qualität führt dazu, dass das Tracking instabil wird, wenn im Bild nicht genug korrespondierende Kantenlinien gefunden werden. Im umgekehrten Fall können keine eindeutigen Korrespondenzen gefunden werden, wenn durch ein sehr detailreiches Modell zu viele Kanten generiert werden. Auch hier könnte eine Selektion der Merkmale Abhilfe schaffen. Markerloses Tracking ohne direkten Modellbezug kann auch anhand von Referenzbildern durchgeführt werden, die verschiedene Ansichten der Umgebung von bekannten Kamerastandpunkten aus repräsentieren [Str01]. Die aktuellen Kamerabilder werden dazu mit den vorab in einer Bilddatenbank gespeicherten Referenz-

bildern verglichen. Die Pose desjenigen Bildes in der Datenbank mit der größten Ähnlichkeit wird für eine Fehlerminimierung herangezogen.

Bei weiteren Ansätzen ohne vorhandenes Modell muss vor dem eigentlichen Tracking eine Referenzrepräsentation der realen Welt erzeugt werden. Durch eine Lernphase werden in [Gen02] durch die Aufnahme der Umgebung vor dem Trackingvorgang 2D Merkmale extrahiert, zu denen mit Hilfe von Markern in der Welt 3D Referenzen erzeugt werden. Danach kann das Tracking anhand des erlernten Merkmalsmodells der Welt ohne den Einsatz der Marker durchgeführt werden. Dieser Ansatz erlaubt auch eine Reinitialisierung, falls das Tracking fehlschlägt, ist aber gleichzeitig auf den Bereich der Umgebung beschränkt, der in der Lernphase erkundet wurde. In [GL04] wird ein Weltmodell ohne Marker aufgebaut. Während der Lernphase wird mit Methoden der Struktur aus Bewegung die Szenenstruktur erstellt. Aus in der Lernphase durch den Benutzer aufgenommenen Referenzbildern werden SIFT Merkmalspunkte extrahiert. Deren 3D Koordinaten und entsprechende Kameraposen der Aufnahmepunkte werden aus den Korrespondenzen in mehreren Bildern rekonstruiert. Der Benutzer muss dann noch manuell die Lage des virtuellen Objekts relativ zum Modell bestimmen. Während des eigentlichen Trackings werden die aktuell im Kamerabild gefundenen Merkmale im Modell gesucht und die Pose durch Minimierung errechnet. Auch hier ist durch das Referenzbild mit den meisten Merkmalsübereinstimmungen zum aktuellen Kamerabild ein Reinitialisierungspunkt gegeben, falls das Tracking fehlschlägt.

2.2.3 Rekonstruktiv

Neuere Ansätze kommen ohne Vorverarbeitung aus. Ein Schwerpunkt der aktuellen Forschung liegt auf der Verwendung von SLAM (Simultaneous

Localisation and Mapping) Algorithmen. In der Robotik dienen diese dazu, dass ein mobiler Roboter durch Sensoren wie etwa Ultraschall eine Karte der unbekannten Umgebung erstellt und seine eigene Position in dieser feststellt. Unter Einsatz einer bildgebenden Kamera kann diese Methode im Anwendungsfeld der AR genutzt werden, um Karten der visuellen Umgebung zu erstellen, die aus Kanten oder Merkmals-Punktewolken bestehen können. Durch Rekonstruktion werden Szenengeometrie und Kamerapositionen aus den Bilderfolgen bestimmt. Dabei ist jederzeit eine Erweiterung der Karte möglich (*extensible tracking*), wenn die Kamera eine bislang unbekannte Umgebung aufnimmt. In der Kamerabildfolge werden Keyframes gesetzt, auf denen eine epipolare Suche nach übereinstimmenden Merkmalen durchgeführt wird und deren 3D Koordinaten dann durch Triangulierung errechnet wird. Diese werden in die Karte eingetragen und dienen dann als Suchreferenz für die Merkmale im aktuellen Kamerabild. In [KM07] wird ein SLAM-basierter AR Ansatz aufgezeigt, der eine rein quantitative Ansammlung von tausenden solcher Punktmerkmale erzeugt (Abbildung 2). Die Arbeit von [SRD06] stellt einen SLAM-Ansatz mit Kanten vor. Wegen der enormen Menge an Daten ist diese Methode jedoch nur auf eine kleine Szene begrenzt und hat mit Verdeckungsproblemen, sowie fehlerhaften Einträgen in der Umgebungskarte zu kämpfen. Änderungen der bereits in die Karte aufgenommenen Szene führen ebenso zu Trackingfehlern, wie nicht erkannte Selbstverdeckung der kartographierten Merkmale. Besonders nachteilig wirken sich fehlerhaft in die Karte eingetragene Merkmale aus, die durch falsche Zuordnung von Korrespondenzen hervorgerufen werden.

Den rekonstruktiven Ansätzen fehlt bei der Erzeugung der Umgebungskarten das Modellwissen über die Welt, weshalb nur eine relative Kamerapose berechnet werden kann. Die fehlende Referenz zu Bezugskoordinaten

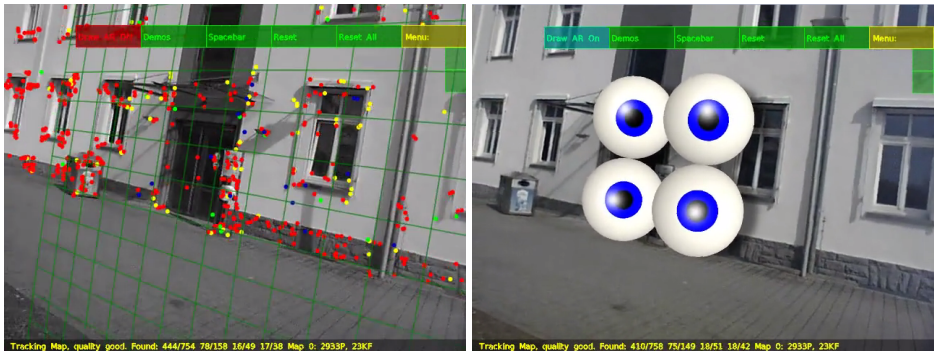


Abbildung 2: Punktewolke und Überlagerung bei SLAM

in der Welt (Initialisierung) wird durch die Erstellung einer geschätzten virtuellen Grundebene anhand der durchschnittlichen Punkteverteilung kompensiert. Die Tiefe der 3D Koordinaten wird jedoch durch Triangulierung bestimmt, was mit einer gewissen Unsicherheit verbunden ist. Denn um diesen Ungenauigkeitsfaktor einzuschränken, müssen die zur Bestimmung der Tiefe herangezogenen Merkmale über ein Minimum an Frames verfolgbar sein. Außerdem ist für die Rekonstruktion mit Hilfe der Struktur aus Bewegung eine ausreichende Translation der Kamera Voraussetzung, um Triangulierung durchführen zu können. Dies führt zu Problemen bei der Initialisierung, falls die Kamera vom Beginn des Trackings an nicht seitlich bewegt wird. Der Ansatz unter Verwendung eines bekannten Modells hingegen, lässt eine direkte und sichere Gewinnung der genauen Weltkoordinaten aus dem Modell zu.

Ohne jegliches Vorwissen über die Welt scheint in vielen realen Trackingsituationen eine sinnvolle Detektion von Merkmalen nur sehr eingeschränkt möglich zu sein. Daher soll die angedachte *Analyse-durch-Synthese* eine wissensbasierte Generierung von Merkmalen ermöglichen. Unter Berücksichtigung von Modell- und Umgebungsvariablen wird eine Voraussage darüber getroffen, wie zuverlässig Merkmale im Kamerabild gefunden wer-

den können und eine Selektion entsprechend ihrer Bewertung bereits bei ihrer Synthese durchgeführt.

3 Analyse-durch-Synthese

Der Ansatz der *Analyse-durch-Synthese* stammt ursprünglich aus dem Bereich der Sprachverarbeitung [BFH+61]. Das Ziel ist es, einen Satz von Parametern zu finden, mit denen sich die Frequenz eines gegebenen Lautes eindeutig beschreiben lässt. Um diese unbekannten Parameter zu bestimmen, wird folgendermaßen vorgegangen. Im Syntheseschritt wird ein Laut künstlich erzeugt und zur Analyse mit dem Eingabelaut verglichen. Aus der Abweichung beider Signale wird die Anpassung der zur Lautsynthese eingesetzten Parameter gesteuert. Dies wird solange wiederholt, bis der gemessene Fehler zwischen synthetischem und realem Laut gering genug ist.

Dieses Prinzip lässt sich mit Hilfe der Computergrafik auf den Trackingkontext übertragen. Von einer initialen Pose aus, die durch GPS-Verortung oder ein grobes Trackingverfahren geschätzt werden kann, wird das Bild eines gegebenen 3D Modells gerendert (Synthese) und mit dem realen Bild der Kamera abgeglichen (Analyse), um die aktuelle Kamerapose zu bestimmen (Abbildung 3). Dabei werden die intrinsischen Parameter der realen Kamera auf die virtuelle Kamera übertragen. Das Ziel ist es, die unbekannten externen Parameter (Pose) der bildgebenden Kamera mit Hilfe der bekannten Parameter der virtuellen Kamera zu bestimmen. Die zu optimierenden Parameter können als Pose-Vektor $p = (t_x, t_y, t_z, r_x, r_y, r_z)$ beschrieben werden, der sich aus Variablen für Position und Orientierung, der Translations- und Rotationswerte der Kamera, zusammensetzt.

Die *Analyse-durch-Synthese* eröffnet zwei grundsätzliche Vorgehensweisen für Optimierung und Bildvergleich. Im merkmalsbasierten Ansatz (Abbildung 4 links) wird die Pose durch die Minimierung des Fehlers der Korrespondenzen zwischen Merkmalen im synthetischen Bild und ihren

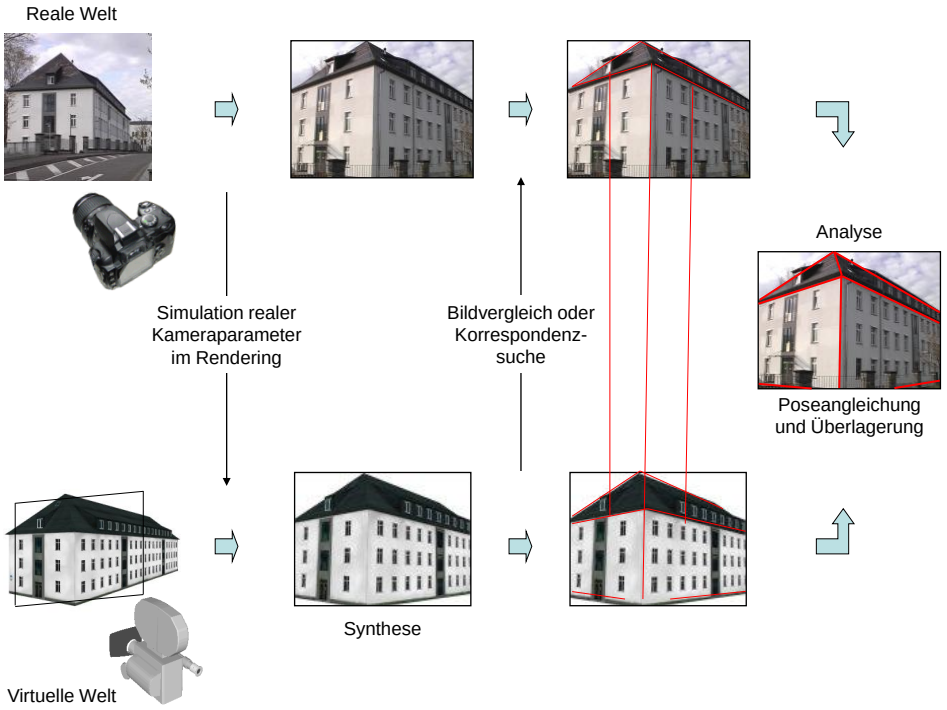


Abbildung 3: Analyse-durch-Synthese

Übereinstimmungen im realen Kamerabild berechnet. Ausgehend von einer Menge an Merkmalen f_r im gerenderten Bild mit der Pose p und den korrespondierenden Merkmalen f_c im Kamerabild, wird der Fehler E minimiert, um die neue Pose $\tilde{p}$ der Kamera zu bestimmen:

$$\tilde{p} = \underset{p}{\operatorname{argmin}}\, E(f_r(p), f_c)\,.$$

Der zweite Optimierungsansatz vergleicht mehrere synthetische Bilder auf Ähnlichkeit mit dem Kamerabild (Abbildung 4 rechts). Der Vergleich kann über den Einsatz von Bildmerkmalen oder ein pixelbasiertes Ähnlichkeitsmaß realisiert werden. Dazu wird die virtuelle Kamera in kleinen Schritten um die letzte korrekte Pose bewegt, um mehrere leicht variierende synthetische Bilder als Vergleichsreferenz zum Kamerabild zu rendern. Die Optimierung approximiert die Pose der realen Kamera durch iterati-

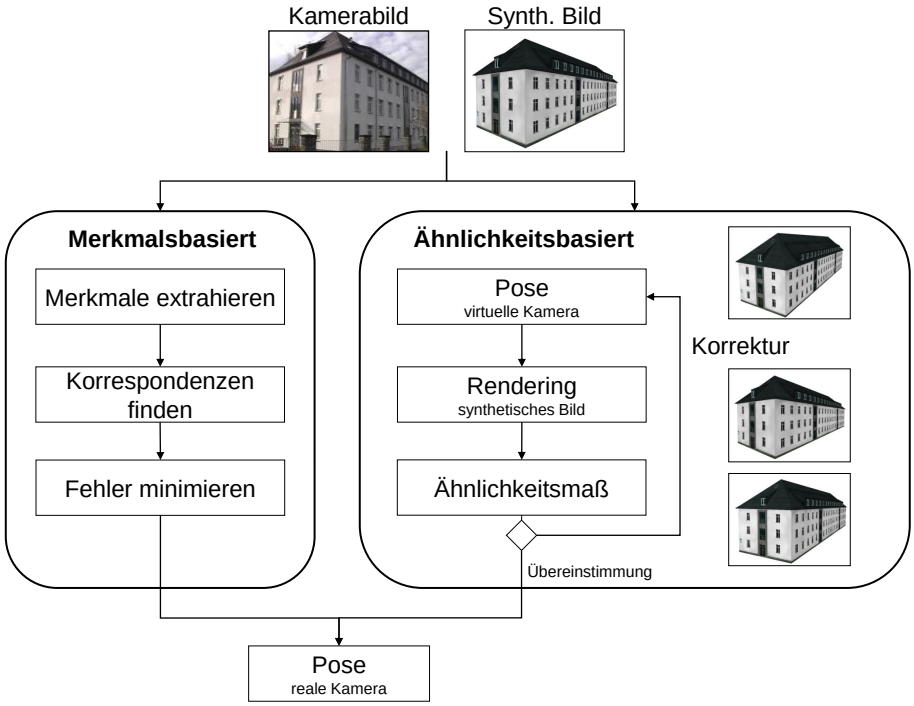


Abbildung 4: Ansätze des Bildvergleichs

ve Korrektur der bekannten virtuellen Pose. Über das Ähnlichkeitsmaß kann das gerenderte Bild mit der höchsten Übereinstimmung zum Kamerabild bestimmt werden, welches die gesuchte Pose liefert. Gegeben sei ein gerendertes Bild R mit der bekannten virtuellen Kamerapose p_r und das Kamerabild C mit der gesuchten Pose p_c, dann muss p_r solange optimiert werden bis die gemessene Ähnlichkeit S zwischen den Bildern maximiert wird:

$$p_r = \underset{p_r}{\operatorname{argmax}}\, S(R(p_r), C(p_c))$$

$$bis$$

$$p_r, p_c = (t_x, t_y, t_z, r_x, r_y, r_z)$$

$$R(p_r) \approx C(p_c) \quad \Rightarrow \quad p_r \approx p_c\,.$$

Die Grundannahme des Ansatzes der *Analyse-durch-Synthese* ist das Vorhandensein eines 3D Modells der Szene. Solche Modelle kommen oft in typischen Anwendungsfällen der Augmented Reality, wie den industriellen Bereichen der Planung, Montage, Lehre und Wartung zum Einsatz. Ein neueres Entwicklungsgebiet wird von touristischen Anwendungen eröffnet. Es ist abzusehen, dass in naher Zukunft sogar der Zugriff auf 3D Modelle ganzer Städte mit ihren Attraktionen und Sehenswürdigkeiten über Onlinedatenbanken wie etwa *Google Earth* zur Verfügung steht, was einen allgegenwärtigen Einsatz von Augmented Reality ermöglicht.

Aus dem Modell können bereits vielfältige Informationen bezogen werden, die später zur Bewertung der Merkmale dienen. Zusätzliches Wissen kann unter der *Analyse-durch-Synthese* aus dem Renderingprozess oder aus dem gerenderten Bild selbst gewonnen werden. So etwa die Kameraposition und Orientierung, die Perspektive, aber auch Beleuchtungs- und Oberflächeneigenschaften wie Reflektionsgrad oder Textur. Die Informationen über die Oberflächenbeschaffenheit des Modells können als Attribute der Materialien annotiert oder im diffusen Fall sogar automatisch berechnet werden [RG06]. Die Entwicklungen im Bereich der photorealistischen Bildsynthese ermöglichen das Sammeln weiterer Informationen über die Lichtquelle. Mit einer High Dynamic Range Kamera (HDR) lassen sich die realen Beleuchtungsverhältnisse rekonstruieren [HSK+05]. Mehrere Renderingverfahren wurden entwickelt, um präzise Schatten zu berechnen und komplexe Materialien mit Hilfe der GPU in Echtzeit darzustellen [RGKM07]. Sogar Raytracing-Verfahren wurden bereits unter dem Aspekt der Echtzeitfähigkeit um die Simulation globaler Beleuchtungseffekte in dynamischen Szenen erweitert [BAM08][SAM07]. In [GEM07] und [KSvA+08] wird die Beleuchtungsschätzung des Fernfeldes durch Nahfeldeffekte und indirekte

Beleuchtungseffekte ergänzt. Diese Arbeiten bilden die Grundlage, virtuelle Szenen mit komplexen Materialattributen unter Berücksichtigung der Beleuchtungssituation in Echtzeit und mit photometrischer und colormetrischer Konsistenz zu rendern.

Beispiele für die Anwendung von *Analyse-durch-Synthese* Techniken im Tracking können bei [KBK07] gefunden werden. Dort wird vorab ein Freiformflächen-Modell der Szene unter Nutzung einer Fischaugenkamera mit Methoden der Struktur aus Bewegung rekonstruiert. Anschließend wird ein synthetisches Fischaugen-Bild aus dem Modell gerendert, um mit dem Kamerabild verglichen zu werden. Der Vorteil einer Fischaugenkamera ist der große Öffnungswinkel, welcher es ermöglicht, einzelne Merkmale über einen längeren Zeitraum zu erkennen, als es bei einer perspektivischen Kamera der Fall wäre. In [SJBP99] wird das Modellieren und Tracken von Gesichtern mit einem *Analyse-durch-Synthese* Ansatz unter Verwendung der Normalen- und Tiefeninformation realisiert. Wuest und Stricker [WS07b][WWS07] nutzen die Modellinformation und den Renderingprozess, um eine tiefenbasierte Kontur des Modells zu extrahieren. Auf dem Tiefenbild des Modells wird eine Kantendetektion durchgeführt und entlang der Modellkontur auf senkrechten Suchlinien nach korrespondierenden Gradienten im Kamerabild gesucht. In der Arbeit von [BM11] werden die Lichtverhältnisse durch GPU-basierte HDR Beleuchtung simuliert, um das Tracking zu verbessern.

Ein Überblick von *Analyse-durch-Synthese* Techniken im Bereich des Motion Capturing menschlicher Bewegungen findet sich in [Moe99]. Mit Hilfe synthetischer Bilder wird das Erkennen von Bewegungen ermöglicht. Ein animiertes Modell eines menschlichen Körpers wird gerendert und das synthetische Bild mit dem der Kamera verglichen, welche die Bewegungen

eines realen Menschen aufnimmt. Die Bewegung des 3D Modells wird solange korrigiert, bis sie mit der realen Bewegung genau genug übereinstimmt. Auch die Modellierung selbst kann mit dem Ansatz der *Analyse-durch-Synthese* verbessert werden. In [ea94] wird dazu eine Rekonstruktion der Modellgeometrie aus Stereobildern durchgeführt. Die erzeugten Polygone werden zu höheren Einheiten zusammengefasst. Mit dem gewonnenen Modell werden synthetische Bilder gerendert und mit den Kamerabildern verglichen. Anhand der Abweichung wird die Struktur des Modells und die Zuordnung der Elemente angepasst, um eine fehlerhafte oder unvollständige Modellierung zu korrigieren.

3.1 Voruntersuchungen

Zunächst sind die grundsätzlichen Einsatzmöglichkeiten von *Analyse-durch-Synthese* Techniken im markerlosen Tracking zu untersuchen. Die Voruntersuchungen befassen sich insbesondere mit den Fragestellungen der anzuwendenden Optimierungsstrategie, der eingesetzten Vergleichsmaße und den Anforderungen an Modell und Rendering. Bezüglich der Optimierungsstrategie gibt es die zwei bereits vorgestellten Vorgehensweisen. Einerseits kann zu jedem Kamerabild genau ein synthetisches Bild von der letzten bekannten Pose aus gerendert und mit dem Kamerabild verglichen werden, wobei sich die neue Pose aus der Minimierung des Fehlers zwischen Merkmalen beider Bilder durch den Einsatz eines Optimierungsalgorithmus ergibt. Ein weiterer Weg ist das Streuen neuer Posen um die zuletzt gültige, wobei zu jeder neuen Pose ein Bild gerendert wird und alle synthetischen Bilder mit dem Kamerabild verglichen werden. Über das Bild mit der geringsten Abweichung kann auf die optimierte Pose geschlossen werden. Bei letzterem Vorgehen ist darüber hinaus die Frage zu beantworten, wie

der Vergleich der Bilder realisiert wird. Als Maß für die Ähnlichkeit der Bilder stehen ein direkter, pixelbasierter Bildvergleich und der Einsatz gängiger Bildmerkmalsdetektoren zur Diskussion. Zu testen sind weiterhin die Anforderungen an die Genauigkeit des Modells und die Beleuchtungseigenschaften. Die Ergebnisse wurden ausgehend von den Vorarbeiten [Ach08] und [Sch08] unter dem Titel „Analysis by Synthesis Techniques for Markerless Tracking “ auf dem *6. Workshop Virtuelle Realität und Augmented Reality* der GI Fachgruppe VR/AR vorgestellt [SAM09].

3.1.1 Merkmalsbasierte Optimierung

Der merkmalsbasierte Ansatz realisiert das Tracking durch Korrespondenzsuche nach übereinstimmenden Merkmalen in Kamerabild und gerendertem Bild. Dazu soll analysiert werden, welche Merkmalsdetektoren und -beschreibungen gut geeignet sind, um Korrespondenzen zwischen den Bildern herzustellen. Weitere Tests betreffen die Aspekte des Detailgrades und den Einfluss von Texturen und Licht auf die Genauigkeit des Trackingergebnisses. Neben dem 3D Tracking der Kamerapose finden Merkmale auch in weiteren Gebieten, wie dem 2D Motion Tracking, der Objekterkennung und 3D Rekonstruktion, in Navigation und Robotik, sowie bei der Entzerrung und Indizierung von Bildern Anwendung. Merkmale beschreiben Strukturen im Bild, die sich von der Umgebung messbar, insbesondere durch große Intensitätsänderung, unterscheiden. Dies können etwa Punkte, Regionen oder Kanten sein. Ein Merkmalsdetektor findet solche Auffälligkeiten im Bild.

Da ein Merkmalsdetektor nur die Pixelposition des Merkmals zurückgibt, ist zusätzlich noch eine lokale Bildbeschreibung der Merkmalsnachbarschaft notwendig, um Merkmale identifizieren und vergleichen zu können.

Ein sogenannter Deskriptor ist ein Vektor, anhand dessen ein Vergleich mit anderen Merkmalsbescheibungen durchgeführt wird, um über deren Distanz ähnliche Merkmale zu Korrespondenzen zusammenzufassen. Für Detektoren und Deskriptoren gibt es zwei wesentliche Anforderungen, die nach [TM08] erfüllt sein sollten:

Wiederholbarkeit durch Invarianz

Wiederholbarkeit bezeichnet nach [SMB00] das Wiederauffinden desselben Merkmals in zwei Bildern unter den geometrischen Transformationen Rotation, Translation, Skalierung und perspektivischer Verzerrung oder photometrischen Veränderungen wie Bildkontrast und Beleuchtungsstärke. Diese Eigenschaften können durch einen entsprechenden Aufbau eines Merkmalsdetektors sichergestellt werden, damit dieser sich invariant gegenüber einer oder mehrerer der genannten Transformationen verhält.

Die Wiederholbarkeit ist messbar und dient der qualitativen Bewertung von Merkmalsdetektoren. Seien p_i und p_i' die Projektionen eines Weltpunktes W_p in zwei Bildern. Merkmalspunkt p_i aus dem Ursprungsbild gilt genau dann als wiederholbar, wenn der korrespondierende Merkmalspunkt p_i' innerhalb einer ϵ-Umgebung in einem weiteren Bild gefunden wird (Abbildung 5). Die Größe von ϵ bezeichnet die Lokalisationsgenauigkeit des Detektors. Die Menge $C(\epsilon)$ der wiederholbaren und damit als Korrespondenz geltenden Punkte ist demnach

$$C(\epsilon) = \{(p_i, p_i') \mid dist(Hp_i, p_i') < \epsilon\}.$$

Dabei ist zu beachten, nur diejenigen Merkmale zu vergleichen, die auch in beiden Bildern sichtbar sind, da der in beiden Bildern sichtbare Bereich etwa durch Rotation oder Skalierung unterschiedlich sein kann. Der

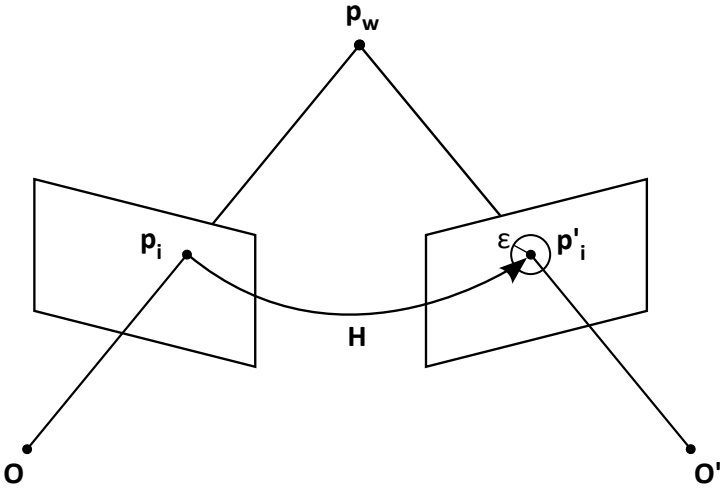


Abbildung 5: Definition der Merkmalskorrespondenz nach [SMB00]

gemeinsame Bereich wird durch die Homographie der Punkte bestimmt, die sich aufeinander abbilden lassen. Für den planaren Fall lässt sich die Relation zwischen p_i und p'_i als Homographie $p'_i = Hp_i$ beschreiben.

Die Wiederholbarkeitsrate $R(\epsilon)$ ist dabei die Anzahl der in zwei Bildern wiederholten Merkmale im Verhältnis zur Gesamtzahl n der gefundenen Merkmale:

$$R(\epsilon) = \frac{|C(\epsilon)|}{n}.$$

Ist die Anzahl der erkannten Merkmale in zwei Bildern unterschiedlich, werden die Merkmale des Bildes mit der geringeren Anzahl zur Berechnung der Wiederholbarkeitsrate herangezogen:

$$R(\epsilon) = \frac{|C(\epsilon)|}{\min(n_i, n'_i)}, \;\; 0 \leq R(\epsilon) \leq 1$$

mit der Anzahl n_i und n'_i der in den übereinstimmenden Bereichen beider Bilder gefundenen Merkmale.

Eindeutigkeit

Die Unterscheidbarkeit von Merkmalen beruht auf dem Informationsgehalt ihres Deskriptors. Merkmale sind eindeutig beschrieben, wenn die Wahrscheinlichkeit des Auftretens eines bestimmten Deskriptors in der Menge aller Deskriptoren gering ist. Je weniger Deskriptoren mit ähnlicher Beschreibung zu finden sind, desto eindeutiger ist die Korrespondenzsuche. Kommt ein Deskriptor also häufig vor, so ist die lokale Bildbeschreibung durch den Deskriptor schlecht gewählt. Dies gilt insbesondere, wenn zum Vergleich der Deskriptoren bildbasierte Ähnlichkeitsmaße, wie etwa die Korrelation verwendet werden [SMB00]. Dabei gilt, je komplexer ein Deskriptor ist, desto besser ist er eindeutig unterscheidbar. Mit der Dimension des Deskriptors steigt jedoch auch der Aufwand für Berechnung und Korrespondenzsuche.

Die im Folgenden kurz beschriebenen Merkmalsoperatoren werden auf beide Bilder angewandt: Harris Corner Detector, Förstner Operator, Kanade-Lucas-Tomasi Detektor (KLT), Smallest Univalue Segment Assimilating Nucleus (SUSAN), Features from Accelerated Segment Test (FAST) und Scale-Invariant Feature Transform (SIFT).

Moravec Interest Operator

Die grundlegende Vorarbeit liefert ein von Moravec 1977 entwickelter, intensitätsbasierter Operator [Mor77]. Er berechnet die mittleren quadratischen Differenzen der Bildgrauwerte zwischen einem lokalen Suchfenster und dessen Verschiebung in die vier Hauptrichtungen. Liegt das Minimum V der vier Differenzsummen V_{1-4} über einem Schwellwert, ist also in alle Richtungen eine ausreichend starke Intensitätsdifferenz vorhanden, so handelt es sich um einen Eckpunkt. Das Verfahren ist jedoch weder rotations- noch skalierungsinvariant, rauschempfindlich und in seiner Genauigkeit

beschränkt.

$$V_1 = \sum_{i,j} (I(i,j) - I(i,j+1))^2$$
$$V_2 = \sum_{i,j} (I(i,j) - I(i+1,j))^2$$
$$V_3 = \sum_{i,j} (I(i,j) - I(i+1,j+1))^2$$
$$V_4 = \sum_{i,j} (I(i,j+1) - I(i+1,j))^2$$
$$V = \min\{V_1, V_2, V_3, V_4\}$$

Harris Corner Detector

Harris und Stephens entwickelten den Moravec-Operator weiter [HS88], indem die diskrete Intensitätsdifferenz durch die Berechnung einer Autokorrelationsmatrix A ersetzt wird. Dazu werden die Gradienten f des Bildes um einen Punkt betrachtet und die Summe der gaußgewichteten Ableitungen in seiner Nachbarschaft W gebildet. Das Verfahren ist rotationsinvariant, wenig empfindlich gegen Rauschen und erreicht damit eine höhere Genauigkeit.

$$A = \begin{bmatrix} \sum_{(i,j)\in W} f_x(i,j)^2 & \sum_{(i,j)\in W} f_x(i,j) f_y(i,j) \\ \sum_{(i,j)\in W} f_x(i,j) f_y(i,j) & \sum_{(i,j)\in W} f_y(i,j)^2 \end{bmatrix}$$

Anhand der Eigenwerte der Matrix kann abgelesen werden, ob es sich um einen Eckpunkt handelt. Dies ist der Fall, wenn beide Eigenwerte groß sind. Ist nur der erste Eigenwert nahe 0 und der Zweite groß, dann handelt es sich um einen Kantenpunkt, liegen aber beide Eigenwerte nahe 0, so ist

die betrachtete Pixelnachbarschaft homogen. Da die Eigenwertberechnung rechenintensiv ist, lässt sich ein Wert für die Stärke des Merkmals über Determinante und Spur der Matrix annähern (Parameter κ ist vorgegeben im Bereich $0.04 \leq \kappa \leq 0.15$):

$$V = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - \kappa\,\text{spur}(A)^2.$$

Kanade-Lucas-Tomasi Detector

Der von Shi und Tomasi entwickelte KLT-Detector [ST94] berechnet wie der Harris Corner Detecor die Autokorrelationsmatrix, die Stärke des Merkmals wird jedoch direkt über das Minimum der Eigenwerte $\min(\lambda_1, \lambda_2)$ bestimmt. Sofern dieses Minimum über einem Schwellwert liegt, wird es genutzt, um aus dem Bild eine Liste von Merkmalskandidaten zu erzeugen, die nach dem kleinsten Eigenwert sortiert ist. So ist es möglich, die Änderung der Merkmalsstärke über die Bildsequenz auszuwerten, was der Selektion stabiler Merkmalspunkte dient.

Foerstner-Operator

Ebenfalls auf der Berechnung der Autokorrelationsmatrix basiert der von Förstner und Gülch entwickelte Förstner-Operator [FG87]. Die Interessantheit eines Punktes wird hier durch die Definition einer Fehlerellipse ausgedrückt, die idealerweise möglichst rund und möglichst klein sein sollte. Form und Größe dieser Ellipse können dabei mit den Eigenwerten der invertierten Autokorrelationsmatrix bestimmt werden. Um den Aufwand zu minimieren, werden zur Berechnung ihrer Rundheit r und Größe g auch bei diesem Operator Vereinfachungen herangezogen:

$$r = \frac{1}{\lambda_1 + \lambda_2} = \frac{1}{\text{spur}(A^{-1})} = \frac{\det(A)}{\text{spur}(A)}$$

$$g = 1 - (\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2})^2 = \frac{4 * \det(A)}{\text{spur}(A)^2}.$$

Die Rundheit wird im Intervall [0,1] angegeben und beträgt 1, wenn die zu Grunde liegenden Eigenwerte gleich groß sind, also die Fehlerellipse einen Kreis darstellt, denn es liegen identische Eigenwerte vor. Als Schwellwert wird eine Rundheit von 0,8 für einen Interessenspunkt angenommen. Bei einer Kante ist mindestens einer der Eigenwerte und damit die Rundheit 0. Da der Wert für die Größe vom Bildkontrast beeinflusst wird, ist hier üblicherweise als Schwellwert der Größen-Mittelwert aller Interessenspunkte im Bild gesetzt. Verglichen mit dem Harris Corner Detector ist die Berechnung für den Förstner-Operator etwas langsamer, jedoch bei gleichzeitiger Verbesserung der Genauigkeit und Invarianz, sowie Robustheit gegenüber Rauschen.

SUSAN

Smith und Brady[SB97] stellten mit SUSAN (Smallest Univalue Segment Assimilating Nucleus) einen Merkmalsdetektor vor, der keine Ableitung der Gradienten berechnet, sondern auf Binarisierung der Pixelintensitäten beruht. Er ist ebenfalls rotationsinvariant und effizienter als ableitungsbasierte Detektoren, aber wesentlich rauschempfindlicher bei fehlender Bildschärfe.

Es wird eine kreisförmige Nachbarschaft mit festem Radius um jedes zu untersuchende Pixel definiert. Die Helligkeit aller Pixel der Nachbarschaft wird anhand eines Toleranzschwellwertes mit der Intensität des Zentrumspixels verglichen und so eine Klassifizierung in *ähnlich* oder *nicht ähnlich* vorgenommen. Die relative Größe des Bereichs mit ähnlicher Intensität zum

Zentrum gibt dabei Aufschluss über die Beschaffenheit der Nachbarschaftsregion. Ist der Bereich groß, so muss die Umgebung homogen sein. Bei etwa einem halben Anteil handelt es sich um ein Kantenpixel. Nimmt der Bereich ähnlicher Intensität ein Viertel oder weniger der Nachbarschaft ein, so wurde eine Ecke gefunden. Um die Fehleranfälligkeit zu reduzieren, wird eine Gauß-Gewichtung zum Zentrum innerhalb der Nachbarschaft vorgenommen und zusätzlich der Abstand des Schwerpunktes im ähnlichen Bereich zum Zentrum bewertet.

FAST

Der von Rosten und Drummond vorgestellte FAST-Detektor (Features from Accelerated Segment Test) [RD05] basiert auf der Methodik des SUSAN-Detektors. Er bietet die mit Abstand schnellste Berechnung gegenüber anderen Merkmalsdetektoren bei vergleichbarer Genauigkeit, reagiert jedoch wesentlich empfindlicher auf Bildrauschen.

Auch hier wird die Intensität zwischen Zentrum und den Umgebungspixeln verglichen. Dazu wird ein fester Kreis mit einem Radius von 3 Pixeln definiert, auf dessen Umkreis genau 16 Pixel liegen. Für diese wird geprüft, ob sie unter Berücksichtigung eines Schwellwertes heller oder dunkler als das Zentrumspixel sind. Üblicherweise gilt eine Ecke als erkannt, wenn mindestens 12 zusammenhängende Pixel des Kreises diese Bedingung erfüllen. Abschließend werden Nicht-Maxima unterdrückt, um zu dicht beieinander liegende Merkmalspunkte zu vermeiden. Dazu werden die Summen der absoluten Differenzen zwischen Zentrum und Umkreis der benachbarten Merkmale verglichen und das stärkere ausgewählt.

Beschleunigung kann der Detektor erfahren, indem die Anzahl der zu testenden Pixel reduziert wird. Dies kann unter anderem durch Algorith-

men des maschinellen Lernens geschehen. Auf einem Satz von Testbildern werden verschiedene Ecken detektiert und klassifiziert. Aus diesen Daten wird ein Entscheidungsbaum aufgebaut, mit dem sich die optimale Reihenfolge der zu testenden Pixel bestimmen lässt. So kann frühzeitig entschieden werden, ob es sich um eine Ecke handelt und unnötige Pixeltests werden vermieden.

SIFT

David Lowe entwickelte mit dem SIFT-Operator (Scale-Invariant Feature Transform) [Low99] eine Kombination aus Detektor und Deskriptor, die invariant gegenüber Rotation, Skalierung und Beleuchtungsänderungen ist. In geringem Maße ist diese auch gegen perspektivische Verzerrungen robust, jedoch im Vergleich zu anderen Verfahren sehr rechenintensiv. Um von der Skalierung unabhängige Merkmale zu finden, wird zunächst eine durch das Difference-of-Gaussians (DoG) Verfahren angenäherte Laplacepyramide aufgebaut. Dabei werden die Differenzen mehrerer unterschiedlich stark gaußgefilterter Versionen des Eingangsbildes erzeugt (Oktave), was unter Halbierung der Auflösung für weitere Skalierungsstufen der Pyramide wiederholt wird. Innerhalb der Oktaven einer Skalierungsstufe wird nun auf den Differenzbildern nach Merkmalspunkten gesucht, indem jedes Pixel mit seinen direkten 8 Nachbarn und den 9 entsprechenden Pixeln der nächst höheren und niedrigeren Ebene verglichen wird. Ein Merkmal stellt das absolute Intensitäts-Maximum oder Minimum in dieser Nachbarschaft dar. Es folgt eine Subpixel-Lokalisierung und eine Filterung von Kantenpixeln durch Betrachtung der Gradienten.

Die Rotationsinvarianz wird durch das Bestimmen der Hauptrichtung des Gradienten für jedes Merkmal sichergestellt. Dies geschieht, indem

auf der Nachbarschaft des Merkmals Stärke und Richtung der Gradienten berechnet werden. Sie werden anhand ihrer Richtung einem Histogramm mit 36 Einträgen (Bins) für Intervalle von je 10 Grad zugeordnet, wobei die Stärke als Wert im entsprechenden Bin gespeichert wird. Der Histogrammeintrag mit dem höchsten Wert gibt die Gradienten-Hauptrichtung an.

Für die Erstellung des Deskriptors wird um jeden Merkmalspunkt eine 16er Nachbarschaft gebildet, die aus Pixel-Fenstern der Größe 4x4 besteht. In jedem Fenster werden wiederum Stärke und Richtung der Gradienten berechnet, wobei deren Richtung nun relativ zur vorherberechneten Gradienten-Hauptrichtung des Merkmalspunktes gesetzt wird. Die Gradienten werden in ein Histogramm mit 8 Bins für Intervalle zu 45 Grad eingetragen und zusätzlich mit einer Gauß-Funktion gewichtet, um ihren Einfluss entsprechend der Entfernung vom Merkmalspunkt abzuschwächen. Die 128 Gradienten-Einträge der 16 Histogramme werden dann in einen Merkmalsvektor geschrieben. Dieser ist abschließend noch zu normalisieren, um Beleuchtungsinvarianz zu gewährleisten.

Mit SURF (Speeded Up Robust Features) existiert eine Weiterentwicklung von Bay [BETG08], die durch Verwendung von Integralbild-Techniken zwar deutlich performanter ist, jedoch weniger genau und nicht invariant gegenüber perspektivischen Verzerrungen [OR13].

Ergebnisse

Für die durchgeführte Untersuchung wird ein Merkmalsdeskriptor eingesetzt, der die gaußgewichtete Pixelnachbarschaft jedes Merkmals beschreibt. Eine Ausnahme bildet die SIFT-Implementierung, welche einen eigenen, bereits beschriebenen Deskriptor nutzt. Das Matching der Deskriptor-Vektoren

zur Korrespondenzfindung wird über die normalisierte Kreuzkorrelation (NCC) durchgeführt. Dabei ging aus den Tests eine optimale Größe der betrachteten Pixelnachbarschaft für den Deskriptor zwischen 9x9 und 11x11 Pixeln hervor, was einen Kompromiss aus Rechenzeit und Stabilität darstellt. Weiterhin wurde ein Suchfenster für das Matching von 30x30 Pixeln um die detektierten Merkmale gewählt. Einige der Korrespondenz-Ergebnisse sind in Abbildung 6 dargestellt.

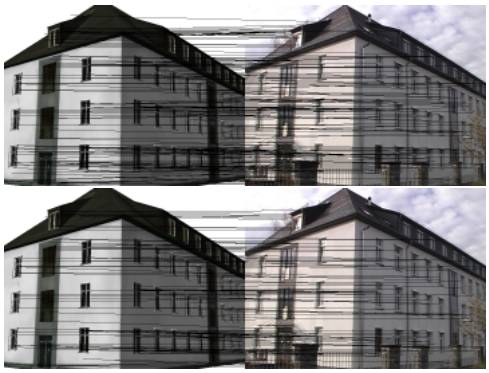


Abbildung 6: SIFT Korrespondenzen auf gerendertem und Kamerabild. Einschränkung auf 50x50 Pixel Suchfenster (links) und unter Einsatz von RANSAC (rechts).

Nach der Detektion von Merkmalen im synthetischen und im Kamerabild werden zunächst fehlerhafte Korrespondenzen mit dem Random Sample Consensus Algorithmus (RANSAC, siehe 9.2) eliminiert. Die 2D Merkmale des synthetischen Bildes werden auf das Modell rückprojiziert, um die 3D Koordinaten der Merkmale zu bestimmen. Der Fehler der resultierenden 2D-3D Korrespondenzen wird anschließend zur Poseschätzung mit einem robusten Tukey Estimator unter einer Downhill-Simplex Optimierung minimiert. Die initiale Pose wird bei dieser nichtlinearen Optimierung als bekannt vorausgesetzt.

In jedem Bild einer Testsequenz mit 50 Frames wird der Rückprojektionsfehler in Pixeln gemessen, aufsummiert und als Root Mean Square Error

(RMSE) angegeben. Nur zu 10-20% der in beiden Bildern detektierten Merkmale konnten überhaupt Korrespondenzen erstellt werden (Tabelle 1). Da es sich um CPU-basierte Implementierungen handelt, liegen die Frameraten zwischen 0,2 FPS (SIFT) und 1 FPS (Harris).

	Merkmale synth. Bild		Merkmale Kamerabild		zugeordnete Korrespondenzen
Harris	800	17ms	800	18ms	125
SIFT	450	1,5s	2200	3,3s	120
KLT	800	23ms	800	25ms	110
Foerstner	900	55ms	1200	55ms	90
SUSAN	530	39ms	1200	49ms	60
FAST	830	7ms	1000	7ms	50

Tabelle 1: Anzahl erkannter Merkmale und zugeordneter Korrespondenzen

Obwohl durchgehend eine ausreichende Menge an Merkmalen auf beiden Bildern gefunden wurde, schnitten die Detektoren FAST und SUSAN mit lediglich 5% Korrespondenzen auf der Merkmalsmenge am schlechtesten ab. Insgesamt zeigten der Harris Corner Detektor und SIFT die besten Ergebnisse, wobei SIFT verhältnismäßig viele Korrespondenzen aus einer stark variierenden Menge von Merkmalen auf beiden Bildern erstellt. Darüber hinaus ist anzumerken, dass besonders SUSAN und KLT eine schlechte Verteilung der Merkmale aufweisen, obwohl letzterer prinzipiell genügend Korrespondenzen liefern würde. Ein Vergleich der Fehler verschiedener Merkmalsdetektoren über die gesamte Bildsequenz ist in Abbildung 7 dargestellt.

Die Merkmale wurden weiterhin auf den Einfluss der Beleuchtungssimulation und Textur getestet. Ambiente Beleuchtung hat nur eine geringe Auswirkung auf die Genauigkeit der Merkmale, wobei SIFT und FAST leicht profitieren. Die Lichtsituation wurde sowohl grob approximiert als auch auch mit Hilfe einer HDR-Kamera exakt rekonstruiert [Hab09]. Die Lichtrichtung wurde dabei mit verschiedenen Mengen von virtuellen Licht-

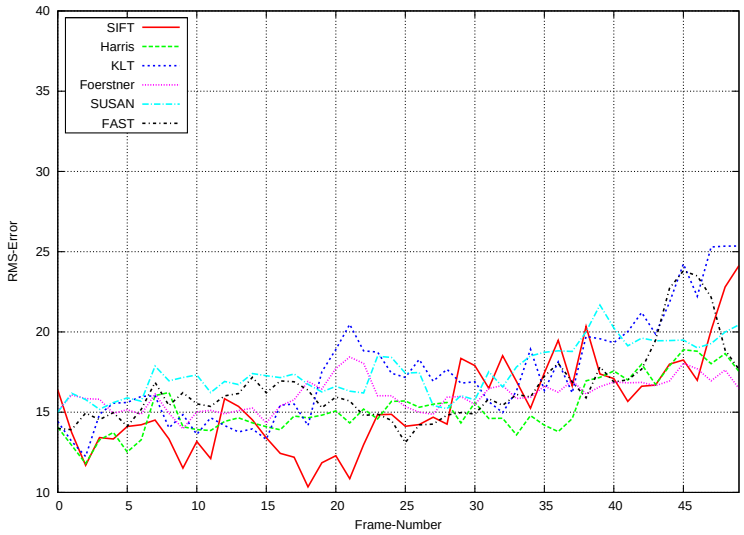


Abbildung 7: Fehler über die Bildsequenz, NCC Korrespondenzen

quellen gesampelt (Abbildung 8). Fehlerhafte Beleuchtung und Schatten führen klar zu schlechteren Ergebnissen, jedoch auch bei genauer Beleuchtungssimulation im synthetischen Bild ist die Beschreibung der Merkmale zwischen den Bildern oft zu unterschiedlich, um stabile Korrespondenzen zu erzeugen. Für die Detektion geeigneter Merkmale ist weiterhin eine Textur unabdingbar, deren Genauigkeit direkten Einfluss auf die gefundenen Merkmale hat, da Punkt- und Eckendetektoren auf der Texturinformation aufbauen. Lediglich der Foerstner-Operator zeigt keine nennenswerte Verbesserung bei Verwendung eines texturierten Modells. Ein ungenaues Modell führt insbesondere dann zu Rauschen der berechneten Pose oder gar zum Verlust des Trackings, wenn die Skalierung stark von der Realität abweicht, oder die Texturierung fehlerhaft ist.

3.1.2 Ähnlichkeitsbasierter Bildvergleich

Bei der zweiten Methode wird ausgehend von der letzten bekannten Pose eine Anzahl neuer Posen gestreut, um mögliche Bewegungen der Kamera

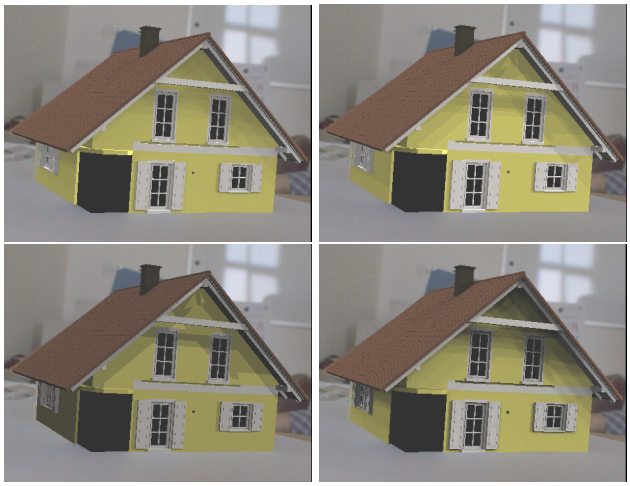


Abbildung 8: Verschiedene Lichtsimulationen (64,16,4 Lichtquellen. Schatten.)

zu simulieren. Dazu werden die Translations- und Rotationsparameter der Pose in festen Schritten oder nach Zufallswerten variiert. Von jeder der neuen Posen wird ein Bild gerendert und mit dem Kamerabild auf Ähnlichkeit verglichen. Der Einsatz von Merkmalen stellte sich auch hier aufgrund der bereits genannten Probleme als nicht erfolgversprechend heraus. Stattdessen wurde ein direkter Bildvergleich auf Basis der reinen Pixelintensitäten durchgeführt.

Da es nicht möglich ist, unendlich viele Bildvergleiche anzustellen, muss der Suchraum eingeschränkt werden. Eine Vereinfachung beim Posestreuen ist es, in jedem Optimierungsschritt $2n$ neue Posen um das letzte Bild mit der höchsten Ähnlichkeit zu generieren. Dabei ist n die Anzahl der Freiheitsgrade der zu optimierenden Parameter. Folglich werden sechs synthetische Bilder für die Translation entlang der drei positiven und negativen Koordinatenachsen gerendert. Weitere sechs für die Rotation um die Achsen und eines aus der aktuellen Kamerapose. Die Intervalle können entsprechend der Bewegungsgeschwindigkeit der Kamera adaptiv gewählt werden. Dabei zeigt eine generell hohe Ähnlichkeit aller Bilder eine langsame Bewegung

an und die Intervallschritte werden dementsprechend klein gewählt. Fällt die Ähnlichkeit unter einen Schwellwert, ist der Suchraum durch die Wahl einer größeren Schrittweite zu erhöhen. Nach dem Vergleich aller gerenderten Bilder mit dem Kamerabild, wird der Posevektor des synthetischen Bildes mit der höchsten Ähnlichkeit als Ausgangspunkt für den nächsten Iterationsschritt gewählt. Die initiale Pose wird auch hier als bekannt angenommen.

Statt Merkmale zu detektieren und Korrespondenzen zu suchen, wird hier ein Ähnlichkeitsmaß eingesetzt, das ohne Vorverarbeitung auf dem Bildinhalt operiert. Die Intensitäten der Pixel an übereinstimmenden Positionen in beiden Bildern können direkt paarweise verglichen werden. Mögliche Maße sind die Betrachtung der Summe der quadratischen Differenzen (SSD) oder die normalisierte Kreuzkorrelation (NCC) der beiden Bilder f und g:

$$d_{SSD}(f,g) = \frac{\sum_{x,y}(f(x,y) - g(x,y))^2}{\sqrt{\sum_{x,y} f(x,y)^2}\sqrt{\sum_{x,y} g(x,y)^2}}$$

$$d_{NCC}(f,g) = \frac{\sum_{x,y}(f(x,y) - f_\mu) * (g(x,y) - g_\mu)}{\sqrt{\sum_{x,y}(f(x,y) - f_\mu)^2}\sqrt{\sum_{x,y}(g(x,y) - g_\mu)^2}} \quad .$$

Weiterhin können die Bilder bezüglich ihres Informationsgehalts vor dem Vergleich abstrahiert werden, etwa durch nicht-photorealistische Rendertechniken oder eine Kantendetektion. Auch ist eine Histogrammanalyse denkbar. Zur Analyse der Ähnlichkeitsmaße wurde ein Referenzbild mit gerenderten Bildern eines Objekts verglichen. Das Objekt wurde beim Rendern in Schritten von 0,2 cm transliert und in Schritten von 0,5 Grad rotiert. Die Ergebniswerte des Ähnlichkeitsmaßes unter NCC ist in Abbildung 9 dargestellt. Die Kreuzkorrelation zeigt eine bessere Verteilung der Werte als der quadratische Fehler, was zu einer besseren Stabilität beim Bildvergleich

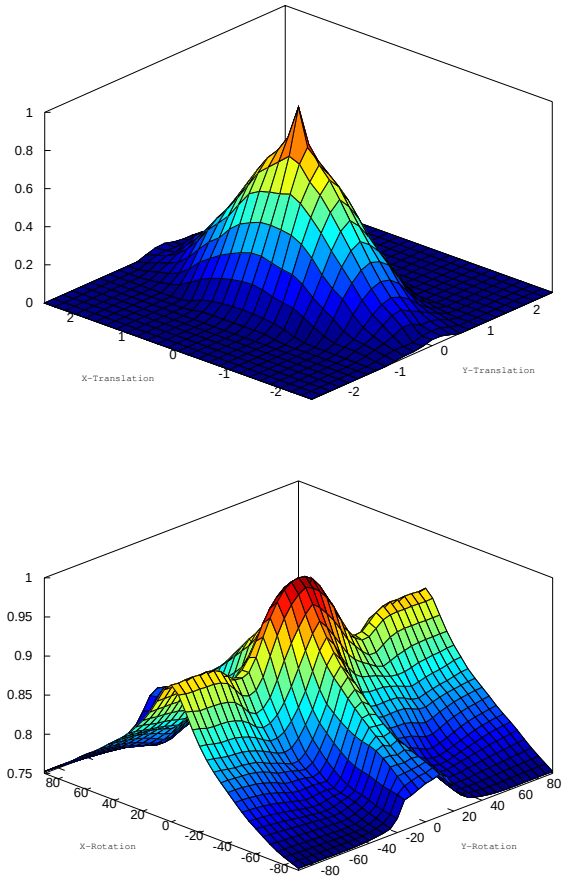


Abbildung 9: Verhalten der normalisierten Kreuzkorrelation unter Translation (oben) und Rotation (unten) eines Teapot-Objekts

führt und darüber hinaus invariant gegenüber Beleuchtungsänderungen im Bild ist.

Bei direktem Bildvergleich zeigte sich auch hier eine Abhängigkeit von der Korrektheit der virtuellen Lichtquelle, allerdings in geringerem Maße, als unter der Verwendung von Merkmalen. Histogramm-basierte Bildvergleiche führten generell zu schlechteren Ergebnissen, während ein Vergleich von vorverarbeiteten Kantenbildern die höchste Präzision erreicht. Da die Vorverarbeitung mit Kantenfiltern, wie etwa dem Sobel-Operator, nicht sehr zeitaufwändig ist und zusätzlich einen beleuchtungsinvarianten Ver-

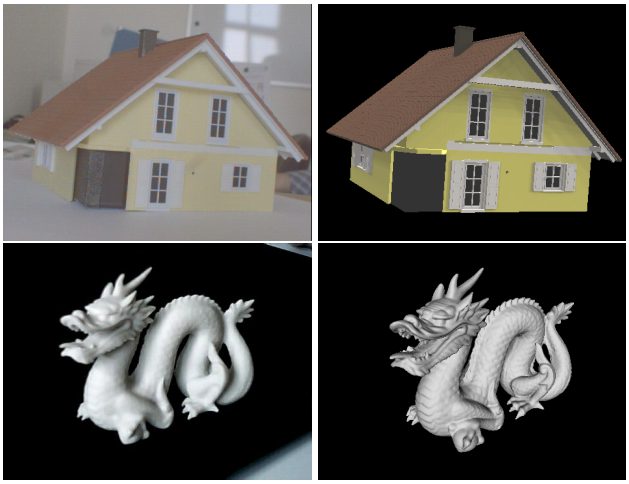


Abbildung 10: Reales (links) und synthetisches (rechts) Bild der Testobjekte

gleich ermöglicht, empfiehlt sich diese Vorgehensweise für die praktische Anwendung, etwa bei der Initialisierung.

Rendering und Vergleich von 640x480 Pixel großen Bildern benötigte etwa 7 ms einschließlich Texturtransfer auf der CPU. Ein kompletter Vergleich von 13 synthetischen Bildern arbeitet mit $\sim$ 3 FPS. Das Streuen vieler Posen führt aufgrund der damit verbundenen hohen Anzahl an Bildvergleichen zu einer starken Minderung der Performanz. Während bei der Verwendung eines abstrahierten Modells ein größerer Suchraum mit gesampelten Posen abgetastet werden muss und damit der Aufwand steigt, kann bei einem sehr detaillierten Modell die Anzahl der zu streuenden Posen verringert werden. Jedoch muss in diesem Fall die Geometrie recht exakt modelliert sein, da sich die Suche nach der optimalen Pose sonst schnell in einem lokalen Minimum verliert. Eine Einschränkung dieses Ansatzes besteht weiterhin in der Mehrdeutigkeit sich wiederholender, selbstähnlicher Strukturen, wie sie insbesondere bei Gebäudemodellen auftreten. Auch hier kann es zum Verlust des Trackings durch lokale Minima kommen.

3.2 Zusammenfassung

Mit der merkmalsbasierten und der ähnlichkeitsbasierten Methode wurden zwei Ansätze der *Analyse-durch-Synthese* im markerlosen Tracking untersucht. Die Ergebnisse zeigen, dass der Vergleich eines synthetischen Bildes mit dem Kamerabild (Abbildung 10) auch unter korrekt simulierter Beleuchtung nicht hinreichend genau ist, um durchgängig stabiles und präzises Tracking zu ermöglichen. Die Tests wurden mit den klassischen Verfahren der Bildverarbeitung zur Detektion und Deskription von Bildmerkmalen sowohl unter approximierter diffuser Beleuchtung als auch unter exakter Rekonstruktion der Lichtquelle mit einer High Dynamic Range Kamera durchgeführt. Zwischen einem gerenderten Bild und dem realen Kamerabild können nicht immer hinreichend gute Korrespondenzen für eine Fehlerminimierung erstellt werden, da die Deskriptoren der Merkmale für diesen Ansatz nicht optimal geeignet sind und auf beiden Bildern nur in geringer Zahl übereinstimmende Ergebnisse liefern. Für gute Trackingergebnisse muss die Beleuchtung annähernd fehlerfrei simuliert und eine möglichst genaue Texturierung vorhanden sein, was nicht in allen Fällen gewährleistet ist. Da jedoch durchweg zu weniger als einem Fünftel der detektierten Merkmale Korrespondenzen gefunden wurden, müssten die Fehlertoleranzen entsprechend höher gewählt werden, was zu mehr fehlerhaften Korrespondenzen führt und dementsprechend die Stabilität des Trackings reduziert. Schlussfolgernd lässt sich sagen, dass die Extraktion von Merkmalen aus einem gerenderten Bild nicht optimal geeignet ist, da nicht dieselben Merkmale wie im Kamerabild gefunden werden. Hingegen scheint der Ansatz des Posestreuens unter dem direkten Bildvergleich zumindest für eine nicht-kontinuierliche, grobe Bestimmung der Pose geeignet zu sein.

Als Folge wird für das weitere Vorgehen entschieden, die für das Tracking verwendeten Merkmale nicht aus dem synthetischen Bild zu extrahieren und über Pixeldeskriptoren zu beschreiben, sondern sie direkt aus der Geometrie des Modells zu gewinnen. Dadurch entfällt auch der doppelte Aufwand für Detektion und Deskription von Merkmalen auf zwei Bildern. In weiteren Untersuchungen sollten daher neue Möglichkeiten im Rahmen der *Analyse-durch-Synthese* entwickelt werden, um das Wissen über Modell, Umgebung und Eigenschaften von Licht und Perspektive für eine Vorhersage gut geeigneter Merkmale zu nutzen. Das Ziel ist die Generierung von wenigen, aber eindeutigen Merkmalen, die mit hoher Wahrscheinlichkeit im Kamerabild wiedergefunden werden können.

4 Initialisierung

In Anwendungen der Erweiterten Realität besteht das zentrale Problem im Bestimmen der Kamerapose, damit das Kamerabild lagerichtig mit visuellen Informationen ergänzt werden kann. Dazu wird mit entsprechenden Trackingverfahren eine kontinuierliche Berechnung der Kamerabewegung durchgeführt. Je nach eingesetztem Verfahren zur Poseschätzung müssen diese Trackingsysteme zunächst initialisiert werden, damit der kontinuierliche Trackingprozess gestartet werden kann. Es existieren zwei Gruppen von Algorithmen. Lineare Methoden berechnen eine globale lineare Lösung in einem Schritt. Sie sind schnell und benötigen keine vorherige Initialisierung. Die Präzision ist jedoch begrenzt, da sie sehr empfindlich auf falsche Korrespondenzen reagieren, was die Robustheit der Pose negativ beeinflusst.

Die zweite Gruppe bilden die nichtlinearen oder indirekten Verfahren. Die meisten von ihnen basieren auf klassischen iterativen Optimierungsalgorithmen wie dem Gauß-Newton Verfahren oder der Levenberg-Marquardt Methode [Lev44][Mar63] zur Approximation einer Funktion. Sie minimieren iterativ durch lokale Linearisierung die zwischen den Korrespondenzen von Bildmerkmalen gemessenen Distanzen, bis das Ergebnis in einem stabilen Distanzminimum konvergiert oder die maximal zulässige Anzahl an Iterationen erreicht ist. Diese Methoden sind sehr präzise und auch bei verrauschten Korrespondenzwerten sehr robust. Allerdings benötigen sie eine initiale Startpose, die bereits möglichst nah am gesuchten Optimum liegen muss. Andernfalls kann die Konvergenzrate sehr langsam werden oder die Methode konvergiert gegen ein lokales Minimum anstatt die global optimale Pose zu finden.

Die Initialisierung kann durch Benutzerinteraktion erfolgen, indem die initialen Korrespondenzen manuell gesetzt werden oder die virtuelle Ka-

mera von Hand der Pose des realen Kamerabildes angeglichen wird. Eine andere Möglichkeit ist die Verwendung von Markern, deren bekannte Position und Lage im Bild als Referenz dienen kann. Beide Praktiken sind jedoch mit zusätzlichem Arbeitsaufwand verbunden und das Anbringen von Markern ist nicht immer praktikabel oder erwünscht, besonders in einem weiträumigen, urbanen Szenario. Darüber hinaus existieren verschiedene Ansätze, welche GPS und Inertialsensoren, sowie bildbasierte Techniken in hybriden Trackingsystemen vereinen, jedoch meist merkmalsbasiert arbeiten. Die Initialisierung kann durch Inertial-GPS und Merkmalssuche auf planaren Ebenen durchgeführt werden [FON09], mit Hilfe von bekannten Keyframes in der Trackingsequenz [RD06] oder über die Vorhersage des GPS-Fehlers und Sampling, wie es [RD07] in ihrem Ansatz auf ähnliche Weise zu dem hier vorgestellten realisieren. Weiterhin wurden Methoden entwickelt um die GPS-Daten von mobilen Geräten mit einer Datenbank von georeferenzierten Bildern zu korrigieren [MABT11].

In dieser Arbeit wird ein zweischrittiges Konzept zur globalen Initialisierung eines modellbasierten Trackingsystems unter Anwendung der *Analyse-durch-Synthese* vorgestellt, welcher auf dem *8th International Symposium on Visual Computing* (ISVC) veröffentlicht wurde [SKM12]. Anhand einer Modelldatenbank, GPS-Koordinaten und der Orientierung eines elektronischen Kompass ist es möglich, den sichtbaren Teil einer Szene zu bestimmen und eine erste grobe Kamerapose zu erlangen. Des Weiteren wird ein Bildvergleichsverfahren zwischen Referenzbildern der Kamera und gerenderten Bildern der Modelle eingesetzt, um diese Pose weiter zu verfeinern und damit GPS-Ungenauigkeiten auszugleichen. Die Initialisierung ist dabei unabhängig von der genauen Arbeitsweise des zugrunde liegenden Trackingsystems. Der Ansatz wurde in einem urbanen Kontext getestet.

4.1 Modellbasiertes Initialisierungsschema

Im ersten Schritt der Initialisierung werden die Koordinaten des Global Positioning Systems (GPS) sowie die Orientierung durch einen elektronischen Kompass benötigt. Sie geben die Position der Kamera und die Blickrichtung der Kamera in [0,359] Grad bezogen auf die Nordrichtung an. Ausgehend von diesen Informationen soll die korrekte Transformation zwischen den Modellen und der realen Welt bestimmt werden. Auch sollen nur diejenigen Modelle geladen werden, welche auch wirklich vom Blickfeld der Kamera erfasst werden. Dazu ist eine Datenrepräsentation der Welt nötig, die eine Abfrage der vom Benutzer gesehenen Modelle erlaubt, sowie eine Koordinatentransformation von der GPS-Position und Weltorientierung in die lokalen Koordinaten des Trackingsystems. Die Bestimmung der Position durch GPS führt zu einer groben Positionsangabe, die eine Ungenauigkeit von mehreren Metern aufweisen kann. Dieses Problem wird im zweiten Teil der Initialisierung behandelt.

Zunächst muss eine adäquate Repräsentation der GPS-Position und eine entsprechende Koordinatentransformation definiert werden. Für die Abbildung der Modelldaten in die reale Welt soll das Universelle Transversale Mercator Koordinatensystem (UTM) dienen [Age89], das international zu Navigationszwecken eingesetzt wird. Der Vorteil von UTM ist, dass die Erdoberfläche in Zonen mit orthogonalem Koordinatensystem eingeteilt wird, was für die Abbildung wünschenswert ist, wenn im Trackingkontext mit einem virtuellen Kartesischen Koordinatensystem gearbeitet wird. UTM segmentiert die Oberfläche in 60 Zonen entlang des Äquators, wobei jede Zone 6 Längengrade umfasst. Orthogonal zum Äquator wird die Oberfläche in 20 Zonen mit jeweils 8 Breitengraden aufgeteilt. Innerhalb dieser Zonen ist die Distanz zwischen den Graden sehr gering, weshalb sie als

parallel angenommen werden können. Eine Einheit in UTM Koordinaten entspricht einem Meter in der Realität. Innerhalb einer Zone kann jeder Punkt über eine Kombination aus Rechts- und Hochwert beschrieben werden. Der Rechtswert definiert die Distanz des Punktes zum Mittelmeridian einer jeden Zone in Metern. Der Hochwert gibt die Distanz des Punktes zum Äquator an. Um bei diesem Vorgehen negative Koordinaten zu verhindern, werden die Rechtswerte westlich des Mittelmeridians mit 500.000 Metern inkrementiert (false easting) und zu den Hochwerten der südlichen Hemisphäre 10.000.000 Meter addiert (false northing). Die einzelnen Zonen werden global eindeutig adressiert (z.B. 32U Hauptzone Deutschland, siehe Abbildung 11).

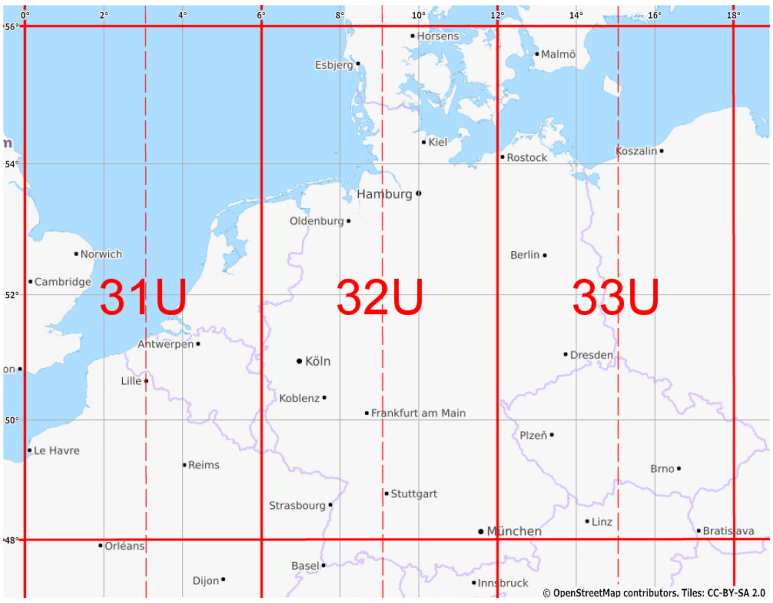


Abbildung 11: Die Einteilung des UTM-Koordinatensystems für Deutschland (Basierend auf freiem Kartenmaterial von openstreetmap.org).

Dank der Orthogonalität des UTM Koordinatensystems und der Skalierung in Metern besteht die Umwandlung von GPS-Koordinaten in das lokale Koordinatensystem einfach aus einer Translation und Rotation, wobei eine Einheit in den Koordinaten des Trackingsystems einem Meter in der

Realität entspricht. Um uns die Welt nun als Ansammlung von Modellen vorzustellen, gehen wir von einer Organisationsstruktur aus, die aus Bodenkacheln besteht. Modelle, die in einer räumlichen Beziehung stehen, werden auf einer gemeinsamen Grundplatte modelliert. Dies kann zum Beispiel eine Straße oder eine ganze Stadt sein. Die Betrachtung ganzer Bodenkacheln anstatt einzelner Modelle ist vorteilhaft, da die räumliche Anordnung zwischen allen Modellen einer Bodenkachel auf Grund des gemeinsamen Ursprungs konsistent bleibt. Außerdem sind topographische Informationen leichter zu handhaben und der Modellierungsprozess wird vereinfacht.

Zu Beginn der Initialisierung sind Position und Orientierung der bildgebenden Kamera durch GPS und Kompass bekannt. Zusätzliche Informationen, die vorab gemessen und gespeichert werden müssen, sind die GPS-Koordinaten des Ursprungs der Bodenkachel und ihre Orientierung R_T in Bezug auf die Nordrichtung. Dieser Parameter ermöglicht es, beliebige Kacheln zu definieren, die nicht kartografisch nach Norden ausgerichtet sind. Bei der Initialisierung wird der Ursprung der Bodenkachel gleichzeitig der Ursprung des lokalen Tracking Koordinatensystems. Die z-Achse des Trackingkoordinatensystems wird als Nord-Vektor definiert und die Bodenkachel dementsprechend gedreht. Das Modell wird mit den aus dem Modellierungsprozess bekannten Parametern R_M und T_M vom Ursprung aus rotiert und transliert. Ein Differenzvektor wird von den GPS-Koordinaten des Ursprungs der Bodenkachel zur GPS-Position der Kamera aufgespannt und beschreibt die Translation T_C der virtuellen Kamera. Der vom Kompass gelieferte Winkel in Bezug auf die Nordrichtung wird als Rotation R_C der virtuellen Kamera gesetzt (Abbildung 12).

Bis jetzt ist die planare Position und Orientierung des Modells in Bezug auf die Kamera bekannt. Die Höhe der Kamera wurde noch nicht berücksich-

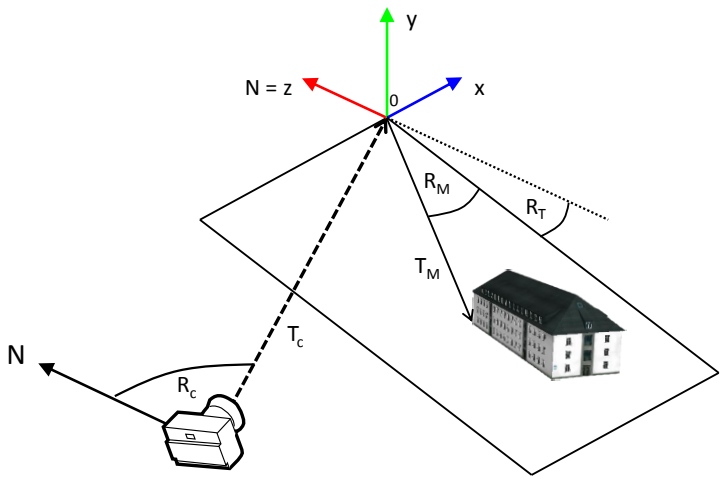


Abbildung 12: Die Transformationen von Kamera, Bodenkachel und Modell.

tigt. Sie könnte ebenfalls durch das GPS bestimmt werden, was jedoch nur mit großer Ungenauigkeit möglich ist. Im Idealfall ist für jede Bodenkachel auch das auf ihr abgebildete Terrain in Form eines Meshes verfügbar und zusammen mit den Modellen gespeichert. Das Terrain könnte beispielsweise aus einer Luftvermessung stammen (Abbildung 15). Wenn Informationen über die Geländebeschaffenheit in der Modelldatenbank hinterlegt sind, kann die Kamerahöhe wie folgt erlangt werden: An der bekannten Position der Kamera wird ein senkrechter Strahl aus einer festen Höhe auf das Terrain geschossen. Aus der Differenz zwischen dem Schnittpunkt mit der Geometrie und der Länge des Strahls bis zum tiefsten Geländepunkt der Kachel lässt sich der Höhenwert des Terrains an diesem Punkt ermitteln (Abbildung 13). Zu diesem Ergebnis wird dann noch die Körpergröße eines angenommenen Standardbenutzers (z.B. 1,80 Meter) addiert.

Nachdem die Transformation von Weltkoordinaten in die lokalen Trackingkoordinaten bekannt ist, müssen die sichtbaren Modelle bestimmt werden. Für jedes Modell werden daher in einem Abstand zwischen 10 und 20 Metern gleichmäßig Kontrollpunkte entlang der Gebäudeumrisse definiert und

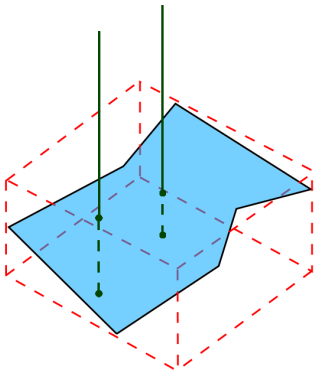


Abbildung 13: Betrachterhöhe durch Gelände-Strahl Schnitttest.

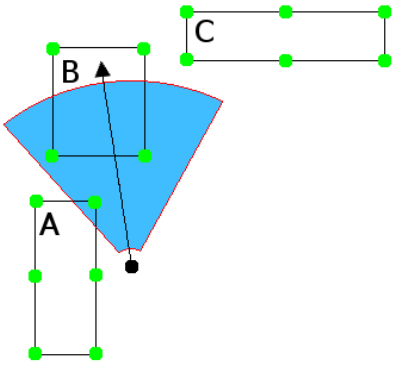


Abbildung 14: Sichtbarkeitstest anhand der Gebäude-Kontrollpunkte.

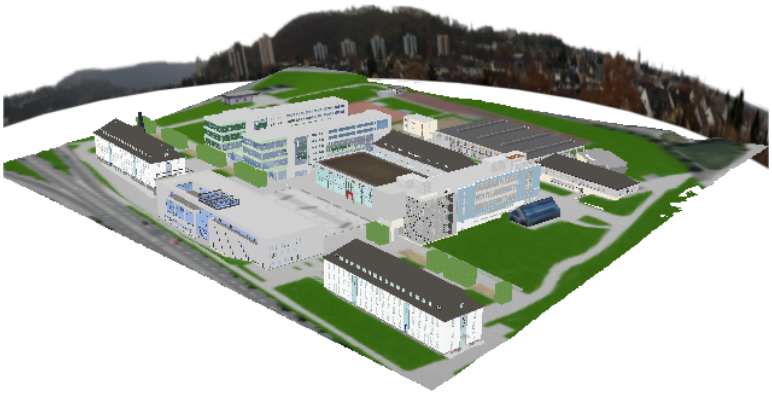


Abbildung 15: Terrain einer Bodenkachel.

ihre GPS-Koordinaten vermessen. So können zur Laufzeit diejenigen Modelle gefiltert werden, die unter Berücksichtigung der aktuellen Kamerapose sichtbar sind. Dieses dynamische Laden ermöglicht eine effiziente Handhabung beliebig großer Umgebungen. In großen Modelldatenbanken wäre es auch möglich, diese Kontrollpunkte automatisch aus der GPS-Position des Gebäudemittelpunktes und seinen Dimensionen berechnen zu lassen.

Abbildung 14 skizziert das Schema zum Testen der Kontrollpunkte gegen die Position und Orientierung der Kamera. Um die Kamera werden ein kleiner und ein großer Radius gezogen, die durch die minimale und

maximale Sichtweite definiert sind. Exemplarisch wird die minimale Distanz auf 0 Meter und die maximale Distanz auf 100 Meter gesetzt. Der Winkel des Sichtfeldes definiert den Suchraum, der durch den Vektor der Blickrichtung halbiert wird. Dieser Parameter wird durch den Öffnungswinkel der Kamera bestimmt. Ein Kontrollpunkt ist dann gültig, wenn er innerhalb der Grenzen des so definierten Frustums liegt. Im gezeigten Beispiel existieren drei gültige Kontrollpunkte von zwei Modellen (A und B), welche geladen werden müssen. Zunächst muss die Distanz zwischen Kamera und Kontrollpunkt innerhalb der Sichtweitenbegrenzung liegen. Dann wird überprüft, ob der Winkel zwischen dem Vektor von Kamera zu Kontrollpunkt und dem Blickrichtungsvektor der Kamera kleiner als der halbe Öffnungswinkel des Kamerafrustums ist. Nachdem alle Kontrollpunkte der Modelle auf der Kachel gegen die Kamerapose getestet wurden, wird der Quellpfad aller Modelle mit mindestens einem gültigen Kontrollpunkt an das Lademodul des Renderingsystems übergeben. Während der Ausführung des Trackingprozesses kann diese Methode dazu verwendet werden, dynamisch diejenigen Modelle zu laden, welche als nächste in das Sichtfeld des Benutzers eintreten.

Die Modelldatenbank in diesem Ansatz baut auf einem XML Schema auf, das einfach zu traversieren ist (Abbildung 16). Für jedes Modell werden in diesem Schema seine eindeutige ID, der Verzeichnispfad zum Laden des Modells, die Translations- und Rotationsparameter in Bezug auf den Ursprung der Bodenkachel und die Kontrollpunkte für den Sichtbarkeitstest gespeichert. Globale Informationen, die auf der Ebene der Bodenkachel verfügbar sein müssen, sind die GPS-Position des Ursprungs, die Orientierung der Kachel basierend auf der Nordrichtung, Geländeinformationen und eine Liste der Modelle, welche der Kachel zugeordnet sind. Gezeigt wird

hier der Ansatz mit nur einer Bodenkachel. Um *die Welt* mit Kacheln zu modellieren, können sie in einem Register verwaltet werden, in dem jede Kachel eine Stadt repräsentiert und mit ihrem Namen referenziert wird. Möglicherweise bietet sich auch eine feinere Unterteilung nach Straßennamen oder einer anderen passenden Semantik an. Diese Parameter können von einem kartenbasierten GPS ausgelesen werden, sodass die der Position des Benutzers entsprechende Kachel geladen werden kann und das System über die Kachel Zugriff auf alle relevanten Modelle erhält.

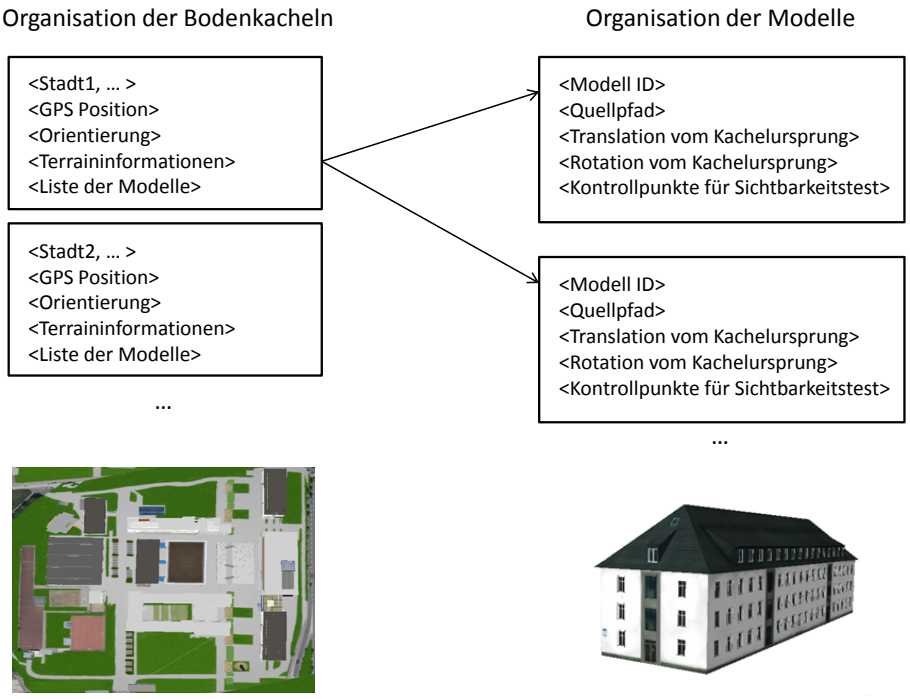


Abbildung 16: Datenschema zur Speicherung der Informationen von Bodenkachel und Modell.

Die mit diesem ersten Schritt gewonnene Initialpose kann sehr grob sein und ist deshalb oft um mehrere Meter zu ungenau um einen nichtlinearen Posealgorithmus zu starten. Dies ist bedingt durch die fehlende Präzision des zivilen GPS-Systems. Zusätzlich kann das Signal durch die Umgebung gestört sein. Außerdem wurde durch die Verwendung des Kompass bisher

nur die Blickrichtung als Rotation um die y-Achse der Kamera berücksichtigt. Es ist jedoch wahrscheinlich, dass der Betrachter seinen Kopf nach oben neigt, wenn er vor einem Gebäude steht. Diese Probleme können durch den *Analyse-durch-Synthese* Teil der Initialisierung gelöst werden.

4.2 Verfeinerung der Pose

Die erste grobe Kamerapose ist meistens noch nicht präzise genug um das Tracking zu starten. Durch die Ungenauigkeit des zivilen GPS-Signals und das Auftreten von Störungen in urbanen Umgebungen würde der Poseschätzer falsch initialisiert werden. Daher ist ein zusätzlicher Schritt zur Poseverfeinerung notwendig. Realisiert werden kann dies durch den Ansatz der *Analyse-durch-Synthese*. Die gesuchten Parameter sind in diesem Fall die Translation und Rotation der realen Kamera, deren Referenzbild mit einer Vielzahl von synthetischen Bildern verglichen wird. Die Poseparameter der virtuellen Kamera werden bei der Erzeugung jedes Bildes variiert, sodass das gerenderte Bild mit der höchsten Ähnlichkeit zum Kamerabild die gesuchten Parameter liefert.

Ausgehend von der ersten groben Pose durch GPS und Kompass, werden mehrere neue Posen für das Rendering erzeugt, indem eine Reihe von Translationen und Rotationen auf die virtuelle Kamera angewendet wird. Mit der groben Pose als Mittelpunkt wird ein Samplingbereich mit einer Kantenlänge von 24 Metern aufgespannt, was mit einer Länge von 12 Metern in jede Richtung der Standardungenauigkeit des GPS-Signals entspricht. Innerhalb des Bereiches wird die virtuelle Kamera in 2-Meter-Schritten nach links, rechts, vorne und hinten bewegt. Die Schrittweite kann abhängig von der Genauigkeit der groben Pose adaptiv gewählt werden. Die Höhe der Kamera wird bei jedem Schritt neu berechnet, wie im vorherigen Abschnitt

bereits beschrieben. Auf jede so generierte neue Pose werden wiederum Drehungen der Kamera um die x-Achse zwischen 0 und 15 Grad in 5-Grad-Schritten angewendet, um die natürliche Nickbewegung des Benutzers zu simulieren, wenn er vor einem Gebäude steht und den Blick hebt. Die horizontale Rotation des Kopfes in Blickrichtung ist durch den Kompass gegeben. Eventuelle Ungenauigkeiten werden durch eine spätere Unschärfefilterung des Referenzbildes ausgeglichen. Des Weiteren wird angenommen, dass der Benutzer den Kopf nicht schräg hält und die Kamera daher um die Blickachse waagerecht bleibt. Die so generierten Posen werden in einer Liste gespeichert und an den Renderer zur Erzeugung der synthetischen Bilder übergeben.

Nachdem die Vergleichsbilder gerendert wurden, muss deren Ähnlichkeit zum Referenzbild der Kamera bestimmt werden, wobei die prägnanten Bildstrukturen der Gebäude hilfreich sind. In einem urbanen Kontext finden sich viele starke Gebäudekanten, deren Gradienten durch Kantendetektoren berechnet werden können. Der Sobelfilter der OpenCV-Implementierung [BK08] mit einer Filtermaske der Größe 3x3 wird auf das Kamerabild und das gerenderte Bild angewendet. Die anschließende Berechnung der Gradientenstärke liefert je ein Intensitätsbild mit der Größe des Gradienten pro Pixel. Die Intensitätswerte I_c und I_r beider Bilder können nun direkt pixelweise verglichen werden. Je höher beide Intensitäten für ein Pixelpaar sind, desto stärker fließt es in den normalisierten Kantenintensitätswert I ein. Um zu verhindern, dass Pixel verglichen werden, die nicht zur selben Kante gehören, werden zusätzlich die Gradientenrichtungen D_c und D_r für jedes Pixel der beiden Bilder berechnet. Je kleiner die Differenz der Gradientenrichtungen in beiden Pixeln ist, desto höher ist der Beitrag des Produkts $I * D$ aus Gradientenstärke und Gradientenrichtungen zur aufsummierten

Bildähnlichkeit $S \in [0, 1]$. Wenn die Differenz der Gradientenrichtungen größer als 8 Grad sein sollte, wird D auf 0 gesetzt und das zu vergleichende Pixel dadurch nicht berücksichtigt. Die folgenden Schritte werden für alle Pixel jedes Bildpaars ausgeführt:

$$I = \frac{I_c}{255.0} * \frac{I_r}{255.0}$$

$$D = 1.0 - \frac{|D_c - D_r|}{8.0}$$

$$S+ = I * D.$$

Die Bildähnlichkeit wächst, je mehr Pixel mit ähnlicher Gradientenstärke und Gradientenrichtung gefunden werden. Nachdem alle gerenderten Bilder auf diese Weise mit dem Referenzbild der Kamera verglichen wurden, entspricht das Bild mit dem höchsten Wert für S der gesuchten Kamerapose. Um Toleranz gegenüber kleinen Kamerabewegungen oder unpräzisen Kompassdaten zu erlangen, wird das Kamerabild vor der Gradientenberechung mit einem Gauß-Filter geglättet. Dadurch werden die Bildkanten verbreitert, was den Pixelvergleich weniger empfindlich gegenüber leichten Verschiebungen macht. In Abbildung 17 ist links der Sobel-Gradient auf einem gerenderten Bild und rechts das entsprechende gaußgefilterte Kamerabild dargestellt. Das gerenderte Bild wurde vom vorgestellten Algorithmus als beste Korrespondenz zum Kamerabild ausgegeben. Abbildung 18 zeigt den Unterschied zwischen der ersten groben Poseschätzung basierend auf GPS und Kompassdaten, sowie das Ergebnis der Poseverfeinerung.

4.3 Ergebnisse

Der Ansatz der modellbasierten Initialisierung wird auf einem AMD Phenom II X4 965 (4 x 3,40 GHz) Prozessor und einer ATI Radeon HD 5850

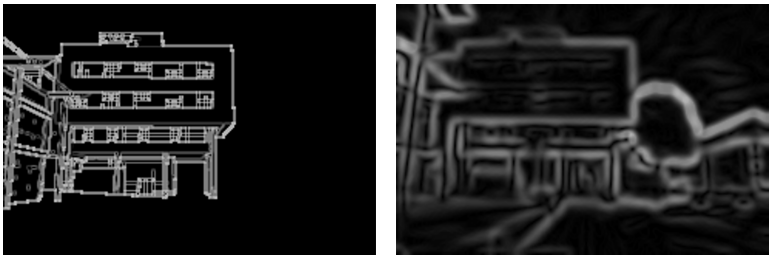


Abbildung 17: Sobelbilder für den Vergleich. Das gerenderte Bild (links) wurde als bestes Ergebnis zum gefilterten Kamerabild (rechts) zurückgegeben.

Abbildung 18: Links: Grobe Initialisierung. Mitte: Poseverfeinerung. Rechts: Kamerabild.

1 GB Grafikkarte getestet. Zum Test dient eine Reihe von Kamerabildern, deren Auflösung 968x648 Pixel beträgt. Um den Prozess des Bildvergleichs zu beschleunigen, werden die Bilder auf 242x162 Pixel herunterskaliert. Zu jedem Bild sind die GPS-Koordinaten und die Blickrichtung der Kamera bekannt, welche zum Zeitpunkt der Aufnahme von einem handelsüblichen Smartphone aufgezeichnet wurden. Anhand dieser Daten werden die sichtbaren Modelle und die erste grobe Kamerapose bestimmt. Die Rechenzeit für diesen Schritt ist vernachlässigbar. Anschließend werden 676 neue Kameraposen um die erste grobe Schätzung herum gestreut. Die virtuelle Kamera wird auf jede dieser Posen gesetzt und jeweils eine Ansicht des Modells gerendert, welche dann mit dem Kamerabild verglichen wird. Zum Rendern werden der virtuellen Kamera die intrinsischen Parameter Brennweite und Hauptpunkt der realen Kamera übergeben. Während der Poseverfeinerung konnten durchschnittlich 200 Bilder pro Sekunde verglichen werden. Dieser Wert ist abhängig von der Anzahl der sichtbaren Modelle. Werden vier

von ihnen angezeigt, sinkt die Vergleichsrate auf 150 Bilder pro Sekunde. Die gesamte Initialisierung benötigt 4-5 Sekunden ohne dass eine weitergehende Optimierung des Prozesses durchgeführt wurde. Die Rechenzeit könnte signifikant reduziert werden, wenn der Bildvergleich anhand eines shaderbasierten Bildfilters durchgeführt würde. Im Falle eines Trackingverlustes kann der Prozess jederzeit mit den aktuellen GPS-Koordinaten und Kompassdaten zur Reinitialisierung durchgeführt werden.

Abbildung 19 zeigt einige Ergebnisse der Initialisierung an verschiedenen Orten des Testszenarios. In der rechten Spalte sind die Kamerabilder zu sehen, links die Überlagerung mit der Ergebnispose. Die mit dem Ansatz erreichte Präzision ist hoch genug um einen nichtlinearen Poseschätzer erfolgreich zu initialisieren. Insgesamt wurde die Initialisierung an 21 Stellen im Testszenario mit verschiedenen Gebäuden durchgeführt und in über 75% der Durchläufe war keine nachträgliche Korrektur der Pose durch den Benutzer mehr nötig. In einigen Fällen weicht die Blickrichtung der Initialpose leicht von der durch den elektronischen Kompass ermittelten Richtung ab, da dieser empfindlich auf magnetische Störfelder reagiert.

Dieser Umstand kann durch die Unschärfefilterung des Referenzbildes korrigiert werden, was zu einer gewissen Toleranz gegenüber leichten Bewegungen der Kamera führt. Durchschnittlich ist eine Genauigkeit der initialen Pose von 2 bis 6 Metern ausreichend für eine erfolgreiche Initialisierung. Größeren Abweichungen der Blickrichtung kann durch die Berücksichtigung einer Rotation der virtuellen Kamera um die Hochachse während des Prozesses der Posestreuung begegnet werden. Dies würde jedoch zu mehr gerenderten Bildern führen und mehr Zeit für den Vergleich benötigen.

Darüber hinaus hat der Detailgrad der Modelle einen wesentlichen Einfluss auf die Vergleichbarkeit der Bilder. Je mehr signifikante Details wie

Abbildung 19: Ergebnisse der Initialisierung nach der Poseverfeinerung. Virtuelle Überlagerung (links) und Kamerabild (rechts).

Abbildung 20: Unzureichender Detailgrad in der Modellierung führt zu schlechterem Matching der Poseverfeinerung.

Fenster und Türen modelliert sind, desto größer ist die Präzision. Zu weit gewählte Samplingschritte, fehlerhaft skalierte oder fehlende Elemente führen zu einer falschen Auswahl der initialen Kamerapose. In Abbildung 20 ist die Abweichung von der Idealpose deutlich zu erkennen, was durch fehlende Fenster im Modell verursacht wird. Es sollte auch erwähnt werden, dass ein zu großer Fehler in den GPS-Koordinaten zum Scheitern der Initialisierung führt. Der Bildvergleich kann keine verfeinerte Pose liefern, wenn der Ausgangsfehler größer ist als der angenommene 24-Meter-Bereich, der von den gestreuten Posen abgedeckt wird. Dem kann durch die Erhöhung des Suchradius zur Streuung der Posen entgegengewirkt werden, verringert aber die Performanz.

4.4 Fazit

Die hohe Verfügbarkeit von GPS- und kompassfähigen Smartphones im Alltag, sowie die mobilen Online-Zugriffsmöglichkeiten auf schnell wachsende Modelldatenbanken wie *Google Earth* bereiten den Weg hin zur Realisierung allgegenwärtiger, urbaner Trackingszenarien (Abbildung 21). Vor diesem Hintergrund wurde die Möglichkeit gezeigt, ein modellbasiertes System zum Tracken der Kamerapose anhand von GPS- und Kompassdaten, sowie einer Modelldatenbank zu initialisieren. Die sichtbaren Modelle werden

Abbildung 21: *Google Earth:* Vision der Initialisierung anhand online verfügbarer Modellbibliotheken. Links: Kamerastandort. Rechts: Sichtbare Modelle.

dynamisch mit Hilfe von annotierten Referenzpunkten ausgewählt, welche gegen Position und Blickrichtung der Kamera getestet werden. Während die grobe Pose durch verrauschte GPS-Signale noch fehlerhaft sein kann, erwies sich der vorgeschlagene *Analyse-durch-Synthese* Ansatz zur Poseverfeinerung als präzise genug, um erfolgreich eine Startpose für den nichtlinearen Poseschätzer zu bestimmen.

Eine höhere Performanz des Initialisierungsprozesses könnte erreicht werden, indem während der Poseverfeinerung ein iterativer Samplingansatz zum Streuen der Kameraposen verwendet wird. Eine bessere Suchstrategie würde zu einer geringeren Anzahl an zu vergleichenden Bildern führen. Weiterhin wird die Poseverfeinerung bisher auf der CPU ausgeführt. Der Bildvergleich könnte jedoch auch mit Hilfe einer Shaderimplementierung auf die GPU verlagert werden. Sollte das Tracking auf einem mobilen Gerät zum Einsatz kommen, kann der Initialisierungsprozess von einem Server übernommen werden, welcher GPS- und Kompassdaten zusammen mit dem Kamerabild empfängt und die berechnete Initialpose an den Client zurückgibt.

5 Pose

Das Problem der Bestimmung von Lage und Orientierung (Pose) einer Kamera im Raum besteht in der Minimierung der Distanzen zwischen Merkmalen in mehreren von der Kamera aufgenommenen Bildern. Anhand eines gegebenen Modells ist es auch möglich, diese Korrespondenzen zwischen projizierten 3D Modellmerkmalen und ihren entsprechenden 2D Bildmerkmalen zu erzeugen. Zur Bestimmung der Kamerapose können zwei Klassen von Algorithmen herangezogen werden: Die linearen Verfahren berechnen die gesuchten Parameter direkt durch Lösung linearer Gleichungssysteme, ohne zusätzliche Informationen neben den Korrespondenzen heranzuziehen. Sie benötigen keine Initialisierung und die Komplexität ist gering, doch ist die Lösung weniger exakt und Rauschen sowie fehlerhafte Korrespondenzen können zu falschen Ergebnissen führen. Ein Überblick findet sich im Anhang (Kapitel 9), eine vertiefende Analyse hierzu wurde in [Gai11] durchgeführt.

Die nichtlinearen Verfahren hingegen optimieren die initial übergebene Pose durch lokale Linearisierung, was durch Approximation zu einer exakteren Pose führt. Dieser Prozess wird iterativ wiederholt, bis eine maximale Anzahl an Iterationen erreicht ist oder das Ergebnis konvergiert. Die meisten in der Praxis eingesetzten Algorithmen stammen aus dem Bereich der linearen Verfahren und arbeiten auf Punkt- oder Kantenmerkmalen. Für die verschiedenen Merkmalstypen ist jedoch keine vollständige Analyse der Parameter unter der nichtlinearen Poseschätzung verfügbar. Daher sollen die Parameter für Punkte, Kanten und deren Kombination analysiert werden.

Die zu klärende Frage betrifft die Wahl des Optimierungsschemas zur Posebestimmung hinsichtlich der Konvergenzgeschwindigkeit, des benötigten initialen Startwertes und des Abbruchkriteriums. Es werden mögliche Para-

metrisierungen der Kamerapose, ihre zu erfüllenden Nebenbedingungen, sowie mehrere Fehlermaße für die im Optimierungsprozess zu minimierenden Korrespondenzen verglichen. Es gilt, diejenige Parameterkombination zu finden, welche die präziseste Optimierung der Pose liefert. Zur Diskussion stehen als mögliche Abstandsmaße der Rückprojektionsfehler, welcher die Distanz zwischen Modell- und Bildmerkmal nach der Projektion in Pixelkoordinaten angibt, und der Objektraumfehler, der die Distanz im Kamerakoordinatensystem bestimmt. Bei Kantenmerkmalen kann der Fehler zusätzlich im Hough-Parameterraum angegeben werden. Die Parametrisierung der Rotation betreffend, muss eine geeignete Repräsentation gefunden und festgestellt werden, ob sie die Eigenschaften einer gültigen Rotation sicherstellt, oder ob diese durch zusätzlichen Aufwand, wie die Berechnung einer Singulärwertzerlegung, erzwungen werden müssen. Mögliche Parametrisierungen der Rotationsparameter sind eine Rotationsmatrix, die Euler-Winkel-Darstellung, Quaternionen oder die Rodrigues-Formel. Des Weiteren ist die Frage zu beantworten, welche Art von Merkmalen einzusetzen ist, also ob die Verwendung von Punkten oder Kanten zum Tracking erfolgversprechender ist, oder ob sich etwa eine Kombination aus beidem als sinnvoll erweist.

Weiterhin werden kritische geometrische Konfigurationen in den Korrespondenzen aufgezeigt, die zu mehrdeutigen Ergebnissen führen können. Das Wissen über das 3D Modell kann genutzt werden, um die Eingabedaten auf diese Konfigurationen zu untersuchen und diejenigen Korrespondenzen auszuwählen, die zu einer stabilen Lösung führen. Anhand dieser Ergebnisse wird die beste Parameterwahl vorgeschlagen, welche für eine hinsichtlich Typ und Anzahl beliebig kombinierte Menge an Merkmalen ein optimales Ergebnis erzielt. Das Verhalten der Merkmalstypen unter verschiedenen

Kombinationen und unterschiedlichen Mengen, sowie der Einfluss von Rauschen auf die Robustheit der Pose wird untersucht.

Die Ergebnisse wurden unter dem Titel „Parameter and Configuration Analysis for Non-Linear Pose Estimation with Points and Lines“ auf der *7th International Conference on Computer Vision Theory and Applications (VISAPP)* vorgestellt [RSM12b] und als Arbeitsbericht veröffentlicht [RSM12a]. Eine detailliertere Darstellung findet sich in [Rei11].

5.1 Verwandte Arbeiten

Das Problem der Kameraposebestimmung mit 2D-3D Punktkorrespondenzen ist als das Perspective-n-Point Problem (PnP) bekannt [FB81] und basiert auf einem System aus linearen Gleichungen. Mehrere Ansätze zur Lösung des P3P Problems wurden in [HLON94] untersucht und auf ihre numerische Stabilität hin verglichen. Während diese linearen Lösungen auf einer festen Anzahl von drei Punktkorrespondenzen arbeiten, verwenden [LMNF09] virtuelle Kontrollpunkte, um eine Lösung für eine beliebige Anzahl an Korrespondenzen mit linearer Komplexität zu ermöglichen. Ein anderer linearer Ansatz für beliebige Mengen an Punkt- oder Kantenkorrespondenzen wird von [AD03] beschrieben, arbeitet jedoch nicht auf beiden Merkmalstypen gleichzeitig. Aus der Disziplin der *Struktur aus Bewegung* stammt eine lineare Lösung mit Punkten, Linien und Ebenen unter einem Multiview-System [Rot03].

Alternativ kann die Pose auch mit nichtlinearen, indirekten Verfahren berechnet werden. Die meisten von ihnen basieren auf klassischen iterativen Optimierungsalgorithmen, wie dem Gauß-Newton oder dem Levenberg-Marquardt Verfahren. Nichtlineare Poseberechnung mit Kantenkorrespondenzen wurde von [KH94] durchgeführt, welche den Einfluss der gewählten

Kantenrepräsentation auf den Optimierungsprozess untersuchten. Eine Lösung für beliebige Kantenzüge eines Objekts wurde von [Low91] gezeigt. Mögliche Fehlermaße für Punktkorrespondenzen wurden von [LHM00] analysiert, die auch Ansätze zur Verringerung der benötigten Iterationen bei der Optimierung vorschlagen. Mit der Erweiterung des bekannten POSIT Algorithmus schlagen [DG99] eine Lösung zur Kombination von Punkten und Kanten vor. Ein Ansatz zur gleichzeitigen Schätzung von 3D Struktur und Kamerabewegnung aus mehreren 2D Ansichten mit Punkten, Kanten und Ebenen unter der Verwendung eines Erweiterten Kalman-Filters wird von [DST08] vorgeschlagen.

In der vorhandenen Literatur wurden bisweilen auf dem Gebiet der linearen Poseberechnung Ansätze zur Kombination von Punkt- und Kantenmerkmalen vorgestellt. Die meisten Autoren verzichten jedoch auf eine umfassende Analyse der Parameter. Daher wird in den Abschnitten 5.5, 5.6 und 5.7 eine vollständige Analyse von Korrespondenz-Fehlermaßen und Parametrisierungen der Kamerapose unternommen, um die beste Auswahl dieser Parameter zur Konstruktion eines nichtlinearen Poseschätzers mit Punkten und Kanten zu finden.

Ein Überblick kritischer geometrischer Konfigurationen für Punke ist in [FB81] zu finden. Die Anzahl möglicher Lösungen bei der Poseberechnung wird im Falle von P3P durch [WMSM91] sowie für P4P von [HW02] und [GT06] bewiesen. Ein Beispiel für kritische Kantenkonfigurationen findet sich in [CH99]. Diese Arbeit widmet sich dem Problem der Erkennung solcher Konfigurationen aus einer Menge von kombinierten Punkt- und Kantenkorrespondenzen in Abschnitt 5.3.

5.2 Definitionen

Der Begriff *Pose* beschreibt die relative Position und Orientierung eines Objekts in der Welt, in Bezug auf den Ursprung eines Referenzkoordinatensystems. Durch *Tracking* wird im Allgemeinen die Pose eines Objekts erfasst, indem die Transformation zwischen dem Koordinatensystem des Objekts und dem Koordinatensystem der Welt bestimmt wird. Daneben bezeichnet Tracking in der Bildverarbeitung das *Verfolgen* der Bewegung von Objekten oder Personen in einem Kamerabild, das sogenannte Motion Tracking. Im vorliegenden Fall des Kameratrackings soll eine Transformation zwischen dem Welt- und dem Kamerakoordinatensystem gefunden werden, welche zu einem gegebenen 3D Objekt und seiner 2D Abbildung, die Rotation und Translation der abbildenden Kamera im 3D Raum liefert. Gesucht ist also der Standort der Kamera und ihre Ausrichtung zum Zeitpunkt der Aufnahme: Wo befinde ich mich? Wohin schaue ich?

Dazu müssen die von der Kamera gelieferten Daten mit denen der Welt abgeglichen werden. Dieser Vorgang wird in der Disziplin des Rechnersehens (Computer Vision) als *Registrierung* bezeichnet. Eine 2D-2D Registrierung findet statt, wenn 2D Bilder eines Objekts, die zu verschiedenen Zeiten, aus verschiedenen Perspektiven oder auch mit unterschiedlicher Modalität aufgenommen wurden, durch entsprechende euklidische oder affine Transformation in einem Koordinatensystem in Deckung gebracht werden. Dies wird angewandt, um aus einer Bildsequenz die Kamerapose zu berechnen und ist darüber hinaus in den Gebieten der Kartographie von Nutzen, um beispielsweise Satellitenbilder und Panoramabilder zusammenzusetzen oder in der medizinischen Bildgebung, um Bilder verschiedener Informationsquellen zu vergleichen oder zu fusionieren. Die Registrierung kann intensitätsbasiert erfolgen, indem die Intensitätsverteilung der Bilder

mit Korrelationsmetriken miteinander verglichen und so die Ähnlichkeit der Bilder bestimmt wird. Bei der merkmalsbasierten Registrierung wird die Transformation zwischen den Bildern über die Minimierung der Abstände von korrespondierenden Merkmalen bestimmt. Des Weiteren kann die Registrierung über den Frequenzraum erfolgen.

Unter projektiver Registrierung versteht man das Abbilden von bekannten 3D Daten auf 2D Daten durch euklidische oder affine Transformation und anschließender projektiver Transformation. Dies ist bei der Abbildung von Weltpunkten durch die Kamera und die folgende Projektion in das Bild der Fall und findet Anwendung beim modellgestützten Kameratracking.

5.2.1 Kamera

Das einer Abbildung durch eine Kamera zugrunde liegende Kameramodell wird durch eine Reihe Parameter beschrieben, die sich in *extrinsische* oder externe und *intrinsische* oder interne Kameraparameter unterscheiden lassen. Die externen Parameter beschreiben die Transformation von dem Welt- in das Kamerakoordinatensystem, welche sich aus einer 3x3 Rotationsmatrix R und einem Translationsvektor $\vec{t^c}$ zusammensetzt (Abbildung 22). Es handelt sich dabei um die gesuchten 6 Parameter der Pose.

Objektpunkte $\vec{p^w} = (x, y, z)$ in einem beliebigen Weltkoordinatensystem mit den Koordinatenachsen x, y und z werden durch eine Rotation R und eine Translation $\vec{t^c}$ als Punkte $\vec{p^c} = (i, j, k)$ in das Kamerakoordinatensystem transformiert:

$$\vec{p^c} = R\vec{p^w} + \vec{t^c}$$

$$\text{mit} \quad R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad \text{und} \quad \vec{t^c} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}.$$

Der Ursprung des Kamerakoordinatensystems liegt im Projektionszentrum **O** der Kamera, die orthogonalen Achsen i und j definieren eine Ebene parallel zur Bildebene und k zeigt als optische Achse vom Projektionszentrum in die Welt.

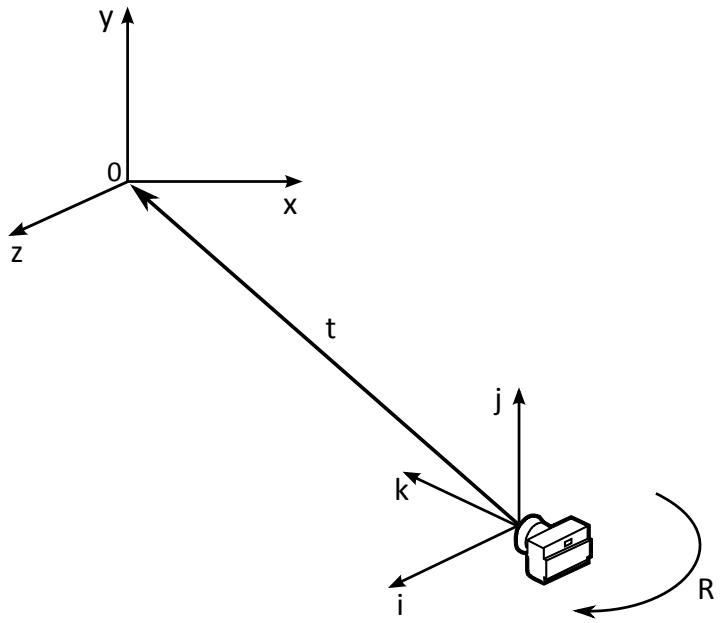


Abbildung 22: Extrinsische Transformation

Die Rotation kann dabei als Projektion des Weltpunktes auf die Koordinatenachsen des Kamerasystems ausgedrückt werden, indem die Achsen des Kamerakoordinatensystems als Zeilen in die Rotationsmatrix geschrieben werden:

$$R = \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \\ k_x & k_y & k_z \end{pmatrix}.$$

Die Transformation

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

lässt sich dann in homogenen Koordinaten als kombinierte Matrix ausdrücken:

$$\begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Die internen Kameraparameter beschreiben unabhängig von der Bewegung der Kamera in der Welt die Abbildung der Objektpunkte aus dem Kamerakoordinatensystem in die Bildebene. Sie sind für jede Kamera fest definiert. Die einfache perspektivische Projektion der Objektpunkte in die Bildebene wird durch den Strahlensatz definiert. Ausgegangen wird von einem Lochkameramodell in Positivlage, das heißt die Bildebene befindet sich vor dem Projektionszentrum (Abbildung 23).

Ein Objektpunkt $\vec{p^c}$ in Kamerakoordinaten wird unter Annahme des Abstandes f der Bildebene vom Projektionszentrum der Kamera, welcher auch als Brennweite oder Kamerakonstante bezeichnet wird, auf den Bildpunkt $\vec{p^i} = (u, v)$ im zweidimensionalen Koordinatensystem der Bildebene abgebildet. Gemäß Strahlensatz gilt

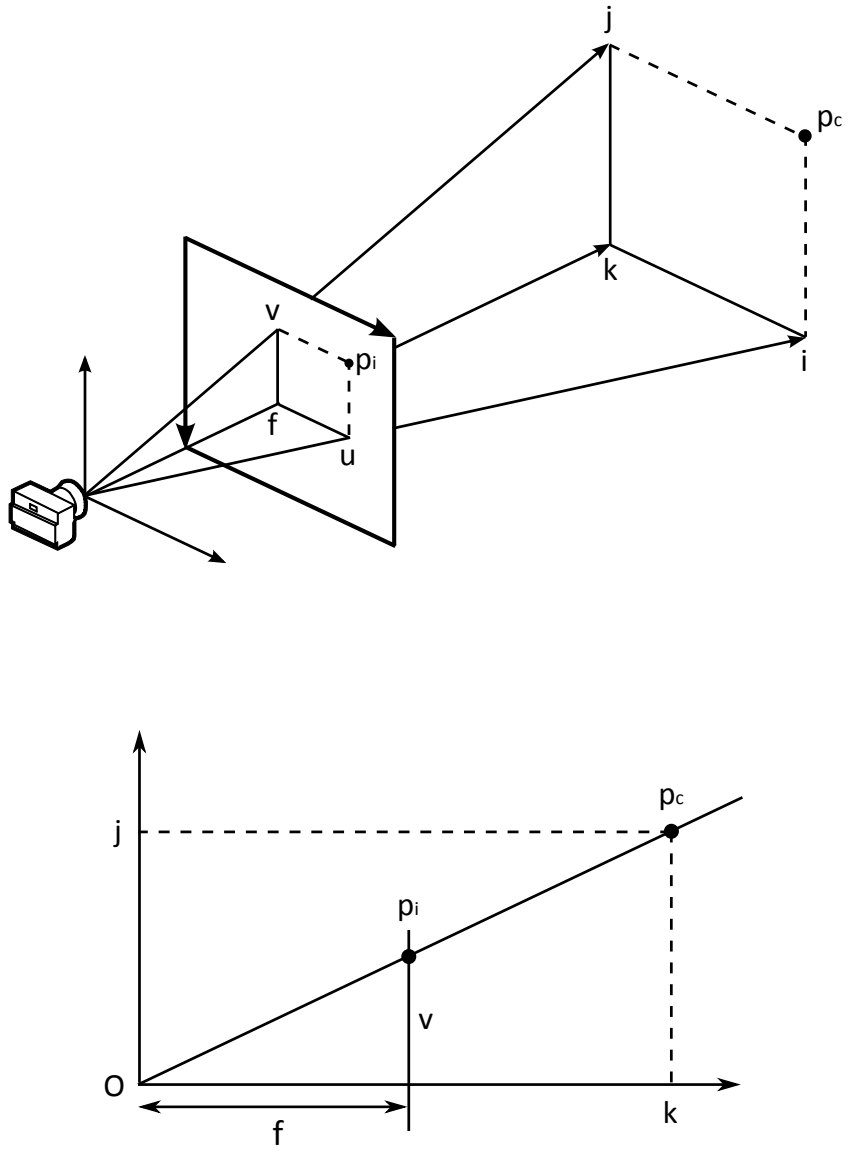


Abbildung 23: Projektion

$$\frac{u}{f} = \frac{i}{k} \Rightarrow u = f\frac{i}{k}$$

$$\frac{v}{f} = \frac{j}{k} \Rightarrow v = f\frac{j}{k}.$$

Mit der daraus folgenden Kameramatrix

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

lässt sich die Projektion darstellen als

$$\vec{p^i} = K\vec{p^c}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\frac{i}{k} \\ f\frac{j}{k} \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix}.$$

Die bisherige Abbildung geht von idealen Bildkoordinaten ohne diskrete Pixel (*Pi*cture *El*ement) aus. Der Bildhauptpunkt, der Schnittpunkt der optischen Achse mit der Bildebene, wird in der Mitte der Bildebene angenommen. Für eine realistische Abbildung müssen weitere interne Parameter eingeführt werden, um die Bildpunkte $\vec{p^i}$ in Pixelkoordinaten $\vec{p^p}$ zu überführen. Der Ursprung des Pixelkoordinatensystems liegt im Allgemeinen in der linken oberen Ecke, wobei die Achse u horizontal nach rechts zeigt und die Achse v senkrecht nach unten, weshalb der Eintrag für f_y zu negieren ist. Der tatsächliche Hauptpunkt der Bildebene wird mit den Koordinaten H_x und H_y angegeben. Um nicht-quadratische Pixel zu berücksichtigen, geht die Breite d_x und Höhe d_y der Pixel des CCD-Chips in die Brennweite mit ein, als $f_x = \frac{f}{d_x}$ und $f_y = \frac{f}{d_y}$. Der Faktor s beinhaltet den

Winkel zwischen den Bildkoordinatenachsen u und v, für dem Fall, dass diese nicht orthogonal ausgerichtet sind. Er ist jedoch in der Regel Null. Mit der erweiterten Kamerakalibriermatrix

$$K = \begin{pmatrix} f_x & s & H_x \\ 0 & -f_y & H_y \\ 0 & 0 & 1 \end{pmatrix}$$

ist die vollständige Abbildung vom Weltkoordinatensystem in das Pixelkoordinatensystem damit

$$\vec{p^p} = K(R|\vec{t^c})\vec{p^w}.$$

Zu den bereits betrachteten 5 Parametern kommen zwei weitere Parameter κ_1 und κ_2 hinzu, sofern eine radiale Verzerrung durch die Kameralinse behandelt werden soll (Abbildung 24). Tangentiale Verzerrungen sind in der Regel nicht relevant. Die Linsenverzerrung wird näherungsweise als Abstand r eines Punktes in unverzerrten Koordinaten vom Ursprung des Bildkoordinatensystems dargestellt. Die Beziehung von idealen Bildkoordinaten p^i und verzerrten Bildkoordinaten p'^i ist

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4) \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\text{mit} \quad r = \sqrt{u^2 + v^2}.$$

Die internen Kameraparameter werden durch vorherige *Kamerakalibrierung* bestimmt und für die Berechnung der Pose als bekannt angenommen. Dadurch reduziert sich die Suche auf die 6 externen Parameter.

Zur Berechnung der vorgestellten Parameter existiert eine Vielzahl von

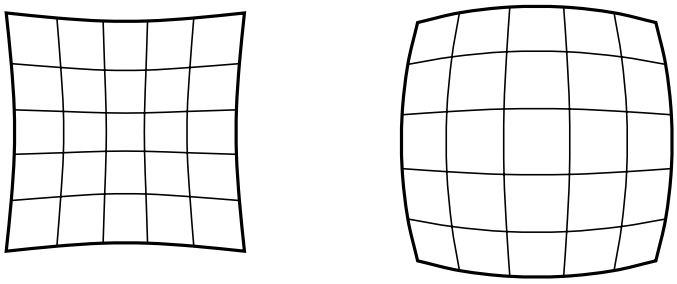


Abbildung 24: Links: Kissenverzerrung, rechts: Tonnenverzerrung

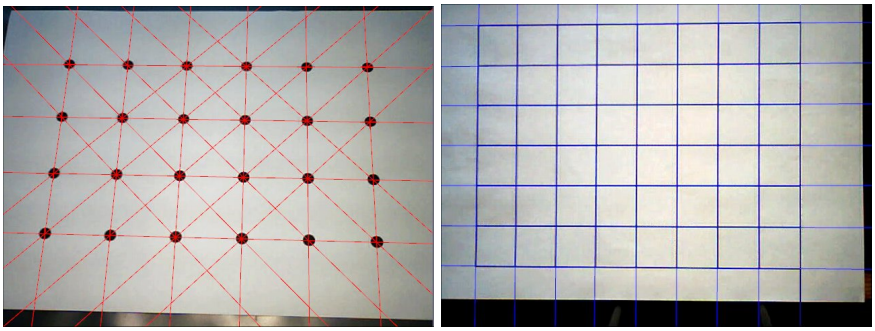


Abbildung 25: Kalibriermuster

Algorithmen, die aus den Korrespondenzen zwischen einer Menge bekannter Weltpunkte $\{p^{w_1}, ..., p^{w_n}\}$ mit $p^{w_n} \in \mathbb{R}^3$ und der entsprechenden Menge ihrer Bildpunkte $\{p^{p_1}, ..., p^{p_n}\}$ mit $p^{p_n} \in \mathbb{R}^2$ die gesuchten Parameter herleiten. Die Weltpunkte sind meist in Form eines vermessenen Kalibriermusters gegeben (Abbildung 25). Die entsprechenden Bildpunkte ihrer Abbildung werden im Kamerabild mit Verfahren der Merkmalsextraktion detektiert. Die Parameterbestimmung kann neben Punkten auch auf Linienkorrespondenzen erfolgen. Im Falle der *Analyse-durch-Synthese* wird auf rückprojizierten 3D Weltpunkten oder Linien und ihren korrespondierenden Entsprechungen im Kamerabild operiert.

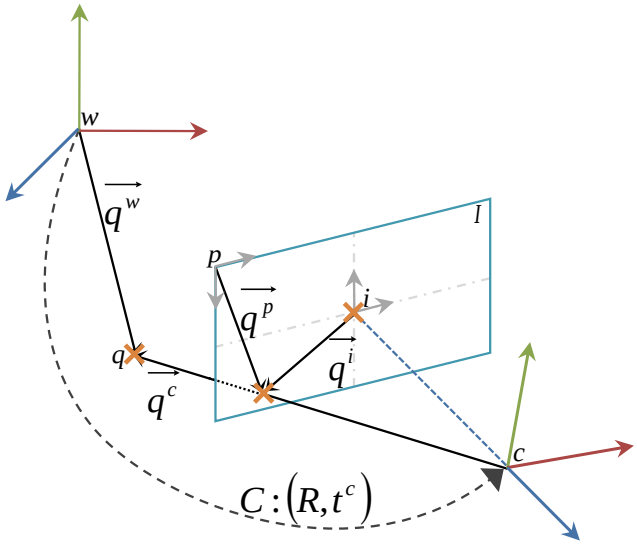


Abbildung 26: Das Poseproblem und die beteiligten Koordinatensysteme für einen Weltpunkt q.

5.2.2 Merkmalskorrespondenzen

Das Poseproblem wird als Berechnung der extrinsischen Kameraparameter, relativ zu einem bekannten Referenzkoordinatensystem - im Allgemeinen handelt es sich um das Weltkoordinatensystem - aus den Korrespondenzen

zwischen 3D Merkmalen des synthetischen Modells und 2D Merkmalen im Bildkoordinatensystem beschrieben werden. Die Bezugskoordinatensysteme der Merkmale werden durch einen Index w für Welt-, c für Kamera-, i für Bild- und p für Pixelkoordinaten gekennzeichnet (Abbildung 26).

Die Normalisierung eines Vektors $\vec{p}$ wird mit $\hat{\vec{p}}$ angegeben. Die Repräsentation der Kamerapose erfolgt durch eine Rotationsmatrix $R \in \mathbb{R}^{3x3}$ und einen Translationsvektor $\vec{t^c} \in \mathbb{R}^3$ als $C : (R, t^c)$ und bezeichnet, wie im vorherigen Abschnitt beschrieben, die Transformation eines Punktes p zwischen Welt- und Kamerakoordinatensystem als $\vec{p^c} = R\vec{p^w} + \vec{t^c}$. Die intrinsische Kameramatrix K wird durch vorhergehende Kalibrierung als bekannt angenommen. Folglich können aus einem gegebenen Weltpunkt $\vec{p^w}$ der Kameramatrix K, die zugehörigen Pixelkoordinaten $\vec{p^p}$ durch perspektivische Projektion berechnet werden:

$$\vec{p^p} = \begin{pmatrix} f_x \frac{p_x^w r_{11} + p_y^w r_{12} + p_z^w r_{13} + t_x^c}{p_x^w r_{31} + p_y^w r_{32} + p_z^w r_{33} + t_z^c} + h_x \\ h_y - f_y \frac{p_x^w r_{21} + p_y^w r_{22} + p_z^w r_{23} + t_y^c}{p_x^w r_{31} + p_y^w r_{32} + p_z^w r_{33} + t_z^c} \end{pmatrix}.$$

Eine *Punktkorrespondenz* aus Modellpunkt $\vec{p^w}$ und dem Bildpunkt $\vec{q^p}$ wird als Tupel $k_p : \left(\vec{p^w}, \vec{q^p}\right)$ dargestellt.

Die Kantenmerkmale im Kamerabild sind als unendliche Geraden $l : (\phi^p, \rho^p)$ definiert. Dabei bezeichnet ϕ^p den Winkel zwischen der Normalen auf der Geraden und der y-Achse des Pixelkoordinatensystems. Der Parameter ρ^p ist die orthogonale Distanz der Geraden vom Bildursprung (Abbildung 27). Da die Normale der Geraden als

$$\vec{n^p} = \begin{pmatrix} \cos\phi^p \\ \sin\phi^p \end{pmatrix}$$

definiert ist, gilt für jeden Punkt $\vec{p^p} \in l$

$$\cos\phi^p p_x^p + \sin\phi^p p_y^p = \rho^p.$$

Eine *Geraden- oder Kantenkorrespondenz* zwischen einer Modellkante, die aus zwei Weltpunkten $s^{\vec{w}}$ und $e^{\vec{w}}$ (typischerweise Start- und Endpunkt) bezeichnet wird, und einer Bildgeraden $l^p : (\phi^p, \rho^p)$, lässt sich durch ein Tupel $k_l : \left((s^{\vec{w}}, e^{\vec{w}}), l\right)$ repräsentieren.

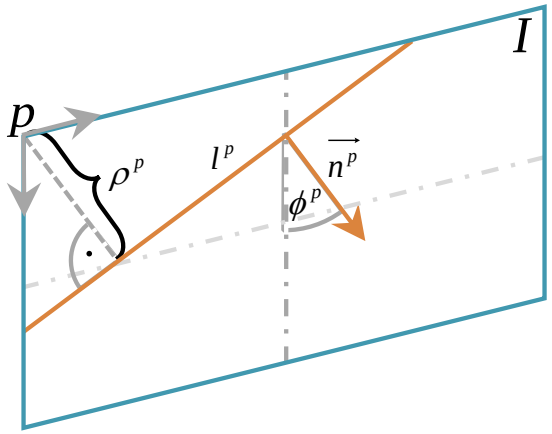


Abbildung 27: Geradenparametrisierung

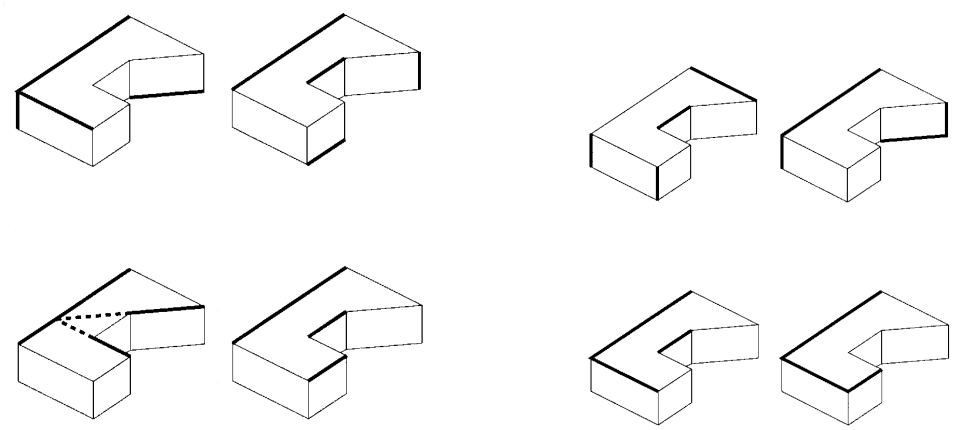
5.3 Geometrische Konfiguration

Es wurde bewiesen, dass ein Auftreten von Konfigurationen, welche zu mehrdeutigen Lösungen führen, unwahrscheinlich ist [WMSM91][GT06]. Dennoch ist es gerade bei der Poseschätzung mit einer kleinen Anzahl an Korrespondenzen empfehlenswert, die geometrischen Eigenschaften der Korrespondenzmenge zu prüfen und sowohl kritische (mehrdeutige) als auch degenerierte (linear abhängige) Konfigurationen zu vermeiden. Bei Verwendung eines 3D Modells können die geometrischen Beziehungen zwischen den Korrespondenzen einfach anhand der bekannten Daten festgestellt werden.

Von größter Bedeutung ist die Anzahl der benötigten Korrespondenzen, um eine eindeutige Lösung bei der Berechnung der Pose zu erhalten. Die Anzahl der Gleichungen, auf denen die Berechnung stattfindet muss mindestens gleich der Anzahl der Freiheitsgrade sein, welche den zu bestimmenden Parametern entsprechen. Wichtig ist weiterhin, ob die Parameter direkt unabhängig voneinander berechnet werden können oder ob eine Zerlegung der Lösung notwendig ist. Das gegebene Problem, die externen Kameraparameter aus Punktekorrespondenzen herzuleiten, wurde von [FB81] als *Perspective-n-Point Problem* beschrieben. Wenn mindestens drei Punktkorrespondenzen gegeben sind, ist der Lösungsraum des Poseproblems endlich [FB81]. Es existieren jedoch bis zu vier mögliche Lösungen.

Bei der Verwendung von vier Korrespondenzen in beliebiger Konfiguration sind bis zu fünf Lösungen möglich. Für den Fall, dass sich die Korrespondenzen in einer koplanaren Anordnung befinden, ist das Problem jedoch eindeutig lösbar [GT06][HW02]. Mit fünf Korrespondenzen können sich im allgemeinen Fall bis zu zwei Lösungen ergeben. Für den koplanaren Fall ist die Lösung ebenfalls eindeutig. Ab sechs oder mehr Korrespondenzen ist das Poseproblem in beliebiger Konfiguration immer eindeutig lösbar. Daraus folgt, dass bei der Verwendung von nur vier oder fünf Korrespondenzen eine koplanare Konfiguration sichergestellt werden sollte. Mögliche Lösungen für Punkte sind in Tabelle 2 dargestellt. Dienen Linien als Merkmale zur Posebestimmung, so gilt hier im koplanaren Fall ein Minimum von drei Linienkorrespondenzen für eine eindeutige Lösung. Befinden sich die Linien nicht in koplanarer Anordnung, so ist das Minimum vier (Abbildung 28)[CH99].

Neben Anordnungen, die zu mehrdeutigen Lösungen führen können, sollten außerdem degenerierte, linear abhängige Konfigurationen vermie-



(a) Nicht zulässig (b) Zulässig

Abbildung 28: Linienkonfigurationen [CH99]

Anzahl Merkmale	Nicht koplanar	Koplanar
1	∞	-
2	∞	-
3	-	4
4	5	1
5	2	1
$\geq$6	1	1

Tabelle 2: Mögliche Lösungen für Anzahl und Konfiguration von Merkmalen.

den werden. Drei Punkte, die auf einer Geraden liegen, sind linear abhängig. Um auf *lineare Abhängigkeit* zu prüfen, wird der Richtungsvektor von einem Punkt zu den beiden anderen berechnet und normalisiert. Wenn das Skalarprodukt zwischen den beiden Richtungen nahe 1 ist, so sind die Vektoren linear abhängig und die Korrespondenzen werden für die Poseberechnung verworfen. Um *Koplanarität* zu überprüfen, wird mit den normalisierten Richtungsvektoren von drei Punkten die Normalform einer Ebene berechnet. Der verbleibende Richtungsvektor des vierten Punktes wird in die Gleichung eingesetzt. Wenn das Ergebnis unterhalb eines definierten Schwellwertes liegt, werden die vier Punkte als koplanar angenommen und

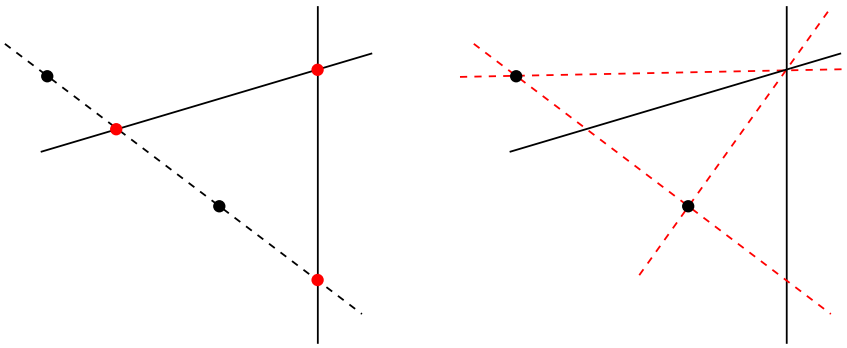


Abbildung 29: 2 Punkte - 2 Linien Konfiguration

die Korrespondenzen als Eingabe akzeptiert.

Bei Geradenkorrespondenzen treten linear abhängige Konfigurationen auf, wenn drei oder mehr Geraden parallel sind oder sich in einem Punkt schneiden. *Parallelität* wird durch die Berechnung des Skalarprodukts der normalisierten Geradenrichtung geprüft. Wenn alle Ergebnisse nahe 1 liegen, sind die Geraden linear abhängig und werden als Korrespondenzen verworfen. Die Prüfung auf einen *gemeinsamen Schnittpunkt* von mindestens drei Geraden erfolgt durch paarweisen Schnittpunkttest. Wenn zwei oder mehr Schnittpunkte existieren und ihre euklidische Distanz unterhalb eines definierten Schwellwertes liegt, gelten die Geraden als kollinear und werden verworfen.

Ferner lässt sich die Poseberechnung auch in Kombination von Punkten und Linien durchführen. Neben vier Linien oder vier Punkten ist auch der Einsatz von drei Punkten und einer Linie sowie drei Linien und einem Punkt möglich. Da drei Punkte die drei Linien eines Dreiecks definieren und umgekehrt drei Linien ein Dreieck bilden, das drei Schnittpunkte erzeugt, sind diese Fälle gleichzusetzen mit der Verwendung von vier Punkten oder vier Linien. Aus zwei Punkten und zwei Linien lässt sich jedoch keine Pose berechnen, da dies zu dem degenerierten Fall von fünf Linien führt, von

denen sich vier in einem Punkt schneiden oder analog zu fünf Punkten, von denen vier kollinear sind (Abbildung 29).

5.4 Nichtlineare Optimierung

Nichtlineare Verfahren dienen der numerisch iterativen Lösung von Optimierungsproblemen, im Allgemeinen zur Minimierung oder Maximierung einer Fehlerfunktion [JS04][Alt02]. Sie sind sehr genau und robust gegen Rauschen in den Messwerten. Sie können jedoch durch eine geringe Konvergenzrate sehr langsam sein und konvergieren unter Umständen nicht gegen das globale sondern nur gegen ein lokales Minimum oder Maximum. Daher bedarf es eines guten initialen Startwertes, der nahe genug am gesuchten Optimum liegen muss, um die Konvergenz in ausreichendem Maße zu gewährleisten. *Kombinierte Verfahren* berechnen linear einige Parameter als Startpunkt für die Iteration und führen dann eine nichtlineare Optimierung durch. Sie sind genauer als rein lineare Verfahren und konvergieren bei guten Startwerten in nur sehr wenigen Iterationen. Je nachdem welches Optimierungsproblem vorliegt, muss eine geeignete Fehlerfunktion f gewählt werden, welche in Abhängigkeit vom gesuchten Parametervektor $\vec{a}$ der Pose solange iterativ minimiert wird, bis ein Fehlerschwellwert unterschritten wird oder die maximale Iterationszahl erreicht ist:

$$f(\vec{a}) = \sum_{n=1}^{2m} r_n(\vec{a})^2 \rightarrow \min$$

mit

$$\vec{r_n} = \begin{pmatrix} \vec{r_1} \\ \vec{r_2} \\ \vdots \\ \vec{r_m} \end{pmatrix}.$$

Der Parametervektor $\vec{a}$ setzt sich sich aus Translation und Rotation zusammen. Die Parametrisierung ist in Abschnitt 5.7 beschrieben. Der zu minimierende Vektor $\vec{r_n}$ wird dabei kombinierter Residuenvektor genannt. Er enthält für jede der m Korrespondenzen einen entsprechenden Residuenvektor, der sich aus den im Folgenden definierten Fehlermaßen ergibt.

5.5 Fehlermaße für Punkte

Da das Ziel die Minimierung der Distanz zwischen den transformierten Modellmerkmalen und den Bildmerkmalen ist, müssen Fehlermaße für die Darstellung der Residuen definiert werden. Es werden drei verschiedene Punktfehlermaße untersucht, die in unterschiedlichen Koordinatensystemen arbeiten (Abbildung 30).

5.5.1 Rückprojektionsfehler

Der bekannte Rückprojektionsfehler misst die Distanz zwischen Bildmerkmal q^p und dem Modellmerkmal p^p nach dessen Projektion auf die Bildebene in Pixelkoordinaten. Folglich wird die Residue jeder Korrespondenz k_p dargestellt als

$$\vec{r}^{RE} = \vec{p^p} - \vec{q^p} = \begin{pmatrix} p_x^p - q_x^p \\ p_y^p - q_y^p \end{pmatrix}.$$

5.5.2 Objektraumfehler

Der Objektraumfehler misst die Distanz in Kamerakoordinaten und wurde von [LHM00] beschrieben. Um die Tiefeninformation des Bildmerkmals zu gewinnen, wird der Richtungsvektor $\vec{p^c}$ vom Kamerazentrum zum Modellpunkt auf den normalisierten Richtungsvektor $\hat{\vec{q^c}}$ des Bildmerkmalspunktes projiziert. Der sich ergebende Skalar dient der Skalierung des normalisierten Richtungsvektors $\hat{\vec{q^c}}$. Beide Vektoren werden anhand ihrer Kamerakoordinaten verglichen. Die Residue jeder Korrespondenz k_p ist daher

$$\vec{r}^{OE} = \begin{pmatrix} p_x^c - \left\langle \hat{\vec{q^c}}, \vec{p^c} \right\rangle \hat{q^c}_x \\ p_y^c - \left\langle \hat{\vec{q^c}}, \vec{p^c} \right\rangle \hat{q^c}_y \end{pmatrix}.$$

Es sollte angemerkt werden, dass Punkte, deren Projektionen auf die Bildebene gleich sind, einen größeren Fehler bei größeren Abständen von der Bildebene erzeugen. Dieses Fehlermaß ist also nicht tiefeninvariant.

5.5.3 Normalenfehler

Der Normalenfehler wird ebenfalls in Kamerakoordinaten gemessen. Vom Bildmerkmalsvektor $\vec{q^c}$ werden zwei orthogonale Normalen $\vec{n_1^c}$ und $\vec{n_2^c}$ definiert, welche die Richtung des Vektors in Kamerakoordinaten beschreiben:

$$\vec{n_1^c} = \begin{pmatrix} -q_y^c \\ q_x^c \\ 0 \end{pmatrix}, \vec{n_2^c} = \begin{pmatrix} -q_z^c \\ 0 \\ q_x^c \end{pmatrix}.$$

Das Skalarprodukt dieser normalisierten Normalen und des Richtungsvektors des Modellpunktes $\vec{p^c}$ liefert ein Maß der Richtungsabweichung zwischen Bildmerkmals- und Modellvektor. Die Residue jeder Korrespon-

denz k_p ist

$$\vec{r}^{NE} = \begin{pmatrix} \left\langle \hat{\vec{n}}_1^c, \vec{p}^c \right\rangle \\ \left\langle \hat{\vec{n}}_2^c, \vec{p}^c \right\rangle \end{pmatrix}.$$

Der Normalenfehler ist ebenfalls nicht tiefeninvariant.

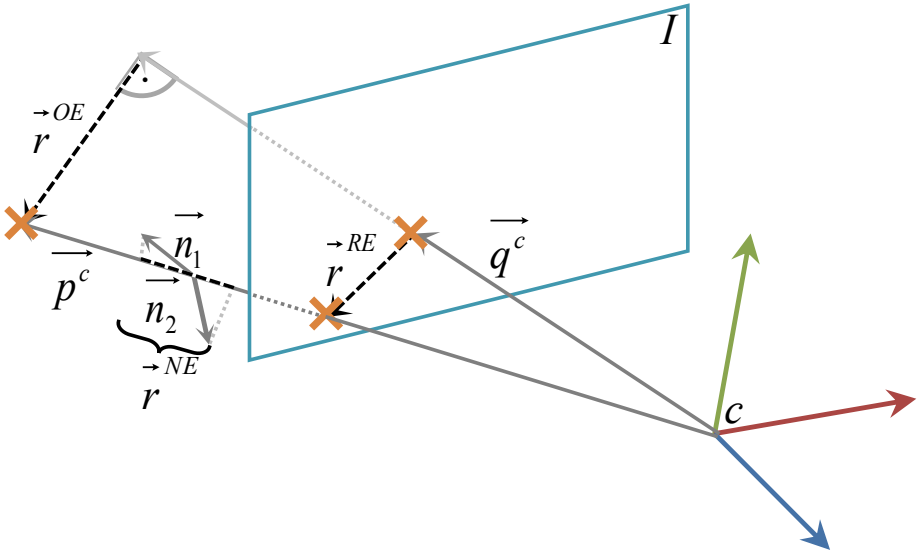


Abbildung 30: Punktfehlermaße. Dargestellt sind das Kamerakoordinatensystem c, die Bildebene I, die Vektoren für Modell- und Bildmerkmal $\vec{p^c}$ und $\vec{q^c}$, die Normalen des Normalenfehlers $\vec{n_1}$ und $\vec{n_2}$ zusammen mit ihren Projektionen auf $\vec{p^c}$ und die Fehlermaße $\vec{r}^{RE}$, $\vec{r}^{OE}$ und $\vec{r}^{NE}$.

5.6 Fehlermaße für Geraden

Es werden drei verschiedene Fehlermaße für Geraden untersucht, welche den Fehler in unterschiedlichen Koordinatensystemen angeben (Abbildung 31).

5.6.1 Winkel-/Abstandsfehler

Der Winkel-/Abstandsfehler arbeitet auf zwei unterschiedlichen Dimensionen und misst die Differenzen der Winkel und Abstände von Bild- und

Modellmerkmal. Der Start- und Endpunkt des Modellmerkmals werden in Pixelkoordinaten $\vec{s^p}, \vec{e^p}$ transformiert, um die entsprechende Gerade im Bild $\ell^p : (\varphi^p, d^p)$ zu berechnen, welche mit dem Bildmerkmal $l^p : (\phi^p, \rho^p)$ anhand der Parameter verglichen wird. Die Residue jeder Korrespondenz k_l ist

$$\vec{r}^{AE} = \begin{pmatrix} \varphi^p - \phi^p \\ d^p - \rho^p \end{pmatrix}.$$

5.6.2 Geradenfehler

Der Geradenfehler misst die Distanz der projizierten Start- und Endpunkte der Modellkante $\vec{s^p}, \vec{e^p}$ zur Bildgerade l^p durch Lösung nach der Normalform der Bildgeraden und wurde unter anderem von [Low91] eingesetzt. Mit der Geradengleichung

$$\cos\phi^p p_x^p + \sin\phi^p p_y^p = \rho^p$$

ist die Residue jeder Korrespondenz k_l definiert als

$$\vec{r}^{LE} = \begin{pmatrix} \cos\phi^p s_x^p + \sin\phi^p s_y^p - \rho^p \\ \cos\phi^p e_x^p + \sin\phi^p e_y^p - \rho^p \end{pmatrix}.$$

5.6.3 Ebenenfehler

Der Ebenenfehler kann als Erweiterung des Objektraumfehlers für Geraden betrachtet werden und wurde von [KH94] angewendet. Das Kamerazentrum und die Bildgerade definieren eine Ebene E^c in Kamerakoordinaten mit der Ebenennormale $\vec{n_E^c} = \begin{pmatrix} \cos\phi & \sin\phi & -\rho \end{pmatrix}^T$, welche sich aus dem Kreuzprodukt des Richtungsvektors der Bildgeraden und der Verbindung des Kamerazentrums zu einem Punkt auf der Geraden herleiten lässt. Für Start- und Endpunkt des Modellmerkmals in Kamerakoordinaten $\vec{s^c}, \vec{e^c}$

kann die Distanz zu dieser Ebene berechnet werden. Die Residue jeder Korrespondenz k_l ist:

$$\vec{r}^{PE} = \begin{pmatrix} \left\langle \vec{n}_E^c, \vec{s}^c \right\rangle \\ \left\langle \vec{n}_E^c, \vec{e}^c \right\rangle \end{pmatrix}.$$

Wie der Objektraumfehler, ist auch der Ebenenfehler nicht tiefeninvariant.

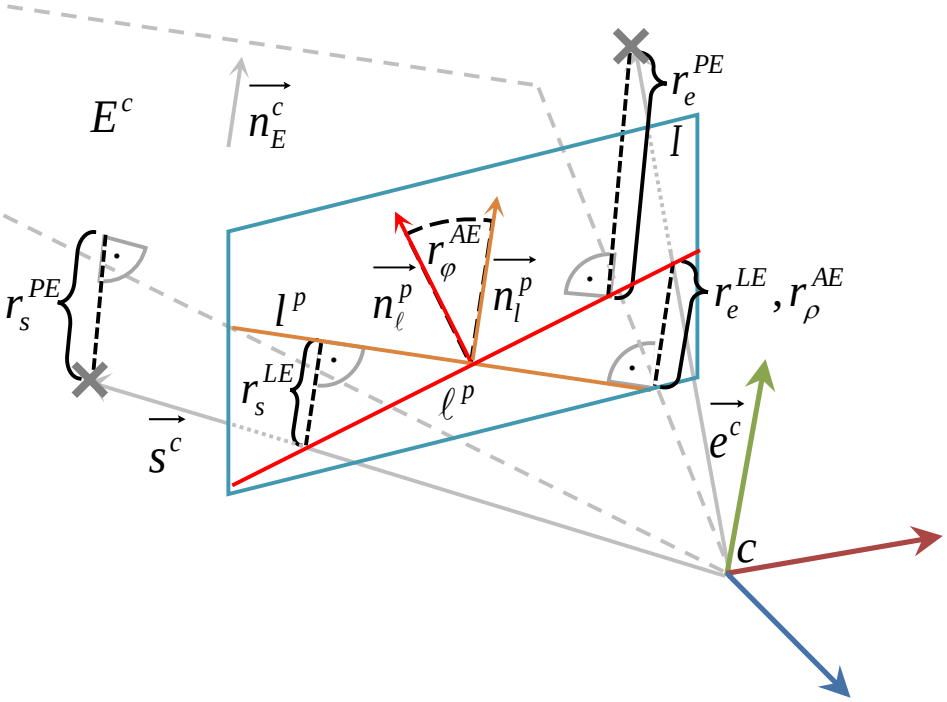


Abbildung 31: Fehlermaße für Geraden. Dargestellt werden das Kamerakoordinatensystem c, die Bildebene I, das Modell- und Bildmerkmal $\vec{s^c}$, $\vec{e^c}$ und l^p, die Gerade ℓ^p, ihre Normalen $\vec{n_l^p}$ und $\vec{n_\ell^p}$, die Ebene E^c mit ihrer Normalen $\vec{n_E^c}$ und die Komponenten der Fehlermaße $\vec{r}^{AE}$, $\vec{r}^{LE}$ und $\vec{r}^{PE}$.

5.7 Parametrisierung

Im Folgenden werden Einfluss und Bedingungen verschiedener Parametrisierungen der Pose $C : (R, t^c)$ untersucht. Die Parametrisierung des Translationsvektors wird durch die drei Skalare t_x^c, t_y^c und t_z^c repräsentiert, welche

die Verschiebung der Kamera in den Ursprung des Weltkoordinatensystems definieren. Der entsprechende Parametervektor ist

$$\vec{a^t} = \begin{pmatrix} t_x^c \\ t_y^c \\ t_z^c \end{pmatrix}.$$

Für die Parametrisierung der Rotation existieren vier Ansätze. Alle Rotationen werden durch ihre entsprechende Transformationsmatrix R beschrieben und müssen die Bedingungen einer gültigen Rotation, also die einer speziellen orthogonalen Gruppe (SOP) mit $R \in SO(3, \mathbb{R})$ erfüllen. Der Matrix R sollte eine orthonormale Basis zugrunde liegen und sie sollte eine Determinante $+1$ besitzen. Im Allgemeinen können diese Bedingungen - auch Drehgruppeneigenschaften genannt - in verschiedenen Stufen des Optimierungsprozesses sichergestellt werden:

I. *Vor der Optimierung*

   Manche Parametrisierungsformen der Rotation erfüllen die Bedingungen der SOP bereits per Definition teilweise oder vollständig.

II. *Während der Optimierung*

   Die SOP können als zusätzliche Elemente in den Residuenvektor aufgenommen und zusammen mit den anderen Residuen optimiert werden. Ein Nachteil dieser Alternative ist, dass ungenaue oder gar falsche Korrespondenzen nur zur Minimierung der Bedingungen führen, sie jedoch nicht vollständig erzwingen.

III. *Nach der Optimierung*

   Nach der Optimierung kann eine Singulärwertzerlegung (SVD), wie in [CM06] beschrieben, angewendet werden, um aus der geschätz-

ten Transformationsmatrix R eine korrigierte Rotationsmatrix $\tilde{R}$ zu berechnen, welche die SOP erfüllt.

Diese Vorgehensweisen schließen sich jedoch nicht gegenseitig aus. Alternative II sollte mit Alternative III kombiniert werden, um die Bedingungen der SOP zu erfüllen.

5.7.1 Matrix-Parametrisierung

Die einfachste Parametrisierung der Rotation ist eine 3x3-Matrix, die sich aus neun unabhängigen Werten zusammensetzt. Die Zeilen der Matrix sind dabei als die drei Vektoren $\vec{i}, \vec{j}, \vec{k} \in \mathbb{R}^3$ des gedrehten Kamerakoordinatensystems definiert, die den Parametervektor $a^{\vec{R}}$ ergeben:

$$a^{\vec{R}} = \begin{pmatrix} i^{\vec{T}} \\ j^{\vec{T}} \\ k^{\vec{T}} \end{pmatrix} = \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \\ k_x & k_y & k_z \end{pmatrix}.$$

Der Vorteil dieser Parametrisierung ist die leichte Berechenbarkeit. Die Bedingungen der SOP jedoch werden nicht erfüllt, sodass die Alternativen II und III angewendet werden müssen: Der Matrix R sollte eine orthonormale Basis zugrunde liegen und sie sollte eine Determinante $+1$ besitzen. Beide Bedingungen können den Residuen hinzugefügt werden. Nachteilig ist zusätzlich, dass die Rotation durch mehr Parameter beschrieben wird, als Freiheitsgrade zu bestimmen sind.

5.7.2 Euler-Winkel-Darstellung

Eine weitere Möglichkeit ist die Darstellung der Rotation durch drei verkettete Rotationen um die Koordinatenachsen. Jede Rotation wird durch eine eigene Rotationsmatrix $R_{\phi_x}, R_{\phi_y}, R_{\phi_z}$ mit ihren Drehwinkeln ϕ_x, ϕ_y, ϕ_z

beschrieben, deren Hintereinanderausführung die vollständige Rotation ergibt:

$$R = R_{\phi_z} R_{\phi_x} R_{\phi_y} = \begin{pmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{pmatrix} \begin{pmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{pmatrix}$$

$$= \begin{pmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z \\ c_y s_z + c_z s_x s_y & c_x c_z & s_y s_z + c_y c_z s_x \\ -c_x s_y & s_x & c_x c_y \end{pmatrix}$$

mit $c_x = \cos\phi_x$, $c_y = \cos\phi_y$, $c_z = \cos\phi_z$, $s_x = \sin\phi_x$, $s_y = \sin\phi_y$ und $s_z = \sin\phi_z$. Der Parametervektor ergibt sich aus $\vec{a^R} = \begin{pmatrix} \phi_x & \phi_y & \phi_z \end{pmatrix}^T$.

Nachteilig bei dieser Parametrisierung ist der extensive Einsatz trigonometrischer Funktionen, obgleich sie nur durch drei Parameter beschrieben ist, was der Anzahl der zu bestimmenden Freiheitsgrade entspricht.

Da die Matrixmultiplikation von drei Rotationsmatrizen eigenschaftserhaltend ist, werden die SOP bereits per Definition erfüllt und müssen nicht erzwungen werden. Dabei ist zu beachten, dass die Reihenfolge der Rotationen bestimmt sein muss, da die Matrixmultiplikation nicht kommutativ ist. Problematisch ist das mögliche Zusammenfallen zweier Rotationsachsen, wobei es zu dem sogenannten *gimbal lock* kommt, dem Verlust eines Freiheitsgrades. Dieser Fall kann vermieden werden, indem abhängig von der Anwendung eine feste Rotationsreihenfolge definiert wird, bei welcher das Überlagern zweier Rotationsachsen unwahrscheinlich ist. Im hier vorliegenden Anwendungsbereich des Gebäudetrackings ist dies ein senkrechter Blick von oben oder unten auf das Gebäude, bei dem die z-Achse mit der y-Achse übereinstimmen würde. Daraus ergibt sich die gewählte Folge der

Rotationsmatrizen $R = R_{\phi_z} R_{\phi_x} R_{\phi_y}$.

5.7.3 Quaternionen-Parametrisierung

Die dritte Darstellung von Rotationen wird durch Quaternionen ermöglicht. Die Rotation wird um eine definierte Achse $\vec{v} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ mit einem Winkel θ durchgeführt. Dabei gilt $\theta = 2\cos^{-1} w$. Das Einheitsquaternion hat die Form $q : (w, x, y, z) \in \mathbb{H}$ und der daraus folgende Parametervektor der Quaternionen-Parametrisierung ist $a^{\vec{R}} = \begin{pmatrix} w & x & y & z \end{pmatrix}^T$. Die zugehörige Rotationsmatrix wird aus dem Quaternion wie folgt berechnet:

$$R = \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & w^2 - x^2 + y^2 - z^2 & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & w^2 - x^2 - y^2 + z^2 \end{pmatrix}.$$

Da q ein Einheitsquaternion ist, werden die SOP nicht vollständig erfüllt und die Alternativen II und III müssen angewendet werden, indem die Bedingung $|q| = 1$ zu den Residuen hinzugefügt wird. Es ist ebenfalls möglich, w durch x, y und z als $w = \sqrt{1 - x^2 - y^2 - z^2}$ zu ersetzen. Der Nachteil dabei sind allerdings mögliche komplexe Lösungen für w. Wie bei der Matrix-Parametrisierung wird auch hier die Rotation durch mehr Parameter beschrieben, als Freiheitsgrade zu bestimmen sind. Von Vorteil sind die einfache Berechenbarkeit sowie das Nichtauftreten kritischer Konfigurationen.

5.7.4 Rodrigues-Formel-Parametrisierung

Eine weitere Möglichkeit ist die Definition einer beliebigen Rotationsachse $\vec{v} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ und eines Rotationswinkels θ. Gegenüber den Quaternio-

nen wird ein Parameter eingespart, da der Rotationswinkel aus dem Betrag der Rotationsachse hervorgeht: $\theta = |\vec{v}|$. Der Parametervektor der Parametrisierung nach Rodrigues ist folglich $a^{\vec{R}} = \begin{pmatrix} x & y & z \end{pmatrix}^T$. Als Rotationsmatrix ergibt sich:

$$R = \begin{pmatrix} \cos\theta + x^2 c_\theta & xyc_\theta - zs_\theta & ys_\theta - xzc_\theta \\ zs_\theta + xyc_\theta & \cos\theta + y^2 c_\theta & yzc_\theta - xs_\theta \\ xzc_\theta - ys_\theta & xs_\theta + yzc_\theta & \cos\theta + z^2 c_\theta \end{pmatrix}$$

mit $c_\theta = \frac{1-\cos\theta}{\theta^2}$, $s_\theta = \frac{\sin\theta}{\theta}$ und $\theta = \sqrt{x^2 + y^2 + z^2}$. Im Falle von $\theta = 0$ gilt $R_R = I_3$. Die SOP sind auch hier bereits per Definition erfüllt und müssen nicht erzwungen werden. Ferner treten keine kritischen Konfigurationen auf, die Berechnung der Rotationsmatrix ist jedoch aufwändiger.

5.8 Optimierung

5.8.1 Newton-Verfahren

Mit dem Newton-Verfahren kann das Minimum einer nichtlinearen Funktion iterativ berechnet werden. Grundlage des Verfahrens ist die Newton-Iteration (Newton-Raphsonsche Methode), welche der Approximation der Nullstelle $f(x) = 0$ einer Funktion mit dem Parameter x dient. Die Nullstelle findet sich durch die erste Ableitung $f'(x)$ einer Funktion (Abbildung 32). Anhand der Tangente wird die Funktion lokal in x_n linearisiert und die Nullstelle der Tangente ergibt die Annäherung x_{n+1} an die tatsächliche Nullstelle der Funktion, welche als Startpunkt der nächsten Iteration dient:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Um nun das Minimum einer Funktion zu berechnen, muss jedoch an-

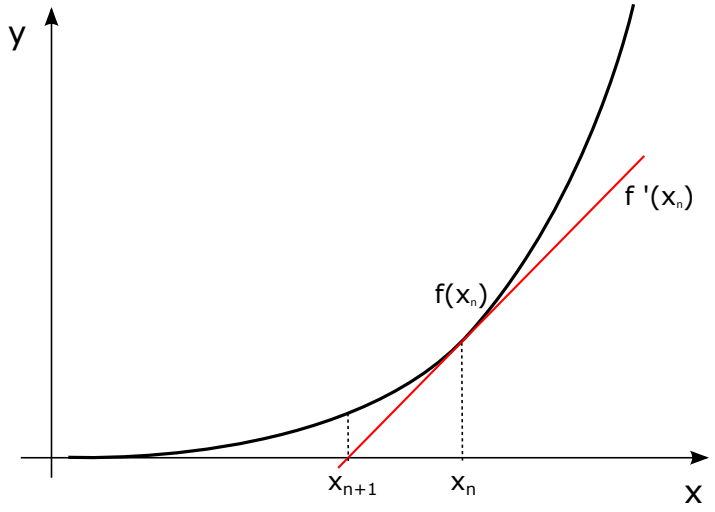


Abbildung 32: Nullstellensuche mit Newton-Iteration

stelle der Nullstelle der eigentlichen Funktion die Nullstelle der ersten Ableitung gefunden werden, denn es gilt

$$\min f(\mathbf{x}) \Leftrightarrow f'(\mathbf{x}) = 0.$$

Der nächste Wert der Minimierung berechnet sich durch die Erweiterung der Newton-Iteration um die zweiten Ableitungen der Funktion f in Form der Hessematrix:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \nabla^2 f(\mathbf{x}_n)^{-1} \nabla f(\mathbf{x}_n),$$

wobei der Gradientenvektor der ersten partiellen Ableitungen als

$$f'(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, ..., \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T$$

defniniert ist und die Hessematrix als

$$H(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{pmatrix}.$$

Ist das Optimierungsproblem mehrdimensional, werden die zu minimierenden Einträge des Residuenvektors $\vec{r}$ zur Jacobi-Matrix oder *Funktionalmatrix* zusammengefasst:

$$J_{\vec{r}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial r_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial r_1(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial r_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial r_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \nabla^2 f(\mathbf{x}_n)^{-1} J_{\vec{r}}(\mathbf{x}_n).$$

Der Parametervektor $\mathbf{x} = (x_1, ..., x_n)$ minimiert die Funktion $f(\mathbf{x})$, wenn es einen Satz $\mathbf{x}^*$ von Parametern gibt, der

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x}$$

erfüllt. Es handelt sich um ein lokales Minimum, wenn die Bedingung nur für eine begrenzte Umgebung von $\mathbf{x}^*$ gilt und nicht für alle $\mathbf{x}$.

Die Konvergenz wird über die Größe der Parameteränderung zwischen zwei Iterationen geprüft. Die Optimierung wird beendet, wenn die Änderung unter einer Fehlerschranke ϵ liegt:

$$|\mathbf{x}_{n+1} - \mathbf{x}_n| < \epsilon.$$

Das Newton-Verfahren konvergiert schnell, ist aber nicht robust gegen

fehlerhafte Daten und durch die Bildung der zweiten Ableitung recht aufwändig. So ist es möglich, dass bei ungünstigen Startwerten, die nicht nahe genug am Minimum liegen, keine Konvergenz erreicht wird, oder das Verfahren nur zu einem lokalen Minimum hin konvergiert.

5.8.2 Gauß-Newton

Das Gauß-Newton-Verfahren ist eine Vereinfachung des reinen Newton-Verfahrens. Die Summe der Residuen von erwarteten und gemessenen Datenpunkten wird minimiert, indem die Funktion bei jeder Iteration durch eine quadratische Näherung ersetzt wird. Die teure zweite Ableitung muss dabei nicht explizit berechnet werden, da sie durch die Jacobimatrix ersetzt werden kann. Das Verfahren ist daher schnell, leidet jedoch unter denselben Stabilitätsproblemen wie das Newton-Verfahren, da die Schrittweite der Konvergenz nicht gesteuert werden kann. Die Hessematrix wird ersetzt mit

$$\nabla^2 f(\mathbf{x}) = J_{\vec{r}}(\mathbf{x})^T J_{\vec{r}}(\mathbf{x})$$

und

$$\nabla f(\mathbf{x}) = J_{\vec{r}}(\mathbf{x})^T \vec{r}(\mathbf{x})$$

sodass sich als Iterationsregel ergibt:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (J_{\vec{r}}(\mathbf{x}_n)^T J_{\vec{r}}(\mathbf{x}_n))^{-1} J_{\vec{r}}(\mathbf{x}_n)^T \vec{r}(\mathbf{x}_n).$$

5.8.3 Gradientenabstiegsverfahren

Beim Gradientenabstiegsverfahren wird bei jeder Iteration der nächste Schritt entlang des negativen Gradienten und damit genau entgegen des

steilsten Anstiegs gewählt (*Down-Hill Methode*):

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \nabla f(\mathbf{x}_n).$$

Der Parameter α steuert die Schrittweite und kann adaptiv bestimmt werden. Verfahren zur Bestimmung der Schrittweite finden sich in [Alt02]. Das Verfahren garantiert zwar die Konvergenz, Nachteil des Gradientenabstiegs ist jedoch eine hohe Iterationszahl, wenn ein ungeeigneter Schrittweitenparameter gewählt wird.

5.8.4 Levenberg-Marquardt

Die Gauß-Newton Optimierung ist vergleichsweise schnell, garantiert aber keine Konvergenz. Das Gradientenabstiegsverfahren ist dagegen bei garantierter Konvergenz recht langsam. Der Levenberg-Marquardt-Algorithmus vereint dabei die Vorteile beider Verfahren. Er entspricht einer Kombination von Gauß-Newton-Verfahren und der Methode des Gradientenabstiegs. Durch eine Schrittweitenkontrolle der Abstiegsrichtung werden absteigende Funktionswerte erzwungen, was zu einer schnelleren Konvergenz führt. Dadurch wird auch bei schlechten Startwerten eine gute Konvergenz erreicht, die jedoch langsamer ist, wenn sich die Startwerte bereits nahe am Minimum befinden.

In jeder Iteration wird überprüft, ob sich der Fehler durch den aktuellen Schritt verringert. In diesem Fall wird die Schrittweite λ verkleinert. Andernfalls wird sie vergrößert, bis ein Wert gefunden ist, der den Fehler reduziert. Mit diesem λ wird dann die nächste Iteration gestartet. Ist der Fehler gering, also die Schrittweite sehr klein, und die Optimierung befindet sich nahe am Minimum, dann arbeitet der Levenberg-Marquardt Algorithmus wie das Gauß-Newton Verfahren mit schneller Konvergenz. Bei großem Fehler mit

großer Schrittweite entspricht der Levenberg-Marquardt Algorithmus dem Abstieg entlang des Gradienten, was die Vermeidung von lokalen Minima begünstigt.

Aufgrund der Vorteile wird eine Levenberg-Marquardt Optimierung, wie in [PTVF92] beschrieben, eingesetzt. Das Ziel ist die Minimierung der Summe der Quadrate aller Einträge des kombinierten Residuenvektors $\vec{r}$ bei Optimierung der Poseparameter $\vec{a}$. Die Iterationsregel ist

$$\vec{a}_{n+1} = \vec{a}_n - \left(J_{\vec{r}}(\vec{a}_n)^T J_{\vec{r}}(\vec{a}_n) + \lambda D \right)^{-1} J_{\vec{r}}(\vec{a}_n)^T \vec{r}(\vec{a}_n)$$

mit

$$D = \operatorname{diag}\left(J_{\vec{r}}(\vec{a}_n)^T J_{\vec{r}}(\vec{a}_n) \right)$$

und der Jacobimatrix $J_{\vec{r}}$ von $\vec{r}$. Als optimal hat sich ein Initialwert für $\lambda = 10^{-3}$ und die Iterationsregel $\lambda_n = \lambda_{n-1}/10$ für Residuenminimierung, sowie $\lambda_n = \lambda_{n-1} * 10^w$ mit $w \in \mathbb{N}$ für Residuenmaximierung herausgestellt.

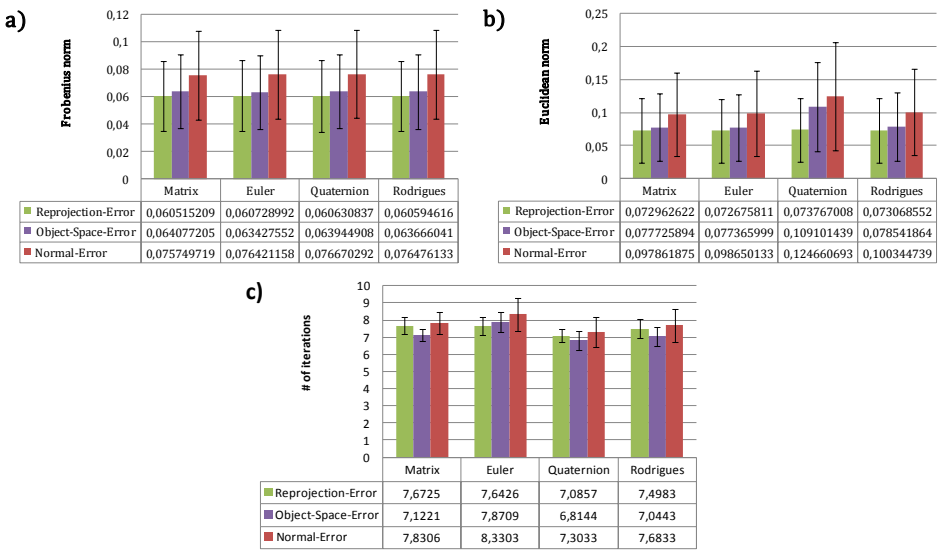
5.9 Skalierung

Der kombinierte Residuenvektor kann aus Korrespondenzen bestehen, die mit unterschiedlichen Fehlermaßen bewertet wurden. Da durch die Diskretisierung und fehlerhaftes Matching nicht von perfekten Korrespondenzen ausgegangen werden kann, führen ungleiche Gewichtungen zu Verschiebungen der Minima der quadrierten Summe. Um optimale Werte für die kleinsten Quadrate zu erreichen, sollten die Dimensionen aller Einträge der Residuen gleich sein. Bis auf den Winkel-/Abstandsfehler haben alle Fehlermaße und SOP eine einheitliche Dimension in Pixeln oder Kamerakoordinaten, sodass die Skalierungsfaktoren der jeweils korrespondierenden Dimension durch die gegebene Kameramatrix erlangt werden können.

5.10 Ergebnisse

Zunächst wurden verschiedene Kombinationen von Fehlermaßen und Parametrisierungen jeweils bei einer nichtlinearen Poseschätzung mit Punkten, Geraden und gemischten Korrespondenzen getestet. Dazu wurden 100.000 Tests mit synthetischen Daten bei einer Auflösung von 640 x 480 Pixeln durchgeführt und aus diesen verschiedene Mengen idealer und künstlich verrauschter Korrespondenzen als Eingabe für die Poseschätzung generiert. Das Objekt befand sich im Ursprung des Weltkoordinatensystems, die initiale Kamerapose bei $(0, 0, -5)^T$ mit einer Rotation von $0°$. Während der Testsequenz wurde die Kamera um $30°$ um die Koordinatenachsen rotiert und mit $(1, 1, 1)^T$ von der initialen Pose transliert. Von dieser neuen Pose aus wurden 2D-3D Korrespondenzen durch perspektivische Projektion der 3D Modellmerkmale in die Bildebene gebildet und anschließend ein gleichförmig verteilter Fehler im Bereich von $[0, 10]$ Pixel in zufällige Richtung aufaddiert. Anhand dieser Korrespondenzen wurde die Pose berechnet. Aufgeführt sind der gemittelte Fehler und die Standardabweichung für Translation, Rotation und die Anzahl der Iterationen. Der Rotationsfehler wird als Frobeniusnorm der Differenzmatrix zwischen realer und geschätzter Rotation gemessen, der Translationsfehler als euklidische Norm des Differenzvektors der Translation.

Zunächst wurde der Einfluss der gewählten Fehlermaße und Parametrisierungen auf die Optimierungsstabilität für Punkte (Abbildung 33) und Geraden (Abbildung 34) untersucht. Bei Punktkorrespondenzen zeigt der Vergleich der Fehlermaße, dass der bildbasierte Rückprojektionsfehler unter allen Parametrisierungen präzisere Ergebnisse liefert, als der Objektraumfehler und der Normalenfehler. Dieser Umstand lässt sich damit erklären, dass im Objektraum weiter von der Kamera entfernte Merkmale eine poten-



	Matrix	Euler	Quaternion	Rodrigues
Reprojection-Error	0,060515209	0,060728992	0,060630837	0,060594616
Object-Space-Error	0,064077205	0,063427552	0,063944908	0,063666041
Normal-Error	0,075749719	0,076421158	0,076670292	0,076476133

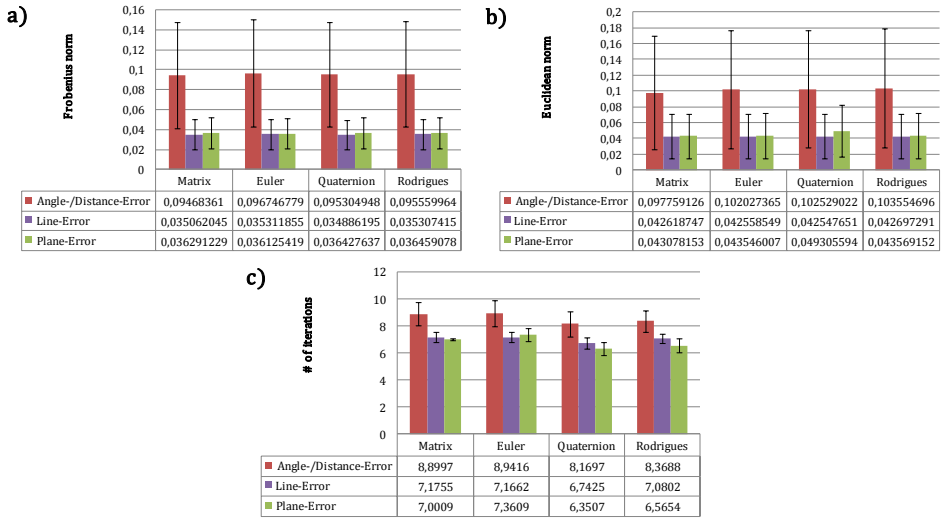
	Matrix	Euler	Quaternion	Rodrigues
Reprojection-Error	0,072962622	0,072675811	0,073767008	0,073068552
Object-Space-Error	0,077725894	0,077365999	0,109101439	0,078541864
Normal-Error	0,097861875	0,098650133	0,124660693	0,100344739

	Matrix	Euler	Quaternion	Rodrigues
Reprojection-Error	7,6725	7,6426	7,0857	7,4983
Object-Space-Error	7,1221	7,8709	6,8144	7,0443
Normal-Error	7,8306	8,3303	7,3033	7,6833

Abbildung 33: Punkte: Kombinationen von Fehlermaßen und Parametrisierungen; a) Rotationsfehler, b) Tanslationsfehler und c) Iterationszahl; Standardabweichung.

tiell höhere Gewichtung erfahren und sich deren Fehler stärker auswirkt. Auf die Anzahl der Iterationen während der Optimierung hat die Wahl des Fehlermaßes keinen nennenswerten Einfluss. Bei der Untersuchung der Parametrisierungen stellt sich heraus, dass die Euler-Winkel Darstellung etwas weniger performant ist und Quaternionen besser abschneiden. Letztere beeinflussen jedoch die Präzision der Translation. Bei Geradenkorrespondenzen führt der Einsatz des Winkel-/Abstandsfehlers zu schlechten Ergebnissen der geschätzten Pose in Verbindung mit höherer Rechenzeit. Geraden- und Ebenenfehler führen unter allen Parametrisierungen zu vergleichbaren Ergebnissen hinsichtlich Präzision und Geschwindigkeit. Auch bei Geradenkorrespondenzen zeigt die Quaternionenparametrisierung eine leicht bessere Performanz gegenüber den anderen Parametrisierungen.

Bei der Auswahl der Optimierungsstrategie zeigt der Algorithmus nach Levenberg-Marquardt als kombiniertes Verfahren aus Gauß-Newton Iteration und Gradientenabstiegsverfahren das beste Verhalten hinsichtlich



	Matrix	Euler	Quaternion	Rodrigues
Angle-/Distance-Error	0,09468361	0,096746779	0,095304948	0,095559964
Line-Error	0,035062045	0,035311855	0,034886195	0,035307415
Plane-Error	0,036291229	0,036125419	0,036427637	0,036459078

	Matrix	Euler	Quaternion	Rodrigues
Angle-/Distance-Error	0,097759126	0,102027365	0,102529022	0,103554696
Line-Error	0,042618747	0,042558549	0,042547651	0,042697291
Plane-Error	0,043078153	0,043546007	0,049305594	0,043569152

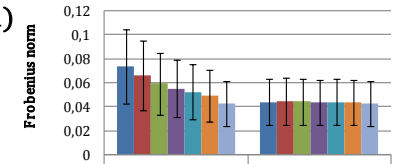
	Matrix	Euler	Quaternion	Rodrigues
Angle-/Distance-Error	8,8997	8,9416	8,1697	8,3688
Line-Error	7,1755	7,1662	6,7425	7,0802
Plane-Error	7,0009	7,3609	6,3507	6,5654

Abbildung 34: Kanten: Kombinationen von Fehlermaßen und Parametrisierungen; a) Rotationsfehler, b) Tanslationsfehler und c) Iterationszahl; Standardabweichung.

Konvergenzsicherheit und Konvergenzgeschwindigkeit. Daher wird dieser zusammen mit der Parametrisierung nach Rodrigues in Kombination mit dem pixelbasierten Rückprojektionsfehler und dem Geradenfehler für eine nichtlineare Poseschätzung mit Punkt- und Geradenkorrespondenzen vorgeschlagen. Diese Kombination stellt den besten Kompromiss zwischen geringstem Posefehler und Performanz dar. Durch die Wahl eines pixelbasierten Fehlermaßes für beide Merkmalstypen ist außerdem keine Skalierung notwendig, um die Differenz der Tiefe zwischen den Korrespondenzen auszugleichen. Der Vorteil der Parametrisierung nach Rodrigues ist, dass sie unabhängig von dem gewählten Fehlermaß die Präzision der Pose nicht beeinflusst. Des Weiteren ist diese Parametrisierung sicher, da keine Sonderfälle wie der Gimbal-Lock auftreten können. Die Bedingungen einer gültigen Rotationsmatrix werden ebenfalls bereits per Definition erfüllt und müssen nicht durch zusätzliche Berechnungen sichergestellt werden. Daher kann auf Zusatzbedingungen im Residuenvektor oder gar

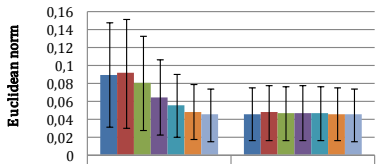
eine Singulärwertzerlegung zur Erzwingung der Bedingungen verzichtet werden.

a)



	Points	Lines
0 additional	0,073585423	0,043728671
1 additional	0,066206981	0,044546786
2 additional	0,059139783	0,04414166
3 additional	0,055122703	0,043747758
4 additional	0,052164676	0,043929681
5 additional	0,049424468	0,043525561
6 additional	0,042585884	0,042585884

b)



	Points	Lines
0 additional	0,089844951	0,046075405
1 additional	0,091286738	0,047450707
2 additional	0,080657104	0,047064991
3 additional	0,064888562	0,047084968
4 additional	0,055519589	0,04639309
5 additional	0,048644953	0,045911735
6 additional	0,045365612	0,045365612

Abbildung 35: Kombination von Punkten und Kanten; a) Rotationsfehler, b) Translationsfehler; Standardabweichung

Abschließend wurden Testläufe durchgeführt, um den Einfluss des Merkmalstyps, der Merkmalsanzahl, sowie den fehlerhafter Korrespondenzen auf die Robustheit der Pose zu bestimmen. Dazu wurde eine minimale Anzahl von sechs Korrespondenzen eines Merkmalstyps mit unterschiedlichen Zahlen des zweiten Merkmalstyps unter variierend stark ausgeprägtem Fehlerrauschen kombiniert (Abbildung 35). Durch das Hinzufügen von Geradenkorrespondenzen zu einer mit Punkten geschätzten Pose, konnte der Fehler in Rotation und Translation unabhängig von der Stärke des Rauschens der Punktkorrespondenzen verbessert werden. Der Effekt konvergiert bei mindestens sechs zusätzlichen Korrespondenzen. Im umgekehrten Fall zeigt sich bei Hinzunahme von Punkten zu einer mit Geradenkorrespondenzen geschätzten Pose ein positiver Effekt auf die Genauigkeit nur dann, wenn die Geradenkorrespondenzen bereits sehr verrauscht sind. Geraden zeigen sich durch ihre Dimension unempfindlicher gegenüber Pixelverschiebungen im Bild. Grundsätzlich ist eine kombinierte Poseschätzung mit gemischten Punkt- und Geradenkorrespondenzen geeignet, um in der praktischen Anwendung die geschätzte Pose zu stabilisieren, besonders

wenn nur sehr wenige Korrespondenzen zur Verfügung stehen.

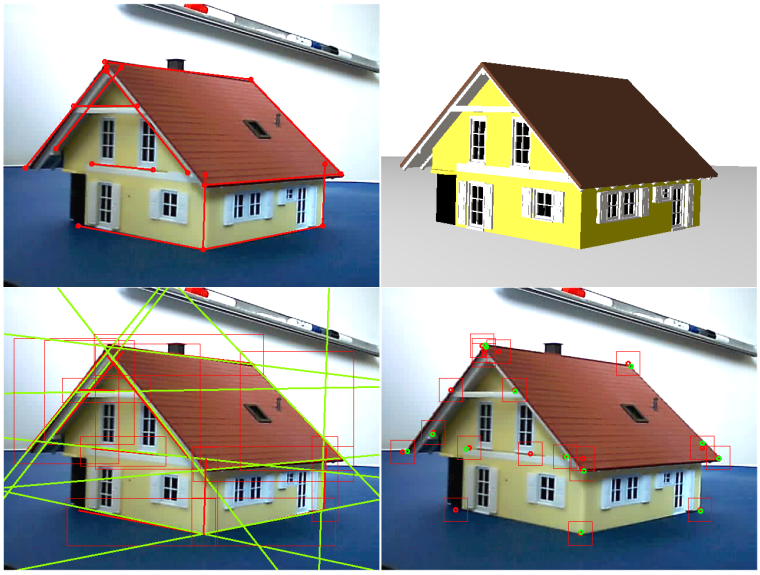


Abbildung 36: Testszene, oben links: Kamerabild mit überlagerter Pose (rot). Oben rechts: Modell. Unten links: Bildkanten-Korrespondenzen (grün) zu gegebenen Modellkanten mit ihren Suchfenstern (rot). Unten rechts: Bildpunkt-Korrespondenzen (grün) zu gegebenen Modellpunkten mit ihren Suchfenstern (rot).

Der aus diesen Ergebnissen konstruierte, nichtlineare Poseschätzer wurde erfolgreich auf realen Videosequenzen getestet. Abbildung 36 zeigt die Videosequenz, das Modell und die verwendeten Korrespondenzen. Nach der Projektion der Modellmerkmale werden Suchfenster von 20 Pixel im Bild gesetzt. Punktkorrespondenzen werden mit dem Harris Corner Detector und Geradenkorrespondenzen über die Hough-Transformation gebildet, indem die geringste euklidische Distanz zwischen projiziertem Modellmerkmal und Bildmerkmal gewählt wird. Die Initialpose wird als bekannt angenommen. Die Poseschätzung für kombinierte Korrespondenzen beanspruchte 1 ms auf der CPU und bewies damit Echtzeitfähigkeit, während die Korrespondenzsuche etwa 30 ms in Anspruch nahm. Weiterhin konnte festgestellt werden, dass die bei der Optimierung benötigte Anzahl an

Iterationen zum Großteil vom Grad des Fehlers in den Korrespondenzen abhängt, jedoch kaum von der Anzahl der eingesetzten Korrespondenzen.

5.11 Fazit

Es wurden Parameter für eine nichtlineare Poseschätzung analysiert, die das beste Ergebnis für eine kombinierte Eingabemenge an Punkt- und Kantenkorrespondenzen liefern. Die Testergebnisse zeigen, dass ein Fehlermaß in Pixelkoordinaten anderen Fehlermaßen sowohl bei der Verwendung von Punkten, als auch bei Kanten überlegen ist. Es hat sich gezeigt, dass eine Parametrisierung der Pose gewählt werden sollte, welche die Nebenbedingungen einer gültigen Rotation auch ohne zusätzlichen Rechenaufwand erfüllt. Bei beiden Merkmalstypen trifft dies auf die Achse-Winkel-Darstellung nach Rodrigues zu.

Mit den Erkenntnissen zu Optimierung, Fehlermaß und Parametrisierung wurde ein nichtlinearer Poseschätzer konstruiert, der auf einer beliebigen, kombinierten Anzahl von mindestens drei Punkt- und/oder Kantenkorrespondenzen arbeitet. Zusätzlich prüft der Poseschätzer die Eingabekorrespondenzen anhand des Modellwissens auf kritische geometrische Konfigurationen, um so mehrdeutige Poseergebnisse zu vermeiden. Bei der Analyse stellte sich heraus, dass sich Kanten besser für stabiles Tracking eignen. Eine durch Punktmerkmale berechnete Pose kann durch die Hinzunahme von Kantenmerkmalen hinsichtlich ihrer Präzision verbessert werden, was im umgekehrten Fall nicht eintritt. Von einer Kombination beider Merkmalstypen wurde daher im weiteren Verlauf abgesehen, da das Tracking vorwiegend auf Gebäudemodellen durchgeführt wird.

Im Rahmen der *Analyse-durch-Synthese* können durch das Wissen über die zu Grunde liegende Geometrie die Konfigurationen von Punkt- und Lini-

enmerkmalen hinsichtlich ihrer Eignung zur Posebestimmung bewertet werden. Dabei ist es möglich, durch Vorauswahl der Merkmale die konkreten Anforderungen des eingesetzten Pose-Algorithmus an die Eingabedaten zu erfüllen, um Ungenauigkeiten, Instabilitäten oder mehrdeutige Lösungen zu vermeiden. Ferner ist es denkbar, je nach Ergebnis der Merkmalsbewertung, das in der aktuellen Situation geeignetste Verfahren zur Posebestimmung dynamisch auszuwählen oder auch verschiedene Verfahren konkurrierend oder stabilisierend gegeneinander antreten zu lassen.

6 Merkmalsmanagement

Üblicherweise werden bei der Umsetzung von markerlosem Tracking alle Merkmale zur Poseschätzung herangezogen, zu denen Korrespondenzen zwischen aufeinander folgenden Kamerabildern (sequentielles Tracking) oder zwischen Modell- und Kamerabild (modellbasiertes Tracking) gefunden werden. Während des Trackingvorgangs kann es jedoch zu Problemen bei der Korrespondenzsuche kommen, wenn die Merkmale im nächsten Bild nicht eindeutig wiederzuerkennen sind. Angenommen wird, dass sich bei der Verwendung einer großen Anzahl von Merkmalen der Einfluss schlechter oder gar fehlerhafter Korrespondenzen nur gering auswirkt. Ein rein quantitatives Vorgehen steht jedoch im Widerspruch zur Bestrebung, Echtzeitfähigkeit - auch auf mobilen Geräten - zu erreichen.

Daher soll ein intelligentes Merkmalsmanagement das planvolle Erzeugen von nur wenigen, aber dafür qualitativ besonders gut geeigneten Merkmalen ermöglichen und so eine Korrespondenzsuche nach potentiell schlechten Merkmalen von vorneherein vermieden werden. Durch das gegebene Modell kann im *Analyse-durch-Synthese* Ansatz zusätzliches Wissen genutzt werden, um ein Qualitätsmaß für die Merkmale zu definieren. Hierbei wird zwischen zwei Arten von Kriterien unterschieden. Es handelt sich um persistente, also unveränderbare Kriterien, die sich aus dem Modell und seiner Verortung in der Welt ableiten, sowie dynamische Kriterien, welche standpunkt- und beleuchtungsabhängig sind. Dies betrifft die Geometrie und Topologie des Modells, Kamera und Perspektive, sowie beeinflussende Renderingparameter wie Beleuchtung, Material oder Erfolgsrate der Korrespondenzsuche (Abbildung 37). Anhand dessen können diejenigen Merkmale selektiert und priorisiert werden, die mit hoher Wahrscheinlichkeit im Kamerabild gut erkennbar sind und für das Tracking

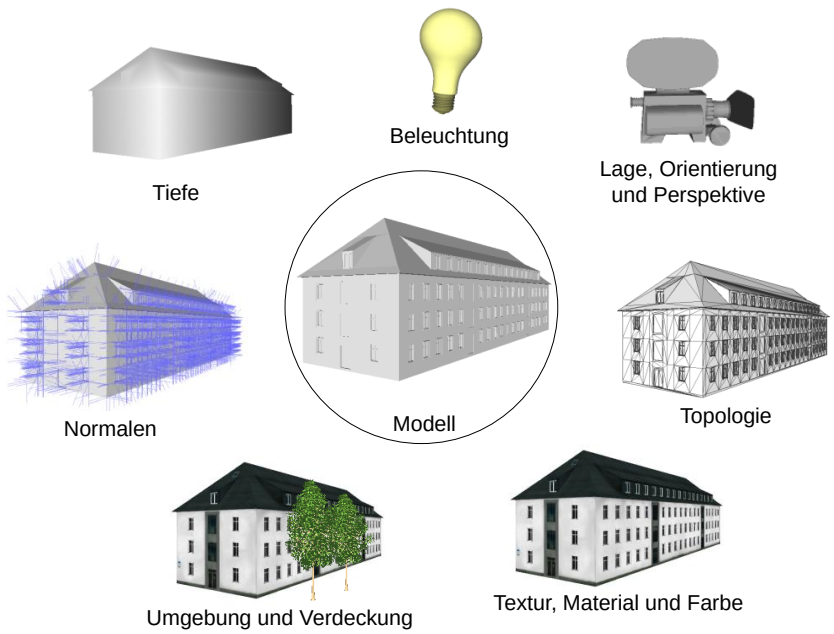


Abbildung 37: Qualitätskriterien

besondere Stabilität versprechen. Durch eine Einteilung in verschiedene Qualitätsstufen können die Merkmale während des Trackings dynamisch auf- und abgewertet werden, sodass immer die am höchsten bewerteten Merkmale eingesetzt werden. Dieses Vorgehen ermöglicht ein dynamisches Ranking der Merkmale, damit der Poseschätzer auf einer minimalen Anzahl von Merkmalen arbeiten kann und bei zu wenigen als gut bewerteten Merkmalen auf niedrigere Priorisierungsstufen zurückgreift. Das Verfahren wurde als „Intelligent Feature Management for 3D Model-Based Camera Pose Tracking“ auf der *9th International Conference on Computer Vision Theory and Applications (VISAPP)* veröffentlicht [SHM14].

Evaluierung und Selektion von Merkmalen wurde bereits erfolgreich auf Bildsequenzen durchgeführt, jedoch nicht unter Einbeziehung eines 3D Modells. In [ST94] wird die Qualität der Bildmerkmale anhand ihrer Veränderung über die Bildsequenz festgestellt, um Verdeckungen und andere

Störeinflüsse zu erkennen. Im ersten Bild der Sequenz werden Merkmale referenziert und die gefundenen Korrespondenzen des aktuellen Bildes mit ihrer Referenz verglichen. Eine hohe Abweichung der Ähnlichkeit zeigt, dass das Merkmal nicht stabil ist und verworfen werden sollte. Dabei wird die Qualität jedoch nur über die Bildintensitäten bewertet. Ein weiterer Ansatz zur Selektion optimaler Merkmale für das Tracking ist in [CLL05] dargestellt. Die Evaluierung und das Ranking der Merkmale beruht hier auf ihrer Eignung, Objekte gut vom Hintergrund zu trennen. In [WPS07] werden die Merkmale anhand der Wahrscheinlichkeit beurteilt, mit der sie von einer rekonstruierten Kamerapose aus zu erkennen sind. Nur solche Merkmale werden zum Tracking verwendet, die von der aktuellen Kamerapose aus mit hoher Wahrscheinlichkeit gut detektiert werden können. Merkmale ohne gültige Rekonstruktion der Kamerapose werden verworfen.

In Abbildung 38 ist der Managementprozess dargestellt. Da das 3D Modell nach dem Laden ein vollständiges Wireframe enthält, werden zunächst die vorhandenen Daten in einem Vorverarbeitungsschritt ausgedünnt, sodass nur sichtbare Kanten erhalten bleiben. Als Kante wird jede Verbindung zwischen zwei zum Modell gehörenden geometrischen Eckpunkten (Vertices) definiert. Dazu wird unter anderem der Winkel benachbarter Flächen verglichen und Verbindungskanten, die in einer Ebene liegen, aussortiert. Zur Reduktion der Datenlast werden Kantensegmente, die auf einer Geraden liegen, zu einem Merkmal verbunden.

Das Resultat ist eine Geometrieliste aus Kantenmerkmalen, welche mit einem Deskriptor-Vektor annotiert werden. Dieser enthält Einträge für die Qualitätskriterien, auf denen das Qualitätsmaß der Merkmale berechnet wird. Die Qualitätskriterien betreffen Informationen über die Geometrie, Kamera und Perspektive, sowie Beleuchtung und Material. Die Einträge

befinden sich je nach Definition im Intervall zwischen 0 und 1 oder stellen Binärwerte dar.

Nur diejenigen Merkmale, die eine Mindestqualität erfüllen, werden von der aktuellen Kamerapose aus gerendert und auf ihre Sichtbarkeit getestet. Durch diese erste Filterung wird bereits eine Reduktion des Rechenaufwandes bewirkt. Die übrigen Merkmale werden zur Korrespondenzsuche an den Matcher übergeben, welcher Rückmeldung über den Erfolg gibt und den Qualitätswert dadurch verfeinert. Zusätzlich ist zu jedem Merkmal ein Historienwert gespeichert, der Auskunft über die Entwicklung der Merkmalsqualität in den letzten Bildern der Sequenz gibt und nach jedem Frame aktualisiert wird. Anhand des endgültigen Qualitätswertes nach Einbeziehung aller Kriterien selektiert der Poseschätzer eine minimale Menge an Korrespondenzen, die zur Berechnung der Pose herangezogen werden.

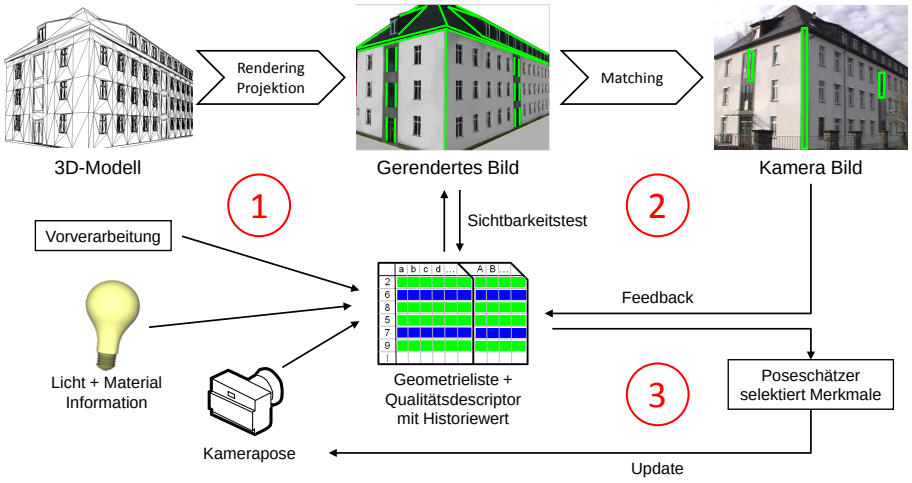


Abbildung 38: Merkmalsmanagement: 1. Geometrieliste nach Vorverarbeitungsschritt, mit einem Deskriptor-Vektor annotiert. Berechnung eines gewichteten Qualitätswertes für jedes Merkmal. 2. Matcher erhält projizierte Geometrie und gibt Qualität der Korrespondenz zurück. 3. Poseschätzer selektiert die besten Merkmale.

6.1 Vorverarbeitung

Beim Laden des Modells wird eine erweiterte, statische Winged-Edge Liste erstellt, die das komplette Wireframe des Modells enthält und den Zugriff auf die benachbarten Flächen jeder Kante erlaubt. Der Winkel benachbarter Flächen wird über die Flächennormalen getestet. Wenn die benachbarten Flächen planar sind, ist die verbindende Kante nicht sichtbar und daher nicht für das Tracking von Nutzen. Diese Kanten werden aus der Liste entfernt und vereinfachen so die Datenlast des Modells. Weiterhin können auch Kanten entfernt werden, deren Flächenwinkel unter einem bestimmten Schwellwert liegt, da diese aufgrund der geringen Intensitätsunterschiede des Lichts auf beiden Flächen von den Methoden der Bildverarbeitung nur schwer zu erkennen sein werden. In einem zweiten Schritt werden Kantensegmente verbunden, die gemeinsame Start- und Endpunkte haben und in der gleichen Richtung angeordnet sind. Diese nicht sichtbaren Segmente entstehen aus den einzelnen benachbarten Geometriedreiecken des Modells und bilden zusammen größere Kanten. Dadurch wird die Datenlast weiter reduziert und die Performanz entsprechend erhöht.

Während der Vorverarbeitung wird zusätzlich die Parallelität der Kanten getestet, indem das Skalarprodukt ihrer Richtungen ausgewertet wird. Parallele Kanten, deren Skalarprodukt etwa 1.0 oder -1.0 beträgt, werden zu Gruppen zusammengefasst. Dies ermöglicht später eine schnelle Berechnung der Qualitätskriterien. Für die Vorbereitung des Sichtbarkeitstests werden den Flächen des Modells zunächst eindeutige Material-Indexfarben im Rotkanal zugeordnet. Dann wird zu jeder Kante eine Liste erstellt, welche die Farben aller Flächen enthält, die nicht mit ihr verbunden sind und sie daher verdecken könnten. Optional kann die Datenmenge des Modells weiter reduziert werden, indem Elemente mit hohem Detailgrad entfernt

werden und so die Redundanz in der Datenstruktur verringern. Dies betrifft vor allem sehr kurze und parallele Kanten (z.B. Fensterrahmen, Abbildung 39). Die verbleibenden Kanten werden in der Geometrieliste gespeichert.

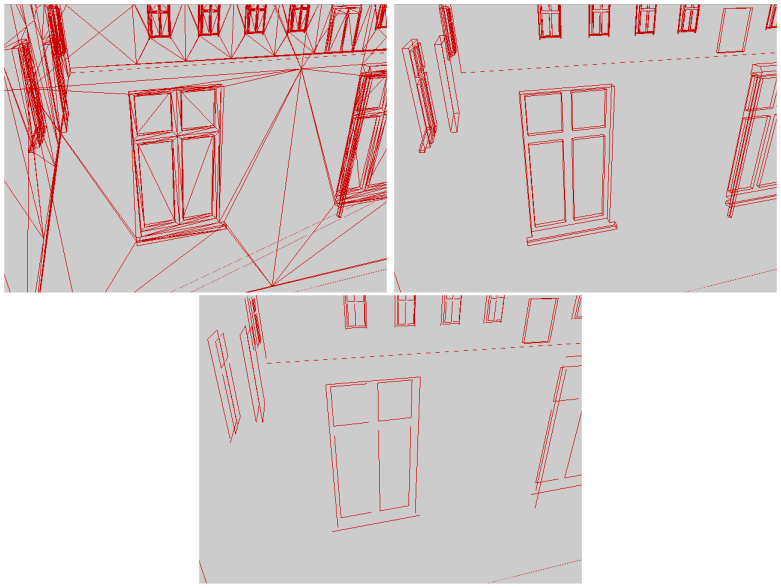


Abbildung 39: Ausdünnung der Daten

6.2 Sichtbarkeitstest

Jede Kante der Geometrieliste wird bei jedem zu rendernden Bild auf die Orientierung der benachbarten Flächen zur Kamera getestet. Kanten, die abgewandte Flächen verbinden, werden nicht betrachtet. Alle Kanten, die der Kamera zugewandte Flächen oder je eine zu- und abgewandte Fläche

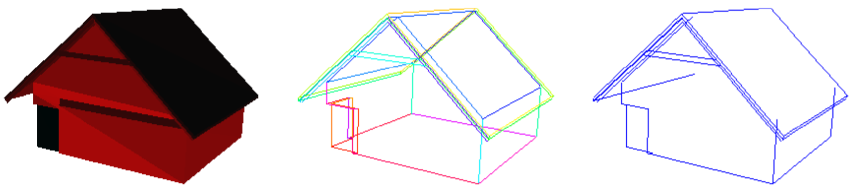


Abbildung 40: Sichtbarkeitstest. Links: Farbindex-Textur. Mitte: Kanten nach Vorverarbeitung. Rechts: Sichtbare Kanten.

verbinden, werden auf Verdeckung getestet. Dazu werden alle sichtbaren Flächen mit ihrer zugeordneten Material-Indexfarbe in eine Textur gerendert. Die Kanten werden nun projiziert und die Farbe für jedes Kantenpixel an der entsprechenden Stelle in der Indextextur ausgelesen. Aus dem Vorverarbeitungsschritt ist eine Liste von Flächen bekannt, welche die Kanten möglicherweise verdecken. Wenn die Farbe nicht in der Verdeckungsliste eingetragen ist, wird eine Zählvariable für diese Kante erhöht. Die Kante wird als sichtbar betrachtet, wenn eine auflösungsabhängige Mindestzahl an Kantenpixeln nicht verdeckt ist (Abbildung 40).

6.3 Länge

Die Stabilität einer Kante während der Korrespondenzsuche ist von ihrer Länge abhängig. Es wäre daher möglich, jeweils die absolute Länge im 3D Raum und ihre 2D Länge in Pixeln zu messen. Ein besserer Ansatz ist die Auswertung der perspektivisch abhängigen Länge in Relation zur Kameraposition. Dazu kann der Öffnungswinkel zwischen Kamera und Kantenmerkmal als Qualitätskriterium herangezogen werden. Vom Kamerazentrum werden zwei Vektoren zu den Start- und Endpunkten der Kante aufgespannt und der einschließende Winkel in Radiant gemessen (Abbildung 41). Der Winkel wird klein, wenn die Kante weit von der Kamera entfernt ist oder die Kantenrichtung annähernd in der Blickrichtung liegt. Die optimale Kantenlänge wird durch den halben vertikalen Öffnungswinkel der Kamera definiert, d.h. die Länge der Kante entspricht nach der Projektion der halben Bildhöhe. Ist der Winkel größer, werden die Werte auf 1 gesetzt, sodass der Eintrag für dieses Qualitätskriterium im Bereich $[0, 1]$ liegt.

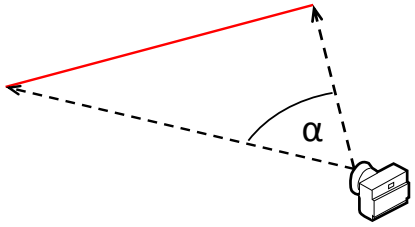


Abbildung 41: Längenkriterium

6.4 Distanz

Kantenmerkmale, die eine geringe Distanz aufweisen, aber parallel ausgerichtet sind, können schwer zu unterscheiden sein und bei der Korrespondenzsuche zu falschen Ergebnissen führen. Daher sollte der Qualitätswert von Kanten mit diesen Eigenschaften verringert werden. Auch bei diesem Kriterium ist ein Qualitätswert, der als perspektivisch abhängige Distanz in Relation zur Kamera bestimmt wird, einem 2D Bildmaß in Pixeln vorzuziehen. In der Vorverarbeitung wurden alle parallelen Kanten gruppiert. Diese werden nun auf ihre Distanz hin überprüft. Dazu werden die zwei Ebenen definiert, welche von den Kanten zusammen mit dem Kamerazentrum aufgespannt werden. Das Kreuzprodukt der Kantenrichtung und des Vektors von der Kamera zu einem Punkt auf der Kante ergibt die Normale der jeweiligen Ebene. Der Winkel zwischen beiden Normalen entspricht dem Öffnungswinkel beider Ebenen aus Sicht der Kamera und wird als Distanz-Qualitätskriterium gespeichert (Abbildung 42). Das Prinzip ist, dass Kanten nach der perspektivischen Projektion in die 2D Bildebene einen höheren Grad an Parallelität aufweisen, je geringer ihr Abstand in 3D ist. Mit zunehmender Distanz im Raum konvergieren die 2D Projektionen in einem Fluchtpunkt. Der optimale Kantenabstand wird als $1/8$ der Bildhöhe definiert. Kanten mit einem geringeren Abstand im 3D-Raum werden aufgrund

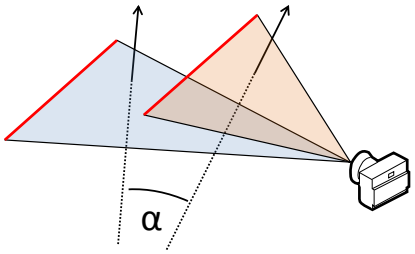


Abbildung 42: Distanzkriterium

ihrer Parallelität als mehrdeutig für die Korrespondenzsuche eingestuft. Distanzen über diesem Wert werden auf 1 gesetzt, sodass der Qualitätswert für das Distanzkriterium im Bereich $[0, 1]$ liegt.

6.5 Silhouette

Es ist anzunehmen, dass Kanten das Modell gut vom Hintergrund trennen können, wenn sie Teil der äußeren Silhouette sind. Insbesondere in Szenarien mit Gebäuden heben sich etwa Dachkanten unter Umständen deutlich gegen den Himmel ab. In Abhängigkeit von der Blickrichtung wird anhand der Flächennormalen getestet, ob eine Kante jeweils eine der Kamera ab- und zugewandte Fläche verbindet. In diesem Fall wird die Qualität durch einen zusätzlichen Eintrag im Deskriptor aufgewertet.

6.6 Richtung

Die Ergebnisse der Korrespondenzsuche sollten am stabilsten sein, wenn Kanten parallel zur Bildebene liegen, d.h. sie ragen nicht in den Raum hinein und haben die optimale sichtbare Länge. Dies lässt sich durch das Skalarprodukt zwischen Blickrichtung der Kamera und dem Richtungsvektor der Kante ausdrücken. Eine senkrechte Anordnung ist am besten geeignet, daher wird das inverse Skalarprodukt als Qualitätswert gespeichert.

6.7 Licht und Material

Die Beleuchtung spielt eine wesentliche Rolle bei der Erkennung von Kanten im Bild. Werden beide angrenzenden Flächen gleich stark beleuchtet, so ist ihre verbindende Kante nur schwer zu erkennen und durch intensitätsbasierte Kantendetektoren schlecht zu identifizieren. Durch die Simulation des momentanen Lichteinfalls, etwa durch Sampling einer HDR-*Lightmap* der Umgebung (Abbildung 43), kann winkelabhängig von Lichteinfallsrichtung und Oberflächennormale die Intensität auf den Umgebungsflächen berechnet werden. Dies geschieht über das Skalarprodukt des Vektors von der Lichtquelle zu den Flächen und den Flächennormalen. Die absolute Differenz der Skalarprodukte beschreibt das Lichtqualitätskriterium im Bereich $[0, 1]$. Auch das Material des Modells kann entscheidend für die Zuverlässigkeit eines Merkmals sein. Wenn das Material schlechte Eigenschaften wie einen hohen Reflektionsgrad hat (z.B. Glasflächen, Fenster), sollte ein zusätzlicher Deskriptor-Eintrag auf 0 gesetzt werden, um die Qualität abzuwerten. Das Testen auf Schattenwurf ist ein weiterer beleuchtungsabhängiger Aspekt bei der Beurteilung von Merkmalen. Diejenigen Merkmale, die im Schatten liegen, werden schlechter oder gar nicht erkannt.

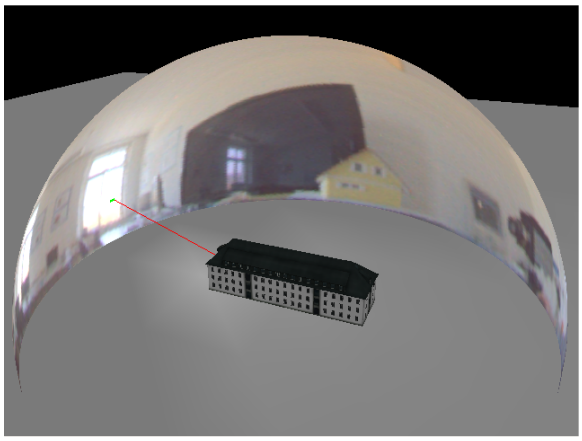


Abbildung 43: Bestimmung der Lichtquelle

6.8 Position

Die Position eines Merkmals in Weltkoordinaten kann Aufschluss darüber geben, ob mit Verdeckungen zu rechnen ist. In einem Outdoor-Szenario mit Gebäuden lässt eine geringe Höhe über dem Boden den Schluss zu, dass sich bewegende Objekte, Personen oder Vegetation die Korrespondenzsuche stören. Das kann vermieden werden, indem die entsprechenden Merkmale von vorneherein ausgeschlossen werden. Ein flexibleres Vorgehen wäre eine dynamische Evaluierung der Merkmalszuverlässigkeit durch Rückmeldung des Erfolgs bei der Korrespondenzsuche. Dies wird unter Einbeziehung der Merkmals-Historie wie im Folgenden beschrieben durchgeführt.

6.9 Korrespondenz-Feedback

Die Korrespondenzsuche soll zu jeder projizierten Modellkante eine entsprechende Kante im Bild finden, die dann als Eingabe für die Poseschätzung dient. Auch hier kann das Wissen aus dem Modell und dem Renderingprozess herangezogen werden, um den Erfolg und die Qualität der Korrespondenzsuche zu bewerten. Da nach der Projektion der Geometriekanten deren Pixellänge im Bild bekannt ist, kann das Verhältnis zur tatsächlich gefundenen Kantenlänge im Bild angegeben werden. Im Idealfall beträgt das Verhältnis 1. Dazu muss der Matcher für jedes Modellmerkmal die Anzahl der im Bild gefundenen Pixel zurückgeben können. Gängige Matchingverfahren wie der Moving Edges Algorithmus [Bou89] und seine Weiterentwicklungen, etwa [WVS05], tasten das Bild auf orthogonalen Suchlinien ab, die entlang der projizierten Modellkante gesetzt werden und sind daher nicht in der Lage, eine Rückmeldung über die genaue Pixellänge der Kante im Bild zu geben. Um dies zu realisieren, wurde ein shaderbasierter Kantenmatcher entwickelt. Auf einem Canny-Kantenbild der Kamera wird der

Shader für jedes Modellmerkmal aufgerufen und generiert mögliche, durch Bewegung verursachte, Verschiebungen der Start- und Endpunkte im Bild. Zwischen allen erzeugten Punkten werden ihre verbindenden Bildlinien berechnet und für jedes Pixel der Linien überprüft, ob ein entsprechendes Kantenpixel auf dem Eingabebild zu finden ist. Die Summe der gezählten Pixel wird über eine Rückgabetextur vom Merkmalsmanagement ausgelesen. Jedes Pixel der Rückgabetextur speichert die Ergebnissumme von genau einer Kante, wobei anhand der Texturkoordinaten die jeweilige Kante adressiert wird. Über das Texturmaximum kann anschließend auf die Kantenkorrespondenz im Bild geschlossen werden. Eine genaue Beschreibung des Matching-Shaders findet sich in Kapitel 7.

6.10 Konfiguration

Nachdem die Korrespondenzen ermittelt wurden, kann eine weitere Analyse ihrer geometrischen Anordnung vorteilhaft sein. Der Poseschätzer erfordert gewisse Kriterien bezüglich bekannter kritischer Konfigurationen der Korrespondenzen, die keinen Beitrag zur Poseschätzung leisten oder gar mehrdeutige Ergebnisse verursachen können. Unter Betrachtung von Distanzen und Winkeln zwischen Flächen und Kanten können diese kritischen Konfigurationen erkannt und vermieden werden.

So sollten Korrespondenzen, von denen drei oder mehr parallel liegen oder sich in einem gemeinsamen Punkt schneiden, vermieden werden. Die Parallelität der Korrespondenzen wird über das Skalarprodukt ihrer Richtungsvektoren bestimmt. Wenn alle Ergebnisse nahe 1 sind, werden die Korrespondenzen als kollinear verworfen. Ein gemeinsamer Schnittpunkt wird festgestellt, indem die Korrespondenzen paarweise auf einen Schnittpunkt geprüft werden. Gibt es mehr als einen Schnittpunkt und liegt ihr

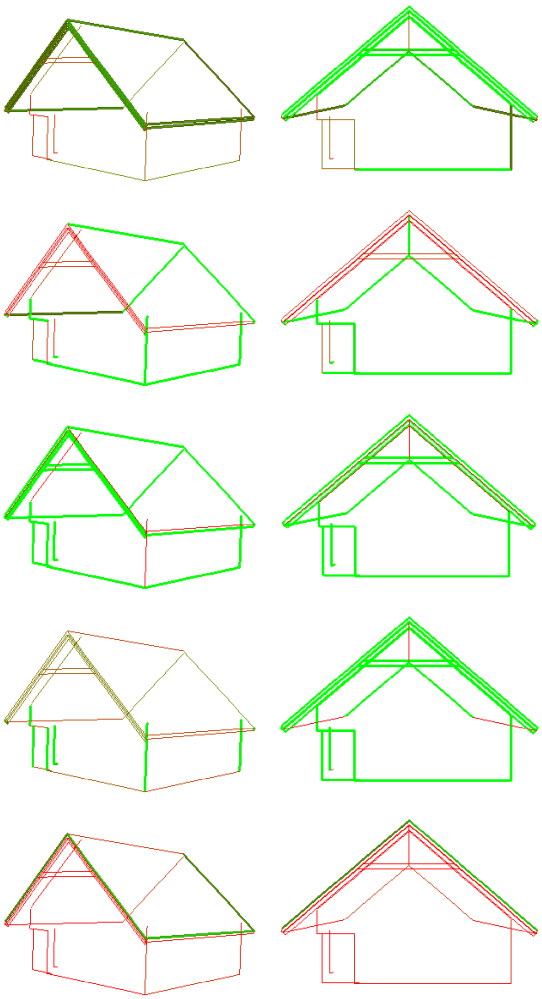


Abbildung 44: Visualisierung einiger Qualitätskriterien aus zwei Ansichten. Grün ist die beste Bewertung. Länge, Distanz, Silhouette, Richtung, Beleuchtung.

Abstand unterhalb eines Schwellwertes, dann werden die Korrespondenzen ebenfalls verworfen. Weitere Ausführungen finden sich in Kapitel 5.

6.11 Historie

Um eine Aussage über die Gesamtqualität eines Merkmals zu treffen, ist es ebenfalls sinnvoll, die zeitliche Entwicklung der Qualität zu betrachten. Ein Merkmal ist dann gut für das Tracking geeignet, wenn es über einen

Mindestzeitraum stabil getrackt werden konnte. Hierbei werden die Qualitätswerte der letzten 10 Bilder in einem Historienwert-Array gespeichert, welches nach jedem Bild aktualisiert wird. Diese Werte werden zu einem Durchschnittswert verrechnet, welcher dann in die Berechnung der Merkmalsqualität im aktuellen Bild eingeht. Das Merkmalsmanagement lernt so, welche Merkmale sich nicht stabil verhalten und dementsprechend abgewertet werden sollten. Beim Start des Trackings ist der Historienwert nicht gesetzt und trägt folglich nicht zur Qualitätsberechnung bei.

6.12 Berechnung der Qualität

Nachdem alle Werte für die einzelnen Qualitätskriterien gesammelt wurden (siehe Visualisierung Abbildung 44), wird die Gesamtqualität in zwei Schritten berechnet. Vor der Projektion werden die Einträge des Deskriptor-Vektors gewichtet gemittelt, um einen Qualitätswert im Bereich $[0, 1]$ zu erhalten. Dazu wird das Skalarprodukt aus Deskriptor-Vektor und Gewichtsvektor berechnet. Das Ergebnis wird mit der Summe aller Gewichte normalisiert (Abbildung 45). Die optimale Gewichtung der Qualitätskriterien entsprechend ihres Einflusses auf das Trackingergebnis wird im Zuge von Tests ermittelt.

$$\vec{Q} := \begin{pmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{pmatrix}, \quad \vec{W} := \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{pmatrix}$$

$$q_{overall} = \frac{\left(\vec{Q}\right) \cdot \left(\vec{W}\right)}{\sum_{i=1}^{n} w_i}$$

Abbildung 45: Berechnung des Qualitätswertes aus den gewichteten Einträgen des Deskriptor-Vektors: $q_{1..n}$ Qualitätskriterien, $w_{1..n}$ Gewichte.

Eine Untermenge der Merkmale, die gemäß dem berechneten Qualitätswert eine Mindestqualität erreichen, wird projiziert und auf Sichtbarkeit getestet. Diese erste Filterung führt bereits zu einer nennenswerten Reduktion der Rechenzeit. Nach der Korrespondenzsuche wird im zweiten Schritt der abschließende Qualitätswert unter Einbeziehung des gewichteten Feedbacks über das Korrespondenzergebnis verfeinert. Jedes Merkmal der Geometrieliste ist nun mit einem Gesamt-Qualitätswert versehen und die Liste kann in Prioritätsklassen geordnet werden. Aus dieser geordneten Liste selektiert der Poseschätzer eine minimale, aber qualitative Menge von Merkmalen zur Berechnung der Pose.

6.13 Ergebnisse

Das Merkmalsmanagement wurde auf synthetischen und realen Kamerabildern getestet. In Indoor und Outdoor Szenen wurden sowohl simple als auch komplexe Modelle unter variierenden Beleuchtungsverhältnissen eingesetzt. Die Auflösungen betrugen 640 x 480 und 1280 x 720 Pixel. Die initiale Kamerapose wurde beim Start des Trackings als bekannt vorausgesetzt, ebenso die intrinsischen Kameraparameter. Der Rotationsfehler wird in Grad angegeben und der Translationsfehler in Objekteinheiten (die Dimension des Objekts ist 2). Zur Berechnung der Kamerapose aus den Kantenkorrespondenzen wird ein nichtlinearer Levenberg-Marquardt Optimierer eingesetzt.

Es gilt herauszufinden, welche der Qualitätskriterien für das Tracking von besonderer Bedeutung sind und wie sie daher zu gewichten sind. In 2200 Testläufen wurde der Einfluss der Kriterien auf das Trackingergebnis untersucht. Im Merkmalsmanagement wurden dazu alle möglichen Kombinationen der Qualitätskriterien mit variierten Gewichtungen getestet und

die Präzision der berechneten Pose anhand des durchschnittlichen Fehlers aufgezeichnet. Das Ergebnis ist der optimale Gewichtungsvektor, welcher die robusteste Pose liefert (Tabelle 3). Länge, Distanz, Silhouette und Licht sollten mit Faktor 3 gewichtet werden.

Länge	3
Distanz	3
Silhouette	3
Richtung	1
Licht + Material	3
Historie	4
Korrespondenz-Feedback	2

Tabelle 3: Optimale Gewichtung der Qualitätskriterien.

Wie vermutet, ist der Einfluss von Silhouettenkanten hoch, das Lichtkriterium allerdings nicht so bedeutend, wie erwartet. Ein positiver Einfluss von zur Bildebene planaren Strukturen konnte nicht verifiziert werden. Die Gewichtung des Historienwertes ist szenenabhängig. Besonders wenn wenige lange Kanten vorhanden sind und starkes Auftreten von Verdeckungen zu beobachten ist, verbessert eine hohe Gewichtung der Historie mit dem Faktor 4 das Ergebnis. In einer Szene mit sehr langen, gut zu trackenden Kanten, hat sich eine Gewichtung von 2 als ausreichend herausgestellt. Zukünftige Untersuchungen sollten sich daher auf den Bereich der Szenenanalyse und automatisierten Gewichtsberechnung konzentrieren. Die Rückmeldung der Korrespondenzqualität vom Matcher hat wie erwartet großen Einfluss auf die Vorhersage stabiler Merkmale und wird daher mit doppeltem Gewicht auf den Durchschnittswert der vorherigen Kriterien gerechnet.

Abbildung 46 zeigt die Kantenmerkmale mit und ohne Merkmalsmanagement auf vier Testszenen. In Abbildung 47 und 48 sind der Translations- und Rotationsfehler sowie die Berechnungszeit für eine Szene mit und ohne

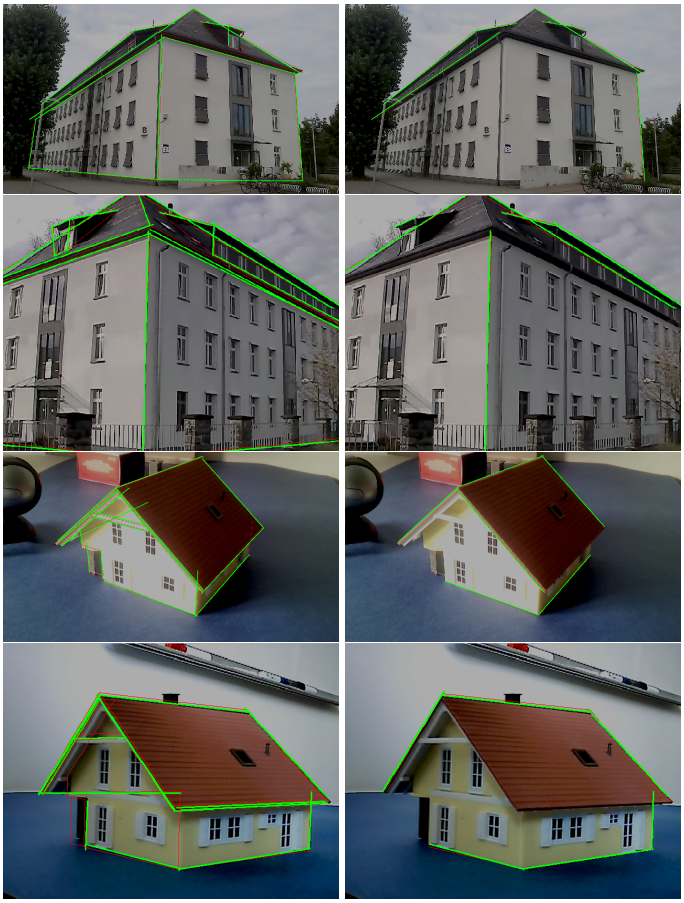


Abbildung 46: Trackingszenen mit ausgewählten Merkmalen (grün). Links: Ohne Management. Rechts: Merkmalsmanagement.

Management gelistet. Im Vergleich zum Tracking ohne Merkmalsmanagement, also der Verwendung aller Modellmerkmale ohne Filterung, konnte das Ergebnis der Pose verbessert werden. Es tritt weniger Rauschen auf und die Pose konnte auch in schwierigen Szenen erfolgreich getrackt werden. Zusätzlich ist im Schnitt über alle Testszenen eine Beschleunigung um den Faktor 2 zu beobachten. Das frühe Filtern einer minimalen Menge an Merkmalen, insbesondere schon vor der Projektion, führte teilweise zur Reduktion der zu verarbeitenden Datenmenge auf 10% der ursprünglichen Merkmale, was insbesondere bei der Anwendung auf mobilen Geräten

von Vorteil ist. Während auf hochaufgelösten HD-Videos mit dem nativen Trackingansatz keine interaktive Framerate erreicht werden konnte, wurde durch Selektion der 6 besten Merkmale echtzeitfähiges Tracking bei gleichzeitiger Verbesserung der Präzision ermöglicht.

6.14 Fazit

Die Ergebnisse zeigen, dass die Einbeziehung von Wissen über Modell, Rendering und Umgebung in die einzelnen Komponenten des Trackingprozesses, sowie die Evaluierung der Merkmale anhand der vorgeschlagenen Qualitätskriterien die Verwendung von beliebig komplexen Modellen ermöglicht, welche nicht speziell für den Einsatz in einem Trackingszenario erstellt wurden. Solche Modelle sind in den typischen Anwendungsfeldern der Augmented Reality, wie Lern-, Konstruktions- und Wartungsszenarien, häufig vorhanden. Auch im touristischen Kontext wächst die Anzahl der online verfügbaren Stadt- und Gebäudemodelle stetig (z.B. *Google Earth*), was in naher Zukunft allgegenwärtige AR-Anwendungen ermöglichen wird. Aus technischer Sicht ist die Untersuchung einer Kombination von *Analyse-durch-Synthese* Techniken mit einem zweiten markerlosen Trackingparadigma wie dem Ansatz des Simultaneous Localisation and Mapping (SLAM) Verfahrens interessant. Es ist durchaus denkbar, die mit diesem Verfahren rekonstruierten Elemente in ein vorhandenes 3D Modell zu integrieren und dieses so zu erweitern. Umgekehrt kann der Rekonstruktionsprozess durch die Überprüfung anhand der Modelldaten validiert und so zu einer höheren Sicherheit beigetragen werden.

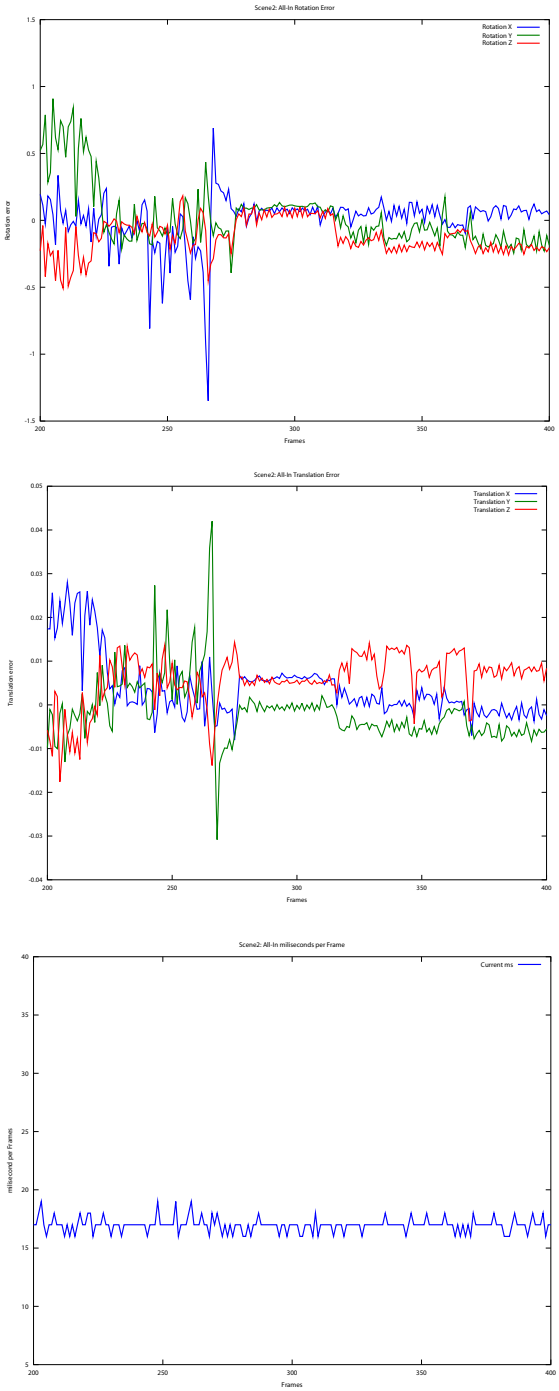


Abbildung 47: Posefehler ohne Management. Rotation, Translation, Zeit.

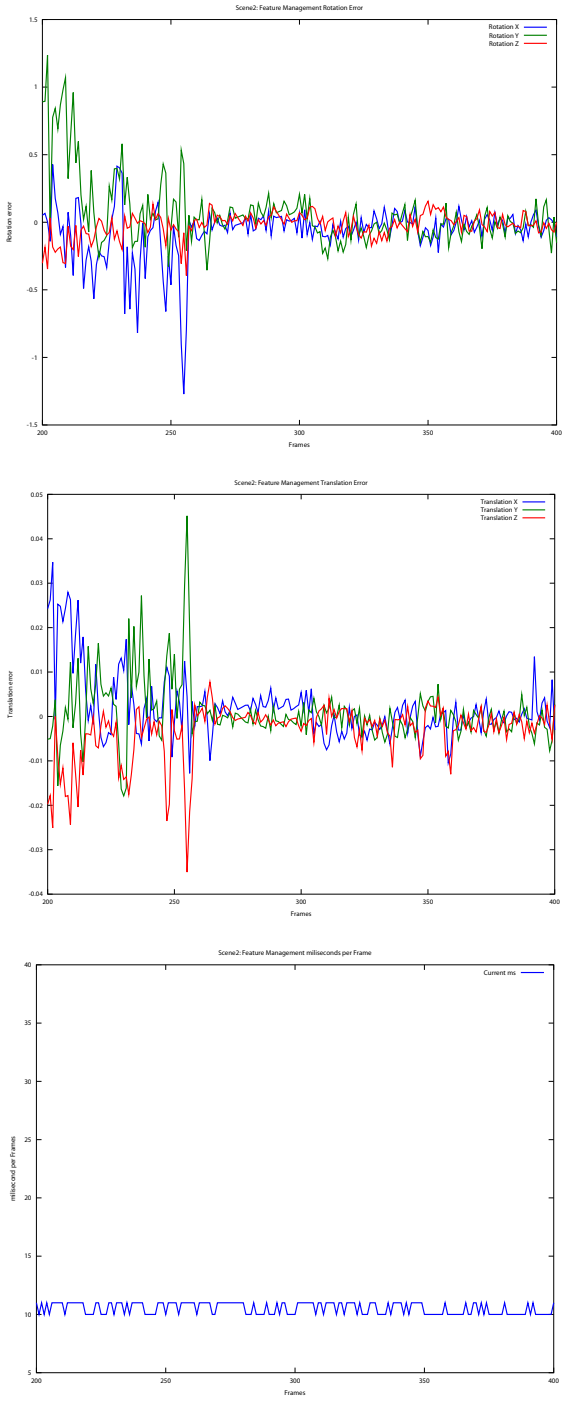


Abbildung 48: Posefehler mit Merkmalsmanagement. Rotation, Translation, Zeit.

7 Korrespondenzsuche

Das Problem der Posebestimmung basiert auf der Suche von 3D-2D Korrespondenzen zwischen Modellmerkmalen und Bildmerkmalen (Matching). Das Ziel ist es, zu Punktmerkmalen, Kanten oder höheren Strukturen des Modells Entsprechungen im Kamerabild zu finden und die Distanzen zwischen den projizierten Modellmerkmalen und den Bildmerkmalen zu minimieren. Die Suche nach diesen Korrespondenzen ist entscheidend für eine gute Poseschätzung. Fehlerhafte Korrespondenzen führen zu starken Abweichungen von der korrekten Pose oder sogar zum Verlust des Trackings.

Ausgehend von den Anforderungen der in Kapitel 6 beschriebenen Bewertungskriterien für Merkmale, soll das eingesetzte Matchingverfahren eine Aussage über die Qualität der Korrespondenz zulassen. In diesem Kapitel wird dazu eine Matchingmethode für Linienmerkmale im modellbasierten Kameratracking vorgestellt, die auf der *20th International Conference on Computer Graphics, Visualization and Computer Vision* veröffentlicht wurde [SHM12]. Unter Nutzung eines Shaders und des Modellwissens über Geometrie und Perspektive soll die Korrespondenzsuche zwischen Modellmerkmalen und Bildmerkmalen verbessert und Qualitätskriterien beschrieben werden. Die GPU wird eingesetzt, um die jeweils besten korrespondierenden Bildlinien zu den Kanten eines gegebenen 3D Modells zu finden. Jede Kante wird dazu mehrere Male mit leichten Verschiebungen von der letzten Kamerapose aus ins Bild gerendert. Der Shader zählt die Anzahl der darunter liegenden Pixel in einem mit dem Canny-Operator gefilterten Kamerabild. Die Korrespondenz wird durch diejenige gerenderte Kante mit der höchsten Übereinstimmung beschrieben. Die Rückgabe der Pixelanzahl kann über den Einsatz von Occlusion Queries erfolgen. Ein anderer vorgeschlagener Ansatz ist ein weiter entwickelter Shader, der das

Ergebnis über eine Textur ausgibt, was die Anzahl der Renderdurchgänge reduziert und performanter ist. Der Matching-Shader ist nicht auf die Arbeit mit Linien beschränkt, sondern kann auch auf andere Strukturen erweitert werden.

7.1 Verwandte Arbeiten

Das Problem der Korrespondenzsuche zwischen Bildern tritt nicht nur beim Bestimmen der Kamerapose auf, sondern darüber hinaus auch in Anwendungen der Objekterkennung und Bildregistrierung, etwa bei der Überlagerung von Bildern verschiedener Quellen und Modalitäten. Mögliche Ansätze lassen sich in folgende Kategorien einordnen. Intensitätsbasierte Ähnlichkeitsmaße betrachten das gesamte Bild oder Bildausschnitte, indem die Werte der korrespondierenden Pixel in den Bildern direkt verglichen werden. Als Maße können dabei die Summe der quadratischen Distanzen der Intensitätswerte oder die Kreuzkorrelation dienen. Eine darüber hinaus weit verbreitete Methode ist es, visuelle Bildmerkmale wie Punkte oder Linien in den Bildern abzugleichen. Dies setzt jedoch die weiteren Arbeitsschritte der Merkmalsdetektion und Merkmalsdeskription voraus.

Detektion und Korrespondenzsuche von Punktmerkmalen ist ein lange entwickeltes Forschungsgebiet. Zunächst wird im Bild nach interessanten Punkten wie Kantenschnittpunkten oder Ecken gesucht. Ist deren Lage bekannt, können die Interessenspunkte anhand ihrer Pixelnachbarschaft beschrieben werden, indem ein Deskriptor-Vektor der Intensitäten ihrer Nachbarschaft erstellt wird. Weitere Informationen zur robusten Beschreibung von Merkmalen können die Skalierung und Orientierung sein, wie sie in SIFT [Low99] und SURF [BETG08] verwendet werden. Die Korrespondenzsuche erfolgt über den Vergleich der Deskriptor-Einträge. Da dieses

Vorgehen sehr rechenintensiv ist, existieren GPU-beschleunigte Implementierungen, wie die des KLT-Trackers [ST94] durch [SFPG06].

Dem gegenüber werden Kantenmerkmale seltener eingesetzt, bieten sich jedoch bei einem Szenario mit Gebäudemodellen an. Zum Erkennen der Kanten im Bild kommen Bildverarbeitungsfilter wie der Sobel-Operator oder weiter entwickelte Techniken wie der Canny-Algorithmus [Can86] zum Einsatz. GPU-Implementierungen dieser Filter werden im Kontext eines Partikelfilter-Frameworks in [KM06] und [BC12] gezeigt. Im einfachsten Fall kann die Korrespondenz im Parameterraum hergestellt werden, indem die Lage der Modellkante und der Bildlinie über ihren jeweiligen Abstand zum Bildursprung und den Winkel zu einer Bildachse beschrieben und anschließend verglichen wird. Dies erfordert jedoch eine Parametertransformation wie etwa das Verfahren zur Geradenerkennung von Hough [DH72], was wiederum sehr aufwändig ist. Daher wird die Korrespondenzsuche hauptsächlich distanzbasiert im Bildraum realisiert, indem der euklidische Abstand zwischen den projizierten Modellkanten und den entsprechenden Gradienten im Bild minimiert wird. Als einfaches Abstandsmaß kann die Projektion der Start- und Endpunkte der Modellkante auf die Bildlinie dienen. In [Low91] wird der senkrechte Abstand zwischen Modellkante und Segmenten der Bildlinien betrachtet und in [Low92] mit der Orientierung kombiniert.

Eine weitere mögliche distanzbasierte Methode zur Korrespondenzsuche bei Linien ist die Anwendung des Moving-Edges Algorithmus [Bou89]. Dieser Ansatz wurde in einigen Tracking-Frameworks eingesetzt, wie etwa [HS90],[DC02],[CMC03] oder [VLF04] um nur einige zu nennen. Die ins Bild projizierte Modellkante wird dabei mit Kontrollpunkten abgetastet und durch diese Punkte werden orthogonale Suchlinien in beide Richtun-

gen aufgespannt. Entlang dieser Normalen wird das Gradientenmaximum im Bild berechnet und die Distanz zwischen dem 3D-Kontrollpunkt und dem gefundenen 2D Bildpunkt minimiert. Die Auswertung multipler Hypothesen kann die Stabilität beim Auftreten mehrerer Maxima verbessern [VLF04][WVS05].

7.2 Der Matching-Shader

7.2.1 Aufbau

Modellbasierte Trackingansätze nutzen ein 3D Modell des Referenzobjekts, wobei in der Regel aus diesem durch das Rendern eines Bildes Modellkanten erstellt werden. Unter Einsatz von Filtern, wie etwa dem Sobel-Operator, werden Bildkanten erkannt und diese auf das 3D Modell zurück projiziert, um ihre entsprechenden 3D Koordinaten zu erlangen. Stattdessen werden im vorgestellten Ansatz Kandidaten für die Kantenkorrespondenzen direkt aus der Modelldatenstruktur selektiert und ein Sichtbarkeitstest durchgeführt. Der Bildverarbeitungsprozess auf dem gerenderten Bild entfällt somit. Der Vorteil einer Kandidatenliste ist, dass die Ergebnisse der Korrespondenzsuche nach ihrer Qualität sortiert und gewichtet werden können. Auf Details hierzu wurde in Kapitel 6 eingegangen.

Zu den Kandidaten der 3D Modellkanten sollen korrespondierende 2D Linien im Bild gefunden werden. Das Kamerabild wird vorab mit dem Canny-Algorithmus gefiltert, sodass natürliche Strukturen als Binärbild ausgegeben werden. Die Modellkanten werden projiziert und mit dem Matching-Shader gerendert. Für jedes Pixel der Modellkante, das während des Rendervorgangs gezeichnet wird, erfolgt ein Aufruf des Pixelshaders, welcher den Intensitätswert an der entsprechenden Pixelposition im Canny-Eingabebild ausliest. Wird ein schwarzes Linienpixel gelesen, so reagiert

der Pixelshader mit der Ausgabe eines Farbwertes. Andernfalls verwirft der Shader das aktuelle Pixel mit dem *discard*-Befehl und der Renderpass wird abgebrochen. Das Konzept ist in Abbildung 49 dargestellt und der entsprechende Shadercode der verwendeten *OpenGL Shading Language* in Auflistung 1 aufgeführt.

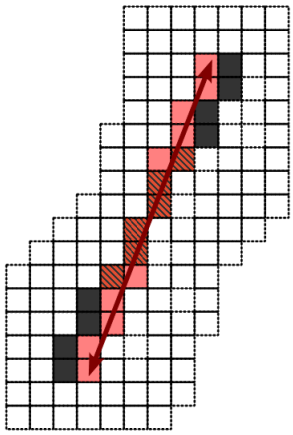


Abbildung 49: Gerenderte Kante (rot), Pixelkante im Bild (schwarz) und gemeinsame Pixel, die gezählt werden (schraffiert)

Die Anzahl der erfolgreichen Renderaufrufe entspricht nun der Anzahl an gezählten Linienpixeln im Bild. Das Auslesen des Ergebnis kann durch das Erstellen einer *Occlusion Query* während des Renderns realisiert werden (Kapitel 7.2.3). Die Anzahl der gezählten Pixel gibt die Wahrscheinlichkeit an, mit der die gefundene Bildlinie im Canny-Bild eine Korrespondenz zur gerenderten Modellkante ist, da die Pixelanzahl ein direktes Maß für die Länge der Bildlinie darstellt. Idealerweise würde das Ergebnis der Pixelzählung der Modellkantenlänge entsprechen.

```
1 //Vertex shader
2 void main()
3 {
4     gl_Position = ftransform();
5     gl_TexCoord[0] = gl_MultiTexCoord0;
```

```
6  }
7
8  //Fragment Shader
9  uniform sampler2D cannyImage;
10 vec3 color;
11 void main()
12 {
13     color = texture2D(cannyImage, gl_TexCoord[0].st).xyz;
14     if(( color != vec3(1.0,1.0,1.0) ))
15         gl_FragColor = vec4(0.0,0.0,0.0,1.0);
16     else
17         discard;
18 }
```

Auflistung 1: GLSL Code des Matching-Shaders

Die Verwendung von weiteren Informationen aus dem Modell kann dazu beitragen, die Korrespondenzsuche zu verbessern. Aus der bekannten Länge der gerenderten Modellkante kann auf die zu erwartende Bildlinienlänge geschlossen und ein Schwellwert für die minimal akzeptierte Anzahl an gezählten Bildpixeln definiert werden. Erfüllt die Bildlinie dieses Kriterium nicht, so wird sie als Korrespondenz zurückgewiesen. Im folgenden Abschnitt werden weitere Kriterien für die Bewertung der Korrespondenzen erläutert.

7.2.2 Erzeugung der Sample-Kanten

Da bei dem hier beschriebenen Tracking der Kamerapose von geringen Bewegungen der Kamera zwischen den einzelnen Bildern ausgegangen wird, müssen die Modellkanten bei der Suche nach Korrespondenzen bezüglich ihrer Position und Orientierung im Bild variiert werden, um die vermutete Bewegung der Kamera zu simulieren. Dies erfolgt durch das

Sampeln einiger neuer Bildkanten nach der Projektion der Modellkante ins Bild. Für jede Modellkante sind Start- und Endpunkt bekannt. Um diese werden in einem Fenster jeweils neue Punkte mit einem Pixeloffset generiert und die Startpunkte mit allen Endpunkten zu neuen Kanten verbunden. Ausgehend von einem 3x3 Fenster würden beispielsweise jeweils 8 neue Punkte erzeugt, die zusammen mit den ursprünglichen Start- und Endpunkten der Modellkante in 81 Sample-Kanten resultieren. Abbildungen 50 und 51 zeigen Beispiele für die Erzeugung von Sample-Kanten. Das Sampling findet im 2D Bildraum statt, die 3D Modelldaten werden dabei nicht verändert. Die Wahl des Offsets zur Erzeugung der Sample-Kanten hängt von der Genauigkeit und Verarbeitungsgeschwindigkeit ab. Ein höherer Offset deckt größere Bewegungen der Merkmale im Bild ab, da mehr Kanten in weiteren Distanzen und Winkeln um die Referenzkante erzeugt werden. Der Nachteil ist ein Performanzverlust, besonders unter dem Einsatz von Occlusion Queries.

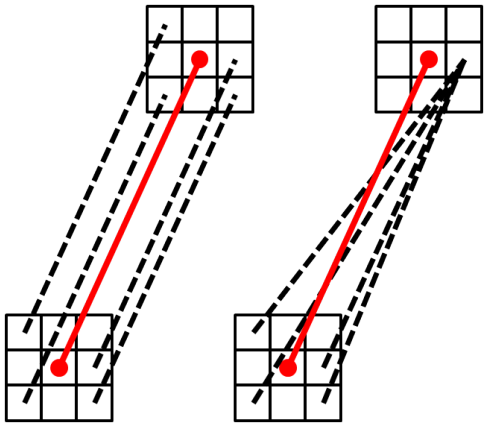


Abbildung 50: Erzeugte Sample-Kanten (gestrichelt) für eine gegebene Modellkante auf einem 3x3 Fenster

Für alle diese Sample-Kanten gibt der Matching-Shader die Anzahl der gezählten Bildpixel zurück, wie in Abschnitt 7.2.1 beschrieben. Die Kante mit dem höchsten Ergebnis ist Kandidat für die beste Korrespondenz. Neben

Abbildung 51: Projizierte Sample-Kanten im Bild. Zur besseren Sichtbarkeit werden nur einige Kanten dargestellt

der bereits erwähnten Pixellänge ist es auch möglich, die Distanz zwischen den Ergebnissen der Pixelanzahl jeder Kante als Qualitätskriterium heranzuziehen. Ähnliche parallele Linien im Bild resultieren in annähernd gleichen Rückgabewerten der Pixelanzahl. Sticht die Kante mit dem besten Ergebnis nicht deutlich hervor, können sie daher als uneindeutige Merkmale erkannt und als Korrespondenz zurückgewiesen werden.

Des Weiteren bietet das Modellwissen auch Informationen über die Tiefe der 3D Kante, welche zur Verbesserung des Samplings verwendet werden können. Weit von der Kamera entfernte Bewegungen führen im Bild zu geringeren Änderungen der Pixelpositionen, während dieselbe Bewegung sich dicht vor der Kamera in einem starken Versatz der Pixel äußert. Anhand der bekannten Tiefen von Start- und Endpunkt der Modellkanten können unterschiedliche Größen für die Sample-Fenster definiert und so eine von der Perspektive abhängige Erzeugung der Sample-Kanten durchgeführt werden. Wenn die Tiefenwerte von Start- und Endpunkt einer Kante sich stärker voneinander unterscheiden, als ein definierter Schwellwert, dann werden für den nahen Punkt mehr Samples generiert, als für den entfern-

ten. Dadurch wird die Gesamtmenge an zu rendernden Sample-Kanten reduziert.

7.2.3 Occlusion Query Management

Die Rückgabe der gezählten Pixel aus dem Matching-Shader kann durch den Aufruf einer *Occlusion Query* erfolgen. Das Verfahren ermöglicht es, die Anzahl der erfolgreichen Ausführungen eines Pixelshaders pro Renderdurchgang von der Grafikhardware in eine Variable zu schreiben. Sollen die Renderaufrufe kontinuierlich Ergebnisse liefern, so erfordert dies eine Management-Routine, die mehrere Occlusion Queries schnell hintereinander ausführen kann. Da in dem vorgestellten Ansatz eine Liste von Modellkanten gerendert wird, muss für jede von ihnen eine separate Occlusion Query durchgeführt werden. Die Grafikhardware limitiert jedoch die Anzahl an Queries, die eine effiziente Rückgabe von Ergebnissen ermöglichen. Der Versuch, das Zählergebnis sofort nach jedem Renderdurchgang anzufordern, würde zu einer zeitlichen Blockierung (Stall) der CPU führen [Fer04].

Da in Abhängigkeit von der eingesetzten Grafikhardware weniger Occlusion Queries sequentiell abgearbeitet werden können, als Modellkanten zu rendern sind, muss die Aufgabe in mehrere Durchläufe aufgeteilt werden. Zunächst wird der maximal mögliche Satz an n Query-Objekten erzeugt. Nun kann ein Teil der Modellkanten gerendert werden, bis die maximale Anzahl an verfügbaren Queries erreicht ist. Danach werden alle Ergebnisse der n Queries ausgelesen, bevor sie mit dem nächsten Block der noch verbleibenden Kanten erneut aufgerufen werden können. Das Ergebnis mit dem höchsten Rückgabewert wird gespeichert. Alternativ kann eine geordnete Liste aus den Ergebnissen erstellt werden, um einen späteren Vergleich

zu ermöglichen. Die absolute Anzahl an Queryaufrufen wird ebenfalls gezählt und der Vorgang endet, wenn alle Kanten gezeichnet wurden (Siehe Auflistung 2).

```
1  create n query objects
2  generate sample edges
3  enable shader
4  load canny texture
5  disable color and depth buffer write
6  while( query count != number of edges ){
7      for n queries{
8              start query
9              render edge
10             end query
11             query count++
12     }
13     for n queries{
14             retrieve result
15             if query result > last query
16                     save result
17     }
18 }
19 enable color and depth buffer write
20 disable shader
```

Auflistung 2: Occlusion Query Management

7.2.4 Werterückgabe über Textur

Die Ergebnisse zeigen, dass die Verwendung eines einfachen Shaders mit Occlusion Query nicht sehr performant ist, wenn viele Kanten für die Korrespondenzsuche herangezogen werden (Abschnitt 7.3). Daher wurde der Matching-Shader weiterentwickelt, um eine größere Anzahl an Kanten in

kurzer Zeit bearbeiten zu können. Der erweiterte Ansatz basiert auf der Erzeugung der Sample-Kanten innerhalb des Shaders und dem Auslesen einer Ergebnistextur. Dadurch wird die Vorberechnung der Sample-Kanten auf der CPU und das Ausführen von Occlusion Queries vermieden. Das transferieren der gesamten Berechnung auf die Grafikhardware reduziert die Anzahl der nötigen Renderdurchgänge. Anstatt jede Sample-Kante einzeln zu rendern, wird nun für jede Modellkante einmal die Ausgabetextur gerendert und die Sample-Kanten intern im Shader berechnet. Somit wird das Problem der blockierten CPU während des Wartens auf das Ergebnis der Occlusion Query gelöst. Die Ergebnisse der durch den Shader gezählten Pixel aller Samples, die aus einer Modellkante erzeugt wurden, können in einem Texture Read-Back erlangt werden.

Dazu werden dem Shader neben der Canny-Textur des Kamerabildes nun auch die Koordinaten der projizierten 2D Start- und Endpunkte der Modellkante sowie der Offset für die Generierung der Sample-Kanten als Eingabevariablen übergeben. Der Shader berechnet die neuen Sample-Punkte in einem Fenster mit dem gegebenen Offset um den Start- und Endpunkt der Modellkante.

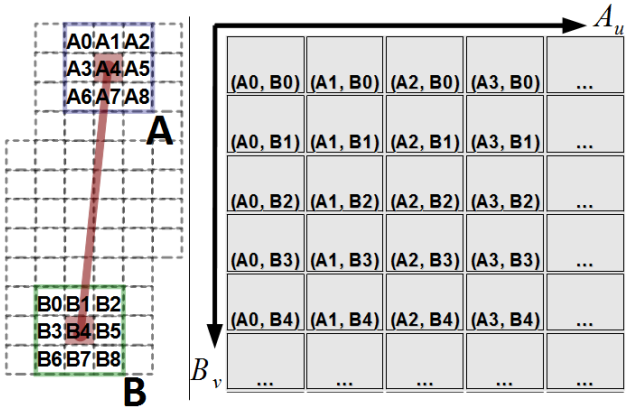


Abbildung 52: Shader Target Texture Organisation

Jeder neue Start- und Endpunkt wird dann zu einer Sample-Kante verbunden. Dies geschieht durch die Berechnung des Bresenham Linienalgorithmus [Bre65] zwischen den Start- und Endpunkten im Shader. Die aus dieser Berechnung hervorgehenden Pixelkoordinaten der Linienpixel werden zum Zugriff auf die Canny-Textur an der entsprechenden Pixelposition verwendet. Die Anzahl der gezählten Pixel wird als Rückgabewert in eine Rendertarget Textur geschrieben. So können in einer Ausgabetextur alle Ergebnisse der Sample-Kanten zu einer Modellkante gespeichert werden. Die Ausgabetextur hat die Größe aller Sample-Kanten, beispielsweise werden bei 9 Sample-Punkten in einem 3x3 Fenster 81 Kanten erzeugt, zu denen Rückgabewerte anfallen. Daher muss die Textur der Größe 9x9 für 81 Ergebniseinträge entsprechen.

Abbildung 52 zeigt die Organisation der Rückgabetextur. Jede Spalte und ihre zugeordnete u-Texturkoordinate entsprechen einem Startpunkt A. Analog bezeichnet jede Reihe und ihre v-Texturkoordinate einen Endpunkt B. Jedes Pixel der Textur ist so für das Ergebnis einer Sample-Kante adressiert. Dem Pixelshader ist die Texturkoordinate (u,v) bekannt, an die er aktuell einen Wert schreiben kann. Anhand dieser Information wird der Offset auf die übergebenen Start- und Endpunkte der Modellkante angewendet und entsprechende Sample-Punkte erzeugt. Jeder Aufruf eines Pixelshaders berechnet somit in Abhängigkeit seiner Schreibposition genau eine Sample-Kante, wie im Folgenden erläutert wird.

Wird der Offset von den übergebenen Koordinaten (x,y) des Startpunktes A der Modellkante subtrahiert, erlangt man den ersten gesampelten Startpunkt A0 mit den niedrigsten Koordinaten im Sample-Fenster. Ausgehend von diesem Punkt kann man durch die Addition von u Modulo der Fenstergröße s die x-Koordinate und durch die Addition von u geteilt durch

s die y-Koordinate des neuen Sample-Punktes berechnen:

s = 2*offset+1
sampleStartX = modelStartX - offset + (u % s)
sampleStartY = modelStartY - offset + (u / s)
sampleEndX = modelEndX - offset + (v % s)
sampleEndY = modelEndY - offset + (v / s).

Dasselbe Vorgehen mit dem Endpunkt B und der Texturkoordinate v resultiert in dem entsprechenden Sample-Punkt. Zwischen diesen Punkten wird nun die Bresenhamlinie berechnet und das Ergebnis der Pixelanzahl an die aktuelle Texturposition geschrieben. Nachdem der Shader die Textur für die Modellkante gerendert hat, wird sie auf die CPU zurückgelesen und das Intensitätsmaximum bestimmt. Mit den Texturkoordinaten (u,v) des Maximums und dem Offset können nun auf dieselbe Weise die Pixelkoordinaten der gefundenen Korrespondenz-Linie zurückgerechnet werden. Es ist außerdem möglich, parallele Reduktion [Fer04] anzuwenden um das Texturmaximum auf der Grafikhardware zu bestimmen.

Neben der Länge der Korrespondenzlinien kann auch hier die Mehrdeutigkeit paralleler Bildlinien berücksichtigt werden. Ihre Resultate sind in der Ausgabetextur diagonal angeordnet, sodass dieses Qualitätskriterium ebenfalls schnell überprüft werden kann. Wird etwa in Abbildung 52 ein Maximum an Position (2,2) gefunden und es befinden sich weitere hohe Ergebnisse auf der Diagonalen von (0,0) bis (9,9), so weist dies auf die Existenz mindestens einer parallelen Bildlinie mit ähnlicher Länge hin, was zur Ablehnung der Korrespondenz führt.

7.2.5 Optical Flow Unterstützung

Starkes Wackeln der Kamera führt zu einem hohen Auftreten von Bewegungsunschärfe im Bild. Da der durch die Sample-Kanten abgedeckte Such-

raum begrenzt ist, kann die korrekte Korrespondenzfindung und in Folge das Tracking der Kamerapose verloren gehen, wenn die Bildlinien ruckartig zu weit bewegt werden. Um dies zu verhindern, können starke Bewegungen durch die Berechnung des Optischen Flusses (Optical Flow) im Bild erkannt werden [BB95]. Optical Flow beschreibt die Bewegung von Pixeln über die Bilder einer Sequenz (Abbildung 53). Die Bewegung der einzelnen Start- und Endpunkte jeder projizierten Modellkante zwischen dem letzten und dem aktuellen Kamerabild können unter der Nutzung einer Optical Flow Funktion, etwa aus der OpenCV-Bibliothek [BK08], berechnet werden. Die so korrigierten neuen Positionen der Start- und Endpunkte werden dann für die Erzeugung der Sample-Kanten verwendet. Die Korrespondenzsuche mit Optical Flow Unterstützung ermöglicht die Reduzierung der benötigten Sample-Kanten, da die Suche in einer kleineren Region stattfinden kann, was zu einer höheren Performanz führt. Eine andere Möglichkeit, die Bewegungsunschärfe auszugleichen, ist der Einsatz eines Inertialsensors [RD06] um schnelle Kamerabewegungen zu erkennen.

Abbildung 53: Optical Flow Berechnung für regelmäßige Punkte im Bild

Abbildung 54: Textszenen (Video und gerendertes Modell)

7.3 Ergebnisse

Der Ansatz zur Korrespondenzsuche wurde anhand einfacher und komplexer Objekte in Indoor- und Outdoor-Trackingszenen getestet (Abbildung 54). Dazu wurden Videosequenzen mit der Auflösung 640x480 unter verschiedenen Beleuchtungsbedingungen aufgenommen. Für die Anwendung des Canny-Filters (Standardeinstellungen threshold1 = 50, threshold2 = 200, aperture = 3) und der Funktion zur Berechnung des Optical Flow wurde die OpenCV-Bibliothek [BK08] verwendet.

Die initiale Kamerapose ist zu Beginn der Sequenz zumindest als grob bekannt vorausgesetzt oder muss durch manuelle Positionierung des Modells im Kamerabild gegeben werden. Die intrinsischen Kameraparameter der Videokamera werden durch Kalibrierung erlangt. Für die Sequenzen sind entsprechende Modelle vorhanden, von denen im Vorfeld Listen mit Kanten erstellt werden, zu denen Korrespondenzen gefunden werden sollen. Zur Berechnung der Kamerapose aus den Linienkorrespondenzen kommt ein nichtlinearer Levenberg-Marquardt Optimierer zum Einsatz (Kapitel 5).

Bei den Tests wurden die Korrespondenzsuche im Hough-Parameter-

raum und zwei weitere distanzbasierte Methoden verglichen. Im ersten Fall werden aus dem Canny-Bild mit der Hough-Transformation [DH72] Parameterbeschreibungen möglicher Geraden im Bild extrahiert, indem unter Anwendung der Geradengleichung $d = x * \cos(\alpha) + y * \sin(\alpha)$ nach Häufungen von kollinearen Punkten gesucht wird. Das Ergebnis ist eine Liste von Parameterpaaren aus Abstand zum Bildursprung und Winkel zwischen Geradennormale und y-Achse, welche die Geraden im Bild definieren. Diese werden dann direkt mit den Hough-Parametern, die aus der projizierten Modellkante berechnet werden können, verglichen. Die aus einem solchen Vorgehen gewonnenen Korrespondenzen erwiesen sich jedoch als nicht sehr stabil, da die Hough-Transformation in Abhängigkeit der gewählten Parameter für die Diskretisierung des dualen Hough-Raumes viele nahe und daher mehrdeutige Geraden erzeugt. Darüber hinaus beruhen die Geradenparameter nur auf einer Lageähnlichkeit, beschreiben aber nicht die genaue Position und Länge im Bild. Erfolgreiches Tracking ist kaum möglich.

Die Realisierung der distanzbasierten Korrespondenzsuche im Bildraum kann durch die Projektion von Start- und Endpunkt der Modellkante auf die per Hough-Transformation erkannten Geraden im Bild erfolgen. Die Modellkante wird dazu in das Kamerabild projiziert und mit einem Suchfenster um diese Kante eine Region definiert, in der die Hough-Transformation ausgeführt wird (Abbildung 55). Die Hough-Geraden werden ins Bild gezeichnet und die gemessene Distanz beider Punkte zur Bildgeraden gibt Auskunft über den Abstand und den Winkel zwischen den Korrespondenzen. Das Längenverhältnis wird dabei jedoch nicht berücksichtigt. Dieser Ansatz ist stabiler als der vorhergehende, dennoch besteht auch hier die Abhängigkeit von den gewählten Parametern. Das Tracking der Kamera-

pose ist möglich, doch mehrdeutige Korrespondenzlinien durch die von der Hough-Transformation verursachten Geradenbündel im Bild erzeugen teilweise störendes Rauschen der Pose. Bei starker Bewegung schlägt die Korrespondenzsuche komplett fehl.

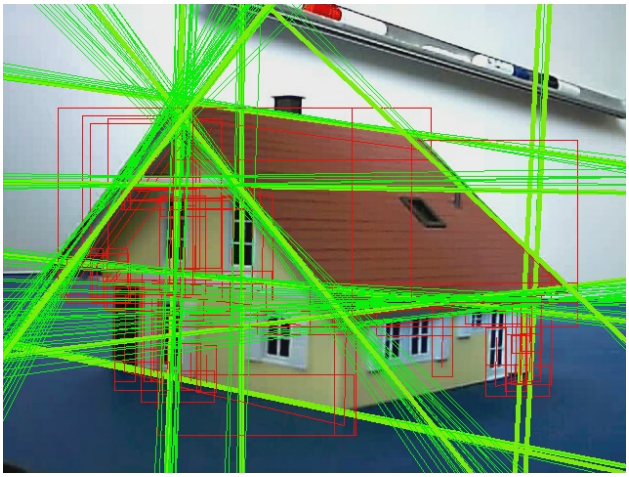


Abbildung 55: Beispiel des Hough-Matchings mit typischen Geradenbündeln.

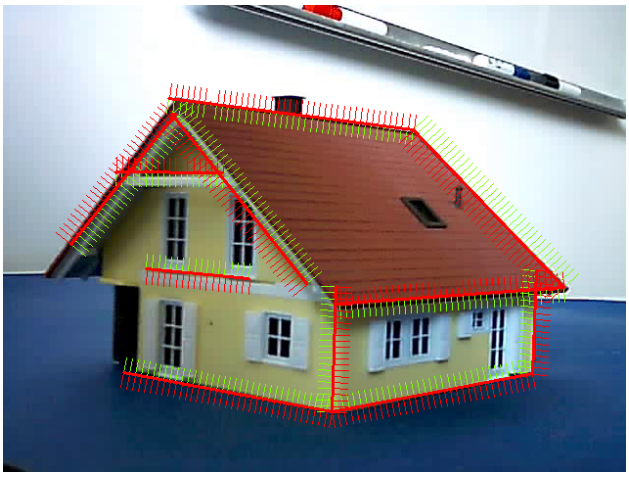


Abbildung 56: Beispiel des Matchings mit orthogonalen Suchlinien.

Eine weitere, häufig eingesetzte Methode ist die orthogonale Suche. Dazu werden Kontrollpunkte entlang der Modellkante definiert, durch welche orthogonale Suchlinien aufgespannt werden. Auf diesen Normalen wird

dann nach starken Gradienten im Bild gesucht (Abbildung 56). Zu jedem 3D Kontrollpunkt ergibt sich so eine Korrespondenz mit einem 2D Bildpunkt. Dieses Vorgehen führt zu einer stabileren Pose. Hierbei wird jedoch ebenfalls nicht die zu erwartende Länge der Linie im Bild berücksichtigt. Außerdem können durch das Sampling Linienteile verpasst werden, wenn die Bildlinie unterbrochen ist.

Generell arbeitet die distanzbasierte Korrespondenzsuche im Bildraum stabil, wenn die Kamerabewegung langsam und gleichmäßig ist, wie anhand der Indoor-Szene (Abbildung 54, oben) getestet werden konnte. Bei schnellen Bewegungen der Kamera (Abbildung 54, unten) tritt jedoch Bewegungsunschärfe auf, welche die Korrespondenzsuche scheitern lässt. Der Matching-Shader kann dabei ein Ergebnis mit geringem Fehler liefern, auch wenn die Videosequenz von einer ruckartig bewegten Handkamera stammt.

Die folgenden Abbildungen zeigen die Ergebnisse der Korrespondenzsuche mit den besprochenen Ansätzen auf der zweiten Testsequenz im Bild unmittelbar nach dem Auftreten starker Bewegungsunschärfe. Die Unschärfe erstreckt sich über 6 Bilder, wobei der Bildinhalt durch eine schnelle Kamerabewegung über 80 Pixel verschoben wird. Abbildung 57 verdeutlicht den Pixelversatz innerhalb der 6 Bilder. Zusätzlich ist das entsprechende Canny-Bild im Moment der stärksten Bewegungsunschärfe dargestellt. Die Bildlinien sind nur teilweise noch sichtbar, was das Korrespondenzergebnis negativ beeinflusst. Gemessen wird jeweils die erreichte Genauigkeit bei der Korrespondenzsuche durch den maximalen Pixelfehler zwischen Modellkante und Bildlinie. Dazu werden 3 Kanten definiert, zu denen im Bild die beste Korrespondenz gefunden werden soll.

Im Ansatz mit der Projektion von Start- und Endpunkt (Abbildung 58) wird die Modellkante stark von der parallelen Dachrinne neben der

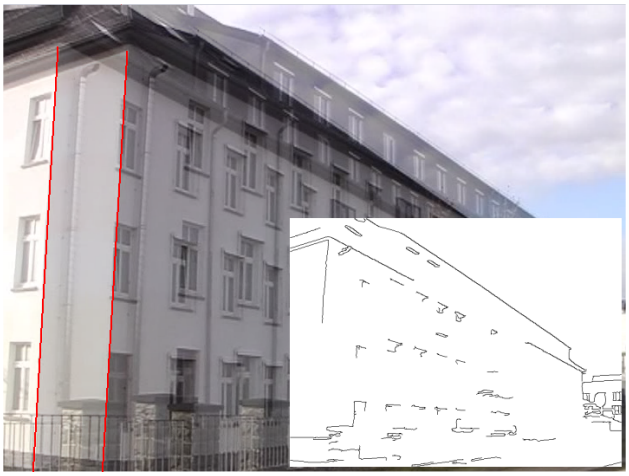


Abbildung 57: Pixelversatz mit starker Bewegungsunschärfe

Hausecke abgelenkt, da dies eine Mehrdeutigkeit darstellt, die nicht erkannt wird. Der maximale Fehler liegt bei 37 Pixeln. Die Korrespondenzsuche mit orthogonalen Suchlinien (Abbildung 59) erreicht eine höhere Genauigkeit von 24 Pixeln, wird jedoch noch von der Bewegungsunschärfe und der damit verbundenen Unterbrechung der Bildlinie gestört. Der Einsatz des Matching-Shaders (Abbildung 60) findet die Korrespondenz bis auf 3 Pixel genau.

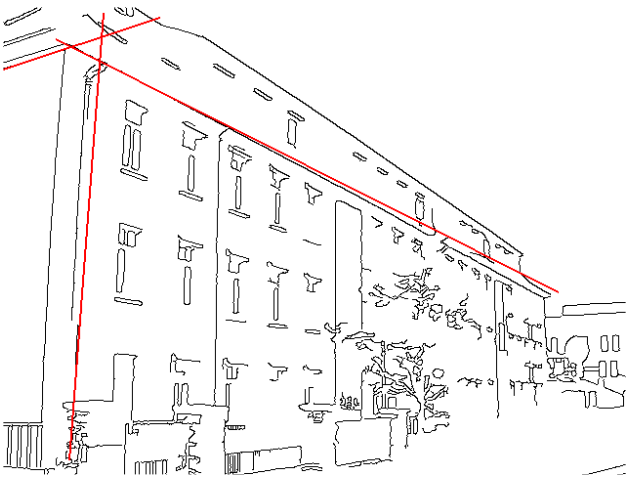


Abbildung 58: Ergebnis des Ansatzes mit Projektion von Start- und Endpunkt

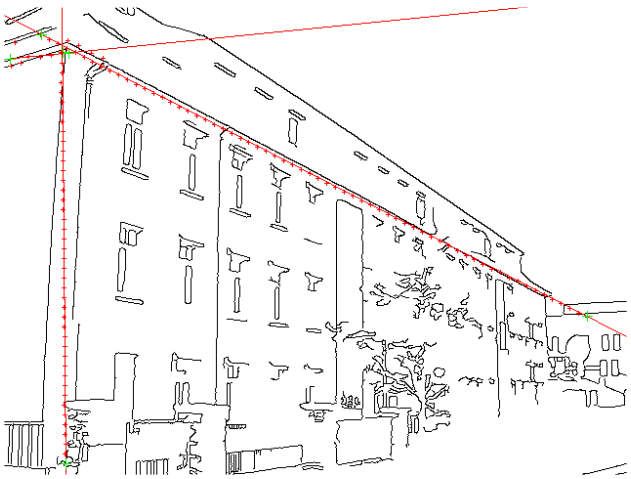


Abbildung 59: Ergebnis des Ansatzes mit orthogonalen Suchlinien

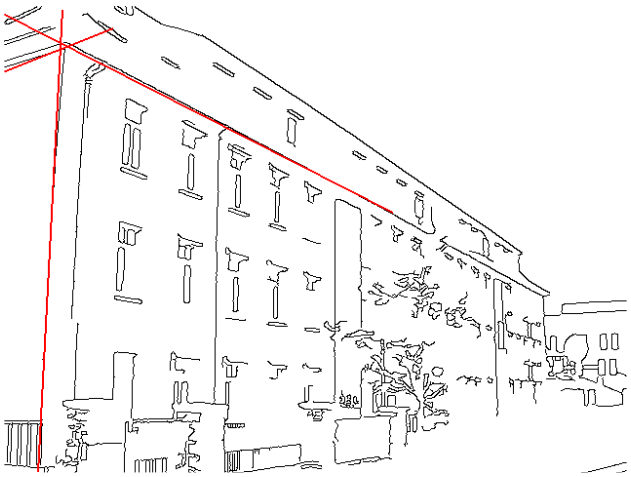


Abbildung 60: Ergebnis des Matching-Shaders nach der Bewegung

Abbildung 61 zeigt den Moment der stärksten Bewegungsunschärfe in der Outdoor-Sequenz zusammen mit der Überlagerung der Poseberechnung, die aus den im vorgestellten Ansatz erstellten Korrespondenzen resultiert. Für die Korrespondenzsuche mit Optical Flow Unterstützung erwies sich ein Sample-Fenster der Größe 4x4 als optimal. Ohne Berücksichtigung des Optical Flow wird der beste Kompromiss zwischen Rechenzeit und Korrespondenzqualität bei einem Sample-Fenster von 7x7 Pixeln erreicht. Der beste Schwellwert zur Akzeptanz von Korrespondenzen lag bei einer

Abbildung 61: Korrekte Poseberechnung bei starker Bewegungsunschärfe

Bildlinienlänge von 3/4 der Modellkante. Abbildungen 62 und 63 zeigen weitere Ergebnisse.

Abbildung 62: Weitere Ergebnisse der Poseberechnung, Indoor-Szene

Tabelle 4 listet die durchschnittliche Rechenzeit in Millisekunden für die einzelnen Komponenten der Korrespondenzsuche auf. Verwendet wurde ein Intel Core2Duo 3.2 GHz CPU und eine nVidia GeForce GTX 285 Grafikkarte. Der Canny-Filter und die Berechnung des Optical Flow werden jeweils einmal für jedes neue Kamerabild ausgeführt und sind unabhängig

Abbildung 63: Weitere Ergebnisse der Poseberechnung, Outdoor-Szene

Canny Filter	5 ms		
Motion Flow	4 ms		
Occlusion Query	97 ms (11)	54 ms (7)	7 ms (2)
Rückgabetextur	18 ms (11)	14 ms (7)	7 ms (2)

Tabelle 4: Rechenzeit

von der Anzahl der Kanten, zu denen eine Korrespondenz gesucht wird. Obwohl die Performanzeinbuße durch die Anwendung des Canny-Filters gering ausfällt, wäre es auch möglich, hier eine GPU Implementierung zu verwenden.

Des Weiteren werden die Rechenzeiten für den Occlusion Query Ansatz und die Erweiterung mit Rückgabetextur verglichen. Die Zahl in Klammern gibt die Anzahl der Kanten an, zu denen eine Korrespondenz gesucht wird. Offensichtlich ist die Verwendung von Occlusion Queries nur dann hinreichend schnell, wenn eine geringe Menge Kanten verarbeitet werden soll. Bei steigender Anzahl an Modellkanten ist der Texture Read-Back Ansatz deutlich schneller. Insgesamt ist eine Korrespondenzsuche in Echtzeit gewährleistet.

7.4 Fazit

Es wurde ein shaderbasierter Ansatz zum Erstellen von Korrespondenzen zwischen Modellkanten und Bildlinien in einem modellbasierten Trackingszenario vorgestellt. Ausgehend von der letzten berechneten Pose werden aus den gegebenen Modellkanten mehrere Sample-Kanten generiert und mit dem Matching-Shader gerendert. Der Shader zählt die Pixel der Bildlinien eines mit dem Canny-Algorithmus gefilterten Kamerabildes an der entsprechenden Position der projizierten Kante. Diejenige Bildlinie mit der höchsten Übereinstimmung der Kriterien Länge und Distanz wird als Korrespondenz akzeptiert.

Es wurden zwei Methoden zur Realisierung des Matching-Shaders beschrieben. Ein Shader, der nur die Kantenpixel zählt und Occlusion Queries zum Auslesen des Ergebnisses einsetzt, ist einfach zu implementieren. Die Bildrate reduziert sich bei der Generierung vieler Sample-Kanten jedoch deutlich, da jede Kante einen eigenen Renderdurchgang benötigt. Ein weiter entwickelter Ansatz nutzt Texturen um das Matchingergebnis schneller zurückzulesen. Der gesamte Prozess der Generierung von Sample-Kanten wird in den Shader transferiert, sodass pro Modellkante nur ein Renderaufruf erfolgt, statt wie bisher bei jeder einzelnen Sample-Kante.

Obwohl in dieser Arbeit nur gerade Kanten aus den Gebäudemodellen verwendet werden, ist das Konzept des Ansatzes nicht auf diesen Typ Merkmal beschränkt. Der Matching-Shader kann für das Suchen nach Korrespondenzen anderer renderbarer Strukturen wie Kreise, gekrümmte Kurven oder NURBS erweitert werden. Dazu muss lediglich der Algorithmus zur Erzeugung der Samples an die jeweilige Struktur angepasst werden. Eine weitere Verbesserung wäre die Integration der Canny-Berechnung in den Shader. Eine zusätzliche nutzbare Information wäre die Richtung des Gra-

dienten der Bildmerkmale. Da die Orientierung der Modellkante bekannt ist, könnten alle Pixel, die nicht dieselbe Kantenrichtung aufweisen, vom Beitrag zum Ergebnis ausgeschlossen werden.

8 Fazit und Ausblick

In der vorliegenden Arbeit wurden die Einsatzmöglichkeiten von *Analyse-durch-Synthese* Techniken im Rahmen des Kamerapose-Trackings untersucht. Es wurden entsprechende Methoden entwickelt und festgestellt, ob diese geeignet sind, anhand von Informationen aus Modell, Renderingprozess und Umgebung die einzelnen Komponenten eines Trackingsystems zu verbessern. Die untersuchten Schwerpunkte liegen auf der Synthese von besonders geeigneten Merkmalen durch einen Managementprozess, sowie auf der Korrespondenzfindung im Kamerabild. Des Weiteren wurde eine Analyse der optimalen Parameter für eine nichtlineare Poseberechnung mit Punkten und Kanten durchgeführt und eine GPS-gestützte Initialisierung des Trackingsystems in einem Outdoor-Szenario vorgestellt.

Zunächst wurden **Voruntersuchungen** zur anzuwendenden Optimierungsstrategie für die Berechnung der Pose, der eingesetzten Vergleichsmaße zwischen Bildmerkmalen und zwischen Bildern, sowie den Anforderungen an Modell und Rendering durchgeführt. Diese zeigten, dass der Vergleich eines synthetischen Bildes mit dem Kamerabild bei Verwendung gängiger Merkmalsbeschreibungen zum Zeitpunkt der Untersuchung auch unter korrekt simulierter Beleuchtung noch nicht hinreichend genau war, um durchgängig stabiles und präzises Tracking zu ermöglichen. Zwischen einem gerenderten Bild und dem realen Kamerabild konnten nur unzureichende Korrespondenzen für eine Fehlerminimierung erstellt werden, da die Deskriptoren der Merkmale auf beiden Bildern nur in geringer Zahl übereinstimmende Ergebnisse lieferten. Folglich wurde vorgeschlagen, die Merkmale direkt aus der Geometrie des Modells zu gewinnen.

Gleichzeitig zeigt jedoch die Entwicklung höherer markanter Merkmale [PSH09] einen vielversprechenden Ansatz, SIFT-Merkmale in einem Ka-

merabild denen aus einem gerenderten Bild zuzuordnen. Dazu werden elementare Merkmale zu semantischen Gruppen zusammengefasst, die höhere Strukturen im Bild repräsentieren. Da in gängigen Verfahren nur die lokal betrachteten Punktmerkmalsdeskriptoren anhand ihres euklidischen Abstandes verglichen werden, ist abzusehen, dass das Matching deutlich verbessert werden kann, wenn stattdessen auch auf semantische Ähnlichkeit getestet wird. Dies gilt insbesondere für sich wiederholende Strukturen, wie sie im hier betrachteten Szenario mit Gebäudemodellen auftreten. Auf neuere Entwicklungen im Bereich der Merkmalsdetektion und Korrespondenzsuche wird im Abschnitt **Forschungsschwerpunkte** noch näher eingegangen.

Ausgehend von den durchgeführten Voruntersuchungen wurde weiterhin eine **Initialisierung** des Posetrackings mit einem ähnlichkeitsbasierten Bildvergleich auf Basis von Kantenbildern realisiert. In einem Outdoor-Szenario mit Gebäudemodellen wurde mit Hilfe von GPS und Kompassdaten eines handelsüblichen Smartphones automatisch das für den Benutzer sichtbare Modell bestimmt und so ausgerichtet, dass ein kontinuierliches Tracking gestartet werden kann. Da die bereits durch GPS und Kompass erlangte Pose meist noch zu grob ist, wird zusätzlich ein Verfeinerungsschritt vorgenommen. Dazu werden synthetische Referenzbilder erstellt, indem um die grobe Pose neue virtuelle Posen gestreut werden. Die so gerenderten Bilder werden anhand von Stärke und Richtung der in ihnen gefundenen Kanten auf Ähnlichkeit mit dem Kamerabild verglichen. Die mit dieser Methode vorgeschlagene Pose konnte in den meisten Fällen ohne weitere Korrektur zum Start des Trackings verwendet werden.

In der Arbeit von [Noh12] wurde dieser Ansatz bereits unter Anwendung eines Partikelfilters erweitert und der Vergleich der Kantenbilder wie

vorgeschlagen auf die GPU verlagert. Somit konnte der bisher für den Initialisierungsschritt vorgenommene Bildvergleich, basierend auf Kantenstärke und Kantenrichtung, beschleunigt und für ein kontinuierliches Tracking mit Reinitialisierung im Falle eines Verlustes der Pose eingesetzt werden. Die Arbeit von [CC13] zeigt eine weitere Geschwindigkeitsoptimierung auf, indem das Streuen der Posen durch den Partikelfilter ebenfalls die Parallelisierung der GPU ausnutzt. Da das vorgestellte System eine RGB-D Kamera einsetzt, können hier für die Ähnlichkeitsbestimmung zwischen dem gerenderten 3D Modell und der aufgenommenen Szene neben Modellpunkten auch Flächennormalen und die Farbe herangezogen werden.

Neben der Verwendung von GPS und Kompassdaten des Smartphones bietet sich auch der Einsatz der Inertialsensorik an, um bei ruckartigen Kamerabewegungen oder Verdeckungen, die einen Poseverlust verursachen können, eine schnelle Reinitialisierung zu ermöglichen. Visuelles Kameraposetracking wurde bereits auf mobilen Endgeräten unter Fusion mit inertialen Sensordaten umgesetzt [TSR15][WIS+18]. Es wurde anhand eines SLAM-Systems gezeigt, dass die Genauigkeit des Trackingergebnis durch stützende Daten verbessert werden kann. Ebenso wurde durch alternierende Anwendung von visuellem und inertialem Tracking eine kontinuierliche Poseschätzung bei Verlust der visuellen Pose ermöglicht. Dabei ergab sich, dass die Daten eines Beschleunigungssensors aufgrund der Rauschanfälligkeit nur für kurze Überbrückungen des Poseverlustes geeignet sind und die Verwendung von Gyroskopdaten überlegen ist.

Zur Berechnung der **Pose** sollte aufgrund der Robustheit gegenüber fehlerhaften Korrespondenzen eine nichtlineare Optimierungsstrategie zum Einsatz kommen. Dazu wurde eine Analyse über mögliche Optimierungsverfahren, Korrespondenz-Abstandsmaße im Bild- und Objektraum, sowie

Parametrisierungen der Rotationsparameter durchgeführt. In Tests wurde der Einfluss des gewählten Merkmalstyps von Punkt- und Kantenkorrespondenzen, der Merkmalsanzahl, sowie der fehlerhafter Korrespondenzen auf die Robustheit der Pose bestimmt. Das Ergebnis ist ein Poseschätzer, welcher in Kombination den Levenberg-Marquardt Algorithmus, eine Rotationsparametrisierung nach Rodrigues und ein pixelbasiertes Abstandsmaß sowohl für Punktmerkmale als auch für Geradenmerkmale einsetzt und als Eingabemenge eine minimale, aber beliebig kombinierte Anzahl aus Punkt- und Kantenmerkmalen akzeptiert. Zusätzlich können die Eingabekorrespondenzen anhand des Modellwissens auf kritische Konfigurationen hin analysiert werden.

Da es während des Trackingvorgangs zu Problemen bei der Korrespondenzsuche kommen kann, wenn die Merkmale im nächsten Bild nicht eindeutig wiederzuerkennen sind, kann das Wissen über Modell und Rendering auf die **Bewertung** der Merkmale und die Korrespondenzsuche angewandt werden. Aus dem gegebenen Modell werden Informationen über die Beschaffenheit und topologische Anordnung gesammelt, sowie beeinflussende Renderingparameter wie Beleuchtung und Perspektive oder die Verdeckungswahrscheinlichkeit berücksichtigt, um ein Qualitätsmaß zu definieren. Die betrachteten Kriterien im Deskriptor sind Länge, Distanz, Silhouette, Richtung, Beleuchtung und Korrespondenzqualität, sowie der zeitliche Verlauf der Merkmalsqualität. Somit soll eine möglichst minimale, jedoch qualitative Menge an Merkmalen ausgewählt werden, welche für die jeweilige Situation am besten geeignet erscheint. In Tests wurde der Einfluss der Kriterien im Merkmalsmanagement getestet, wobei die Präzision der berechneten Pose verbessert werden konnte. Zwar zeigen die Ergebnisse, dass die Einbeziehung von Wissen über Modell, Rendering

und Umgebung in die einzelnen Komponenten des Trackingprozesses die Verwendung von beliebig komplexen Modellen ermöglicht, welche nicht speziell für den Einsatz in einem Trackingszenario erstellt wurden. Doch ist noch eine starke Abhängigkeit von der Korrektheit des Modells gegeben und die Evaluierung der Merkmale anhand der vorgeschlagenen Qualitätskriterien szenenabhängig. Der Fokus zukünftiger Forschung sollte daher in der Szenenanalyse liegen, insbesondere aber auf der Suche nach eindeutigen und robusten Korrespondenzen.

Zur Bewertung der Korrespondenzqualität muss die **Korrespondenzsuche** für jedes Modellmerkmal die Anzahl der im Bild gefundenen Pixel zurückgeben können. Unter Einbeziehung des Modellwissens ist die zu erwartende Länge eines Modellmerkmals nach der Projektion bekannt und das Verhältnis zur tatsächlich gefundenen Länge im Bild kann als Kriterium angegeben werden. Um dies zu realisieren, wurde ein shaderbasierter Kantenmatcher entwickelt. Auf einem Kantenbild der Kamera werden für jedes Modellmerkmal variierte Kantenmerkmale im Bild simuliert und überprüft, ob ihnen entsprechende Kantenpixel auf dem Eingabebild zu finden sind. Die Summe der gezählten Pixel wird über eine Rückgabetextur effizient ausgelesen, sodass über das Texturmaximum auf die Kantenkorrespondenz im Bild geschlossen werden kann.

Zur Verbesserung der Robustheit beim Finden von korrekten 2D-3D Linienkorrespondenzen könnte eine spezielle Beschreibung der Kanten bezüglich ihrer Umgebung beitragen. Ein solcher Deskriptor für Linien findet sich bei [HS12]. Der dort vorgestellte LEHF-Deskriptor (Line-based Eight-directional Histogram Feature) wird in einem auf Liniensegmenten arbeitenden SLAM-System eingesetzt. Die Beschreibung basiert auf einer Berechnung der Gradienten in acht Richtungen, ähnlich dem SIFT-Deskriptor.

Mehrere Richtungshistogramme werden entlang von Linien orthogonal zum Kantenmerkmal erstellt und kombiniert. Der Deskriptor ist daher echtzeitfähig und rotationsinvariant.

Eine weitere Verbesserung des Kantenmatchings ist unter Einsatz einer RGB-D Kamera zu erwarten. Da die Tiefenwerte pro Pixel aus dem Rendering des Modells bekannt sind, können diese mit dem Tiefenbild der realen Szene abgeglichen und als weiteres Kriterium für die Bewertung der Korrespondenzqualität herangezogen werden. Der vorgestellte Matchingshader kann auf einfache Weise parallel auf Kantenbild und Tiefenbild zugreifen und folglich bei zu großer Abweichung im Tiefenvergleich falsch zugeordnete Kantenpixel verwerfen. So wird die Wahrscheinlichkeit von fehlerhaften Korrespondenzen durch nahe beieinander liegende Kanten weiter verringert. Zusätzlich kann die Tiefeninformation bei der Suche nach Kanten genutzt werden, die durch Tiefenänderungen entstehen. Die intensitätsbasierte Kantensuche auf dem Farbraum wird um die Tiefenkanten ergänzt oder kann zur Verifikation echter Objektkanten gegenüber etwa Schattenkanten genutzt werden.

Da die erfolgversprechendsten zukünftigen **Forschungsschwerpunkte** neben der modellbasierten *Analyse-durch-Synthese* auf SLAM-Verfahren liegen, untersucht die Arbeit von [Gem12] die Kombinationsmöglichkeiten beider Techniken. Vor- und Nachteile der Verfahren werden gegenübergestellt und denkbare Ansatzpunkte zum Ausgleich skizziert. Tracking mit einem gegebenen Modell ist, wie bereits bemerkt, immer abhängig von der Genauigkeit der Modellierung. Die SLAM-Verfahren hingegen leiden unter Ungenauigkeiten in der Tiefenschätzung und der Aufnahme fehlerhafter Korrespondenzen bei dynamischen Umgebungen, sowie an Problemen mit begrenzter Größe und Skalierung der Karte. Die Arbeit legt die Grundlagen

für die technische Realisierung eines kombinierten Trackings mit direktem Vergleich der Qualität der berechneten Posen aus beiden Verfahren, wobei alternierend die Pose mit dem jeweils geringeren Rückprojektionsfehler genutzt wird. Dabei zeigte sich, dass solch eine parallele Berechnung der Pose nicht sinnvoll ist, da die Angleichung und Skalierung der Koordinatensysteme beider Verfahren zu ungenau für eine Bewertung ist. Stattdessen wird vorgeschlagen, die eingesetzten Techniken in einem Verfahren zu kombinieren, indem SLAM um das Modellwissen ergänzt wird. Denkbar ist die Korrektur der Punktewolke durch die Tiefeninformationen des Modells, da die rekonstruierten Punkte über die Ebenen des Modells auf Plausibilität geprüft und entsprechend gruppiert werden können. Gleichzeitig ist es möglich, das Modell um eindeutig wiederzuerkennende Texturpatches aus der SLAM-Karte zu erweitern. Eine Umsetzung der Verbindung von rekonstruierter Punktewolke und gegebenem 3D Modell steht noch aus.

Einen umfassenden Überblick zu Entwicklungen im Bereich der SLAM-Verfahren geben [CCC+16] und [ATea20]. Demnach sind diese Trackingverfahren nach über 30 Jahren noch immer das vorherrschende Forschungsthema, insbesondere im Bereich der Robotik, auf dessen Gebiet die größten Fortschritte im Hinblick auf Robustheit und Zuverlässigkeit zu erwarten sind. Die Breite der bereits veröffentlichten Methoden ist sehr groß, da diese häufig auf spezifische Anwendungen und für den Einsatz in definierten Umgebungen hin entwickelt wurden. Im Allgemeinen lassen sie sich nur schwer generalisieren und auf neue Problemstellungen anpassen, da oft bestimmte Voraussetzungen erfüllt sein müssen, etwa Merkmalsart und -häufigkeit oder eine statische Szene.

Die gängigste Klasse der SLAM-Verfahren bilden die *merkmalsbasierten* oder *indirekten* Ansätze bei denen Merkmale aus den Kamerabildern

extrahiert, anhand ihrer Nachbarschaft möglichst eindeutig durch einen Deskriptor beschrieben und per Korrespondenzsuche mit den Merkmalen folgender Bilder abgeglichen werden. Dieses Vorgehen setzt jedoch das Vorhandensein einer ausreichenden Menge von erkennbaren Merkmalen in der Umgebung voraus. Problematisch ist zusätzlich, dass die Zuverlässigkeit der Algorithmen zur Erkennung und Beschreibung von Merkmalen von Schwellwerten abhängt, die oft nicht allgemeingültig bestimmt werden können. Des Weiteren sind viele Merkmalsdetektoren geschwindigkeitsoptimiert, was eher zu Lasten der Präzision geht [CCC+16].

In [ATea20] werden die merkmalsbasierten Verfahren nach der Art der eingesetzten Merkmale unterschieden: *Low-level* Merkmale, wie Punkte, Ecken und Kanten, die nur wenig Bildinformation nutzen und keine semantische Ordnung besitzen. *Medium-level* Merkmale, die flächige Elemente wie Ebenen beschreiben und in Umgebungen von Vorteil sind, die wenig Textur enthalten, etwa in Innenräumen. *High-level* Merkmale, die durch semantische Struktur Objekte beschreiben können und sich so gleichermaßen für Anwendungen im Innen- und Außenbereich eignen. Darüber hinaus können diese Merkmale auch kombiniert in *hybriden* Verfahren eingesetzt werden.

Erwähnt sei an dieser Stelle *PL-SLAM*, die Verbindung von Punkt- und Kantenmerkmalen. Dieser Ansatz soll in allen Arten von Umgebungen einsetzbar sein, gerade auch in gering texturierten Szenen und bei Auftreten von Bewegungsunschärfe, wenn Punktmerkmale wenig zuverlässig sind, sei es durch eine zu geringe Merkmalsdichte oder schlechte Verteilung. Die Hinzunahme von Kanten verbessert hier die Genauigkeit und Robustheit des SLAM-Systems deutlich, ohne dabei die Effizienz zu beeinträchtigen [PVA+17][GOMZN+19].

Die neuesten Entwicklungen im Bereich SLAM verzichten bereits ganz auf vorherige Merkmalsdetektion und Korrespondenzsuche in den Kamerabildern. Diese *direkten* Verfahren arbeiten auf den reinen Pixelinformationen. Betrachten sie das ganze Bild, wird ihre Methode als *dense* bezeichnet. Gegenüber den merkmalsbasierten Verfahren sind sie robuster bei geringer Texturierung oder Unschärfe, sind jedoch auch rechenintensiver. Einen Geschwindigkeitsvorteil bieten Methoden, die als *sparse* oder *semi-dense* bezeichnet werden und nur Bildbereiche mit ausreichend starkem Gradienten berücksichtigen. Auch die Kombination aus merkmalsbasierter und direkter Methode hat sich als besonders effizient erwiesen [CCC+16].

In den Arbeiten von [ESC13] und [ESC14] werden Tiefenkarten der Umgebung aus direkten Stereo-Pixelvergleichen rekonstruiert (Large-Scale Direct Monocular SLAM). Im Sinne der Komplexitätsreduktion werden nur jene Bereiche im Bild betrachtet, die einen ausreichenden Intensitätsgradienten aufweisen. Dieser Ansatz löst die größten Nachteile bisheriger SLAM-Verfahren, da er das Mapping und Tracking von Szenen mit großen Änderungen in Skalierung und Rotation bei gleichzeitiger Echtzeitfähigkeit auf der CPU beherrscht. Die Umsetzung solcher direkt-visuellen Verfahren unter Verwendung von reduzierten Tiefenkarten wurde ebenfalls bereits für den mobilen Einsatz demonstriert [SEC14].

Um Effizienz und Genauigkeit weiter zu steigern, wenden [CH19] die direkte Methode nur auf Kantenzüge im Bild an und vermeiden so die Verarbeitung redundanter Information im Gegensatz zur Betrachtung des gesamten Bildinhalts. Mit dem Canny Kantendetektor werden auf jedem neuen Kamerabild Kanten extrahiert und als Maske für die Gradientensuche im Referenzbild herangezogen. Der Fehler zwischen allen Kantenpixeln beider Bilder wird minimiert, um die Bilder per rigider Transformation in

Deckung zu bringen. Eine Minimierung ausschließlich über die Bildkanten kann laut der Autoren als ausreichend angesehen werden, um den Fehler über das gesamte Bild zu minimieren und die Pose mit hoher Exaktheit zu bestimmen.

In den letzten Jahren hat im Bereich der maschinellen Verarbeitung von Bilddaten vermehrt das Konzept der *Convolutional Neural Networks* (CNNs) Verwendung gefunden. Es handelt sich dabei um einen *Deep Learning* Ansatz, der speziell für die Verabeitung von mehrdimensionalen Eingabedaten entwickelt wurde und in der lernenden Bildverarbeitung zurzeit am erfolgreichsten eingesetzt wird.

Ein CNN ist ein künstliches neuronales Netz, das die Arbeitsweise von Neuronen im visuellen System der menschlichen Wahrnehmung imitiert. Die innerhalb des Netzwerks miteinander verbundenen Neuroneneinheiten eines CNNs sind hierarchisch in Ebenen organisiert und ebenenübergreifend miteinander verbunden. Die Ebenen können abwechselnd aus Filterkernen bestehen, welche parallele Extraktion von Strukturen aus den Eingabedaten ermöglichen (*Convolutional Layer*), und aus Samplingschichten, welche die Daten durch Aggregation der wichtigen Informationen in ihrer Auflösung reduzieren und so in eine abstraktere Repräsentation überführen (*Pooling Layer*). Beginnend mit der Pixelebene werden so zunächst einfache Bildelemente wie Kanten oder Ecken erlernt, welche sich dann in Kombination zu komplexeren Strukturen wie etwa Konturen zusammensetzen und schließlich in ortsunabhängig klassifizierbare Objekte münden.

Die größten Fortschritte hin zur allgemeinen Einsetzbarkeit von CNNs wurden in den letzten Jahren bei der Beschleunigung der Rechenzeit, der Reduzierung des Speicherverbrauchs, sowie der erforderlichen Trainingsmenge gemacht [CCC+16][GBC16].

In der Arbeit von [KBM+15] wird solch ein CNN mit dem Trackingansatz der *Analyse-durch-Synthese* kombiniert. Als Eingabesensor dient eine RGB-D Kamera, deren Bilder mit einer Reihe gerenderter Bilder des Modells der Szene verglichen werden, um über die höchste Übereinstimmung die Pose zu bestimmen. Anhand eines Wahrscheinlichkeitsmodells lernt das CNN für gerenderte und aufgenommene Bilder die Berechnung einer zu minimierenden Energiefunktion. Herangezogen werden dazu pixelweise Vergleiche zwischen den gerenderten Vorhersagen des Modells mit dem jeweiligen Tiefenbild des Objekts, dem Bild der Objektkoordinaten und der Segmentierungsmaske des Objekts. Das CNN auf einem einzelnen Objekt zu trainieren ist ausreichend, um den Einsatz auf verschiedenen Objektgeometrien und -darstellungen zu ermöglichen. Dabei ist das Verfahren sehr stabil gegenüber starker Okklusion der Szene und verrauschten Hintergründen.

Eine interessante Weiterentwicklung sind lernende Merkmalsdetektoren, die zunächst mit einfachen synthetischen Geometrien trainiert werden. Hier zeigt sich die Möglichkeit, aus den in der *Analyse-durch-Synthese* gerenderten Bildern entsprechende Trainingsdaten zu gewinnen. In der Arbeit von [DMR18] wird ein selbstlernender Ansatz vorgestellt, bei dem domainübergreifend anhand von synthetischen Daten ein CNN trainiert wird, um das Erkennen und Beschreiben von Merkmalen auf realen Kamerabildern zu verbessern. Bei diesen Daten kann es sich um Strukturen wie Dreiecke, Kanten, Würfel, Schachbrettmuster und Sterne handeln, für deren Punkte eindeutige Lokalisierungen vorliegen. Das Training erfolgt über die Erzeugung von Bildpaaren, die einer bekannten homographischen Transformation unterzogen werden. Im Ergebnis werden klassische Merkmalsdetektoren bezüglich der Wiederholbarkeit besonders unter großen Änderungen der Perspektive und der Beleuchtung übertroffen, bei gleichzeitiger Erhöhung von Anzahl

und Dichte der korrekt zugeordneten Merkmalskorrespondenzen.

In den Voruntersuchungen der vorliegenden Arbeit wurden Versuche unternommen, mit Standardmethoden Merkmalsdetektion und Korrespondenzsuche auf realen und gerenderten Bildern durchzuführen, indem durch photorealistische Bilderzeugung eine möglichst hohe Übereinstimmung in den Bildern erreicht wird. Die Vermutung, dass dieses Vorgehen zu einer ausreichenden Zahl guter Korrespondenzen führt, hat sich nicht bestätigt. Stattdessen schlagen [DRP+19] einen CNN-basierten Ansatz vor, der das Finden von zuverlässigen Korrespondenzen auch unter großen Abweichungen in den Bilden ermöglicht. Dazu zählen Änderungen der Perspektive, schwierige Beleuchtungsbedingungen, wie Tag- und Nachtaufnahmen, Störung durch Bewegungsunschärfe, schwach texturierte Szenen und sogar unterschiedliche Stilisierung der Bilder.

Standard-Merkmalsdetektoren berücksichtigen nur lokale Bildbereiche und einfache Bildstrukturen, weshalb sie instabile Ergebnisse bei starken Bildänderungen liefern. Der vorgeschlagene Ansatz hingegen kombiniert Merkmalsbeschreibung und -lokalisierung. Dazu wird unter Einsatz eines CNN eine der SIFT-Methode ähnliche Reihe von Merkmalskarten erstellt. Auf diesen werden globale Deskriptoren über alle Ebenen der Auflösungspyramide berechnet und gleichzeitig die Merkmalspunkte als lokale Maxima in den Karten erkannt. So soll eine optimale Wiederholbarkeit bei gleichzeitiger maximaler Unterscheidbarkeit erlangt werden. Dieser Ansatz des simultanen Beschreibens-und-Erkennens von Merkmalen zeigt sich den bisherigen zweistufigen Methoden, welche Merkmale unabhängig voneinander zunächst erkennen und dann beschreiben, klar überlegen.

Aufgegriffen wird der Ansatz von [ZSS20], die ihn auf den Vergleich von gerenderten Bildern eines 3D Modells mit realen Kamerabildern anwenden,

um mit erlernten Merkmalen robuste Korrespondenzen zu erzeugen. Ihre Arbeit soll halbautomatisches Generieren von Referenz-Kameraposen für Benchmarks ermöglichen. Die Autoren stellen fest, dass klassische Merkmale wie ORB, SIFT und SURF Probleme in der Zuordnung zwischen Bildern haben, welche starken Änderungen der Aufnahmebedingungen unterliegen. Die von [DRP+19] vorgestellte Methode sei hingegen geeignet, zuverlässige Korrespondenzen auf gerenderten und realen Bildern zu ermöglichen, da ihr Ansatz keinen hohen Grad an Realismus im Rendering notwendig macht.

Abschließend ergeben sich aus dem aktuellen Stand der Forschung zwei mögliche zukünftige Anknüpfungspunkte für die vorliegende Arbeit. Während diese einen starken Fokus auf Kantenmerkmale legt, welche sich direkt aus dem Modell gewinnen lassen, wurden insbesondere im Bereich der Korrespondenzfindung zwischen stark disparaten Bildern große Fortschritte erzielt. Daher erscheint es gerade auch im Hinblick auf die in Kapitel **Pose** vorgestellte Berechnung vielversprechend, den gewählten Ansatz um eine Kombination mehrerer Merkmalstypen zu erweitern und entsprechend neu zu bewerten. Des Weiteren bietet sich durch die Entwicklung lernender CNNs die Möglichkeit, das **Merkmalsmanagement** für Kanten zu erweitern. Neben den bisher aus dem Modell herangezogenen Kriterien zur Qualitätsbeurteilung der Merkmale können zusätzliche Informationen aus dem Bild selbst gewonnen werden, welche Aussagen über die Struktur der Kantenzüge zulassen. Während das eingesetzte Kantenmatching nur einzelne Bildkanten berücksichtigt und deren Länge bewertet, wäre so die Erkennung von höheren Objektbestandteilen aus zusammenhängenden Kanten und eine Zuordnung zu Objektklassen möglich.

9 Anhang

9.1 Lineare Berechnung der Kamerapose

Lineare Verfahren lösen lineare Gleichungssysteme, bei denen jede Gleichung eine Bedingung aufstellt, welche für die Gesamtlösung erfüllt sein muss. Diese Verfahren sind echtzeitfähig und benötigen nur kurze Rechenzeiten, da sie die Berechnungen in geschlossener Form direkt durchführen. Es sind neben den Eingabekorrespondenzen keine initialen Bedingungen zu erfüllen. Die Genauigkeit der linearen Berechnungen ist jedoch begrenzt, da sie empfindlich gegenüber Ausreißern in den Messdaten sind und sich fehlerhafte Korrespondenzen negativ auf die Robustheit der Ergebnisse auswirken. Wenn die Berechnung der Pose linear durchgeführt wird, ist es nicht möglich, hochgradig nichtlineare Parameter zu berücksichtigen, wie etwa die durch optische Linsen verursachte Verzerrung. Ist die Anzahl der Korrespondenzen größer als 6, sind geschlossene Lösungen in der Regel nicht mehr effizient. Eine Übersicht der linearen Verfahren zeigt Tabelle 5.

Verfahren	Minimale Korrespondenzen	Parameter
Homographie	4 Punkte	P
DLT	6 Punkte	P
Tsai-Lenz	5 Punkte (koplanar) 7 Punkte (nicht koplanar)	K R T
POSIT (Punkte)	4 Punkte (nicht koplanar)	R T
POSIT (Linien)	3 Linien (koplanar, kein gemeinsamer Schnittpunkt, nicht parallel) 4 Linien (nicht koplanar, kein gemeinsamer Schnittpunkt oder Parallelität bei 3 Linien)	R T
Fluchtpunkte	3 Fluchtpunkte	R
Pose aus Marker	4 Linien und 4 Schnittpunkte	R T

Tabelle 5: Übersicht der linearen Verfahren. Minimale Voraussetzungen und Ergebnisparameter (P Projektionsmatrix, K Kameramatrix, R Rotationsmatrix, T Translationsvektor)

9.1.1 Homographie

Die einfachste Form der linearen Beschreibung einer Blickpunktänderung O nach O' ist die planare Homographie. Es handelt sich dabei um eine projektive Transformation $p'_i = Hp_i$ in homogenen Koordinaten, die Bildpunkte p_i einer Ebene mit einer entsprechenden 3x3 Matrix H auf Punkte p'_i einer anderen Bildebene abbildet, wobei p_i und p'_i aus der perspektivischen Projektion desselben Weltpunktes p_w erzeugt werden (Abbildung 64). Da H Rang 8 hat, lassen sich mit $n \geq 4$ Punktkorrespondenzen 8 lineare Gleichungen aufstellen, die das Problem lösen. Voraussetzung ist, dass von den 4 koplanaren Punkten keine 3 Punkte kollinear sind.

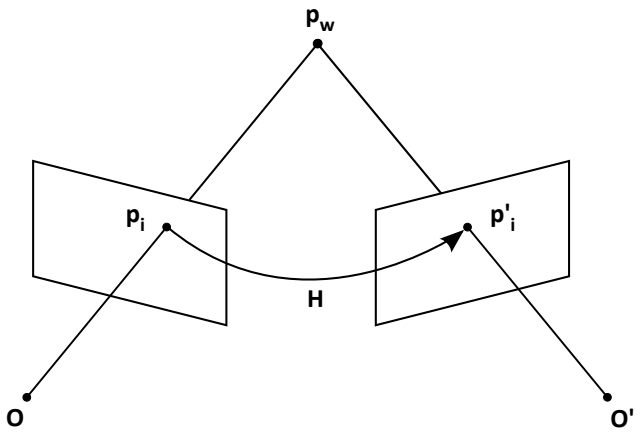


Abbildung 64: Planare Homographie

9.1.2 Direkte Lineare Transformation

Die Erweiterung der Homographie auf den 2D-3D Fall ist die Direkte Lineare Transformation (DLT) [HZ10]. Sie dient der Kamerakalibrierung und berechnet implizit 11 Kameraparameter, darunter 6 externe für die Freiheitsgrade der Rotation R und der Translation t, sowie 5 interne mit dem Bildhauptpunkt H_x, H_y, der Brennweite f, und der Pixelgröße d_x, d_y. Das

Ergebnis ist eine vollständige 3x4 Projektionsmatrix M mit den Elementen m_{ij}, die alle n Weltpunkte p_w in Bildpunkte p_i abbildet:

$$p_i = Mp_w$$

$$\begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix},$$

woraus sich ein homogenes lineares Gleichungssystem A aufstellen lässt, das $2n$ Gleichungen besitzt. Für jede Punktkorrespondenz gilt:

$$u_n = \frac{m_{11}x_n + m_{12}y_n + m_{13}z_n + m_{14}}{m_{31}x_n + m_{32}y_n + m_{33}z_n + m_{34}}$$

$$v_n = \frac{m_{21}x_n + m_{22}y_n + m_{23}z_n + m_{24}}{m_{31}x_n + m_{32}y_n + m_{33}z_n + m_{34}}$$

$$A = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & -u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & -v_1 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -u_nx_n & -u_ny_n & -u_nz_n & -u_n \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_nx_n & -v_ny_n & -v_nz_n & -v_n \end{bmatrix}.$$

Es lässt sich das Nullraumproblem

$$A\mathbf{m} = 0$$

mit

$$\mathbf{m} = \begin{pmatrix} m_{11} & m_{12} & ... & m_{34} \end{pmatrix}^T$$

definieren. Da A Rang 11 hat, kann das Gleichungssystem mit $n \geq 6$ Korrespondenzen unter Einsatz der Singulärwertzerlegung $A = UDV^T$ gelöst werden. Der Spaltenvektor von V, der dem kleinsten Singulärwert entspricht, beschreibt den Nullraum und ist damit der gesuchte Lösungsvektor. Es wird jedoch immer die gesamte Projektionsmatrix berechnet, sodass die einzelnen Parameter voneinander abhängig sind und die Orthonormalitätsbedingung einer Rotationsmatrix nicht sichergestellt wird. Sollen nur die externen Parameter bestimmt werden, muss zunächst die Projektionsmatrix mit der inversen Kamerakalibriermatrix korrigiert werden und eine Parameterzerlegung folgen.

9.1.3 Tsai-Lenz

Mit dem Kalibrierverfahren nach Tsai und Lenz [TV98][Tsa87][LT88] können interne und externe Kameraparameter unabhängig voneinander berechnet werden. Daher ist bei bereits bekannten internen Parametern mit diesem Verfahren eine direkte Berechnung der Pose möglich. Im Gegensatz zur DLT wird hier auch die Linsenverzerrung k_1, k_2 berücksichtigt. Das Verfahren kann auf planaren (5 Punkte) und nichtplanaren (7 Punkte) Konfigurationen arbeiten.

Der Algorithmus ist beispielhaft für ein kombiniertes Verfahren aus linearen und iterativen Anteilen und basiert auf zwei Stufen. Zunächst werden Rotationsmatrix R und die Komponenten t_x, t_y des Translationsvektors linear berechnet. Danach folgt eine initiale lineare Lösung für f und t_z, die zusammen mit den Verzerrungsparametern k_1, k_2 durch eine iterative, nichtlineare Optimierung angenähert wird.

Zusätzlich wird der Skalierungsfaktor s eingeführt, um die *aspect ratio* der horizontalen und vertikalen Abstände der Pixelzellen auf dem Ka-

merachip auszugleichen. Er kann jedoch nur bei der Verwendung einer nichtplanaren Punktkonfiguration mit $n \geq 7$ bestimmt werden. Bei der Nutzung einer planaren Punktkonfiguration wird die Tiefenkomponente $z = 0$ gesetzt und die Berechnung mit $n \geq 5$ Punkten durchgeführt, was jedoch mit der Einschränkung der Berechenbarkeit des Skalierungsfaktors einhergeht.

9.1.4 Perspective-n-Point Problem (PnP)

Ein weiterer Ansatz zur Berechnung der Kamerapose beruht darauf, für n gegebene Weltpunkte und ihre korrespondierenden 2D Abbildungen die Länge ihrer Projektionsstrahlen zu bestimmen. Die grundlegende Frage ist, wie viele Punktkorrespondenzen benötigt werden um dieses Problem zu lösen. Für einen oder zwei Weltpunkte gibt es unendlich viele Positionen entlang ihrer homogenen Projektionsstrahlen, welche dasselbe Bild erzeugen. Daher ist das Poseproblem mit weniger als drei Punkten nicht lösbar. Drei Weltpunkte reduzieren die möglichen Lösungen auf bis zu vier. Die Mehrdeutigkeit entsteht dadurch, dass die Objektebene derart um die Achsen AB, BC und AC gedreht werden kann, sodass unter Beibehaltung der abgebildeten Projektionsstrahlen und der Abstände der Punkte untereinander, ein jeweils zweiter Schnittpunkt der Weltpunkte mit dem Projektionsstrahl entsteht, der dasselbe Bild erzeugt (Abbildung 65).

In Abbildung 66 sind Anordnungen von Kamerazentrum O und drei gegebenen Weltpunkten aus der Sicht der Kamera aufgeführt. Eine Mehrdeutigkeit entsteht genau dann, wenn sich die Kamera in einer Ebene befindet, welche senkrecht auf der Objektebene steht und dabei die Höhe einer Dreiecksseite enthält. Keine weitere Lösung entsteht, wenn die Entfernung der Kamera zu der Objektebene gering ist, da in diesem Fall der neue

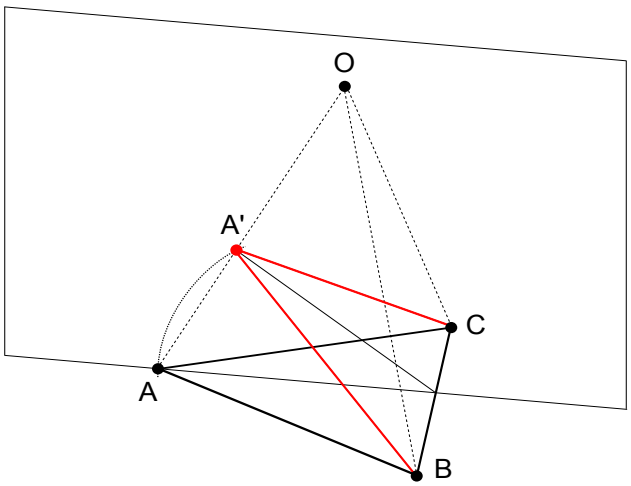


Abbildung 65: Mehrdeutigkeit von 3 Punkten

Schnittpunkt hinter der Kamera liegen kann. Wenn die Kamera senkrecht über einem Weltpunkt oder außerhalb der Objektebene steht, ergibt sich nur eine Lösung, da der Projektionsstrahl nur tangiert, jedoch kein weiteres Mal geschnitten wird.

Ist die Kamera direkt senkrecht über dem Orthozentrum der Dreicksebene positioniert, und somit im Schnitt aller Ebenen aus den Dreieckshöhen, so ergeben sich durch drei weitere entstehende Schnittpunkte insgesamt genau vier Lösungen. Einen Sonderfall bildet eine Anordnung der Weltpunkte im rechtwinkligen Dreieck. Hier befindet sich das Orthozentrum im rechten Winkel. Ist die Kamera über dem rechten Winkel, können nur bis zu drei Lösungen entstehen, da die Dreieckshöhe gegenüber dem rechten Winkel den senkrechten Projektionsstrahl nur tangiert und kein weiterer Schnittpunkt entsteht.

Eine Analyse der möglichen kritischen Punkt-Kamera Anordnungen wird in [WMSM91] gegeben. Am wahrscheinlichsten ist demnach, dass bis zu zwei Lösungen auftreten. In [ZH05] und [ZH06] werden weitere geometrische Bedingungen für das Auftreten mehrerer Lösungen aufgezeigt.

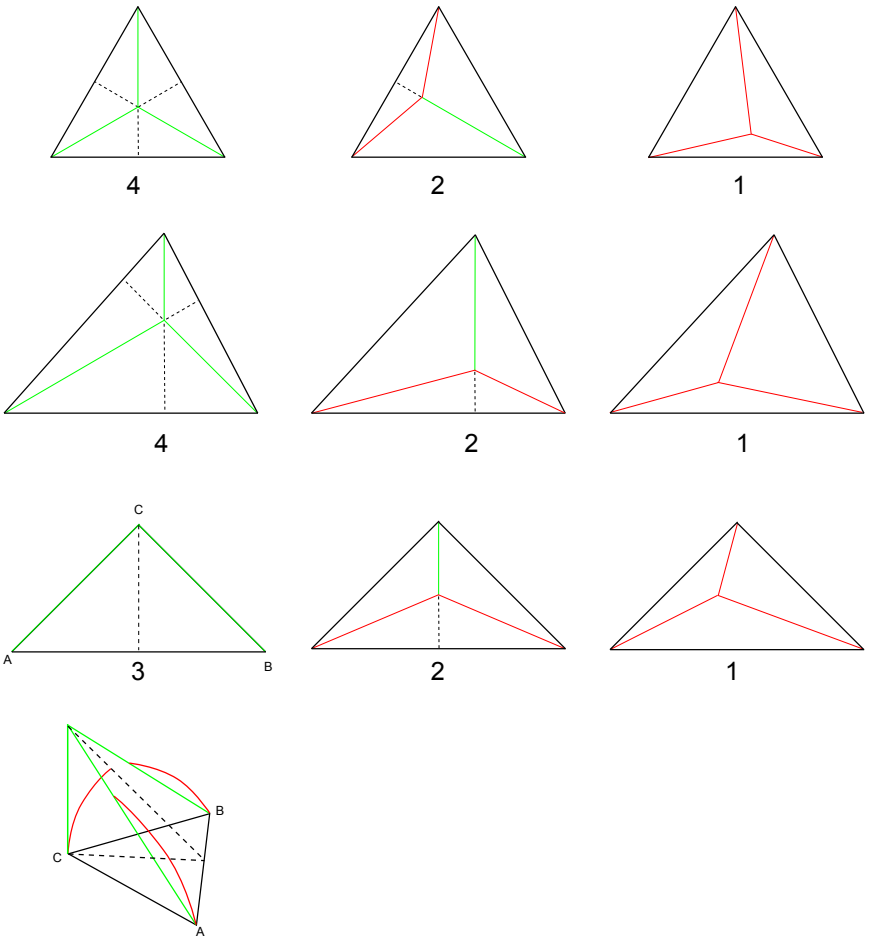


Abbildung 66: Kamera-Punkt-Anordnungen und mögliche Lösungen

Wenn sich das Kamerazentrum in der senkrechten Ebene mit einer Dreieckshöhe befindet und seine Projektion gleichzeitig innerhalb des Umkreises der Objektebene (*danger cylinder*) liegt, dann existieren vier Lösungen. Liegt das Kamerazentrum genau auf dem Umkreis, existieren drei Lösungen.

Sind drei Punkte gegeben (P3P) so bestimmt sich die Länge der Projektionsstrahlen vom Kamerazentrum zu den Weltpunkten wie folgt: Ihre Abstände $x = AB$, $y = BC$ und $z = AC$ untereinander, sowie die Winkel $\alpha = a \sphericalangle b$, $\beta = b \sphericalangle c$ und $\gamma = a \sphericalangle c$ zwischen jedem Paar von Weltpunkten sind bekannt. Gesucht sind die Längen der Verbindungen $a = OA$, $b = OB$ und $c = OC$ zwischen Weltpunkt und Kamerazentrum, welche per Kosinussatz erlangt werden können:

$$x^2 = a^2 + b^2 - 2ab\cos(\alpha)$$

$$y^2 = b^2 + c^2 - 2bc\cos(\beta)$$

$$z^2 = a^2 + c^2 - 2ac\cos(\gamma).$$

Das Gleichungssystem kann mit einem biquadratischen Polynom direkt gelöst werden. Auch wurde eine geometrische Lösung vorgeschlagen, bei der die Weltpunkte entlang ihrer Projektionsstrahlen iterativ verschoben werden, bis alle drei Seitenlängen der Dreiecke mit den Abständen der Weltpunkte übereinstimmen [FB81]. In [HLON94] ist eine Übersicht verschiedener Lösungsansätze beschrieben.

Sind die Entfernungen der Weltpunkte entlang ihrer Projektionsstrahlen vom Kamerazentrum O aus bestimmt, kann dessen Lage gemäß Abbildung 67 in Weltkoordinaten berechnet werden. Durch Projektion der Strahlen auf die Ebene der Weltpunkte werden die Punkte L' und L'' konstruiert. Sie definieren Ebenen, die senkrecht auf den Verbindungen der Weltpunkte

stehen und einen gemeinsamen Schnittpunkt R haben, der die kürzeste Entfernung der Kamera von der Objektebene darstellt.

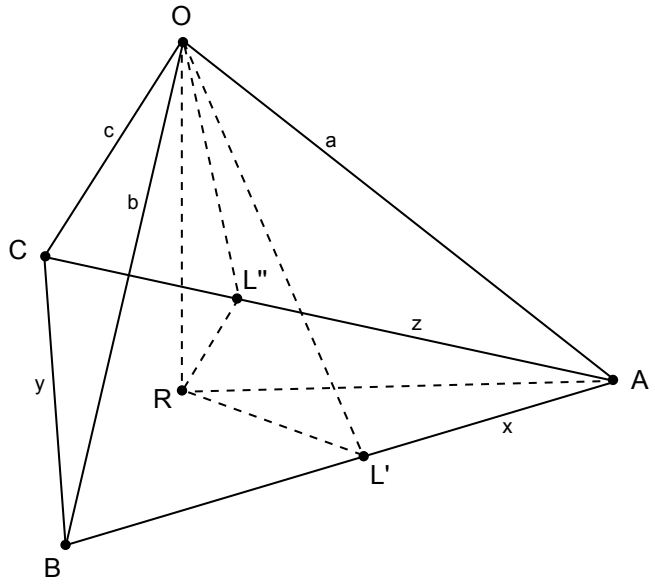


Abbildung 67: P3P Position

Die Rotation der Kamera wird durch die Orientierung der Bildebene in Weltkoordinaten bestimmt. Mit der Brennweite f werden die Abstände zwischen Kamerazentrum und jeweiligem Bildpunkt $|OA'|, |OB'|$ und $|OC'|$ auf ihren Projektionsstrahlen OA, OB und OC bestimmt (Abbildung 68). Mit den so erlangten Weltkoordinaten der Bildpunkte kann die Gleichung der 3D Bildebene aufgestellt werden. Die Normale der Ebene durch das Kamerazentrum ergibt die Z-Achse der Kamera und den Bildhauptpunkt H. Um die endgültige Orientierung zu erlangen, muss noch ein Vektor HA' vom Bildhauptpunkt zu einem der 3D Bildpunkte berechnet werden.

Es ist möglich, den P3P Ansatz auf größere Mengen von Korrespondenzen zu erweitern, indem jeweils für Gruppen von drei Korrespondenzen ihre Lösungen mit P3P Verfahren errechnet werden und diese auf Überein-

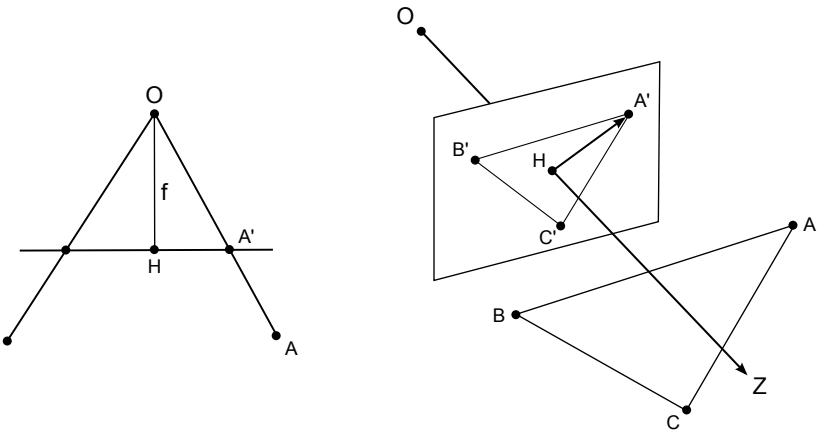


Abbildung 68: P3P Orientierung

stimmungen hin überprüft werden. Statt der Lösung mit Polynomen, ist es auch möglich, die Koeffizienten per SVD zu berechnen [QL99]. Horaud [HCLL89] löst das P4P Problem mit Ersetzung von vier nicht koplanaren Punkten durch ein Bündel von drei Linien. Statt nach der Tiefe zu lösen, umschreibt Lepetit [LMNF09] vier oder mehr 3D Weltpunkte als gewichtete Summe von vier nicht koplanaren oder drei planaren Kontrollpunkten und löst das Problem nach ihren Koordinaten. Ab sechs Punkten ergibt sich eine eindeutige Lösung, da mit 12 resultierenden Gleichungen alle 12 Einträge der Transformationsmatrix bestimmt werden können.

9.1.5 POSIT

Der POSIT-Algorithmus (Pose from Orthography and Scaling with Iterations) von DeMenthon und Davis [DD95] berechnet die Pose aus den Korrespondenzen von mindestens 4 nichtplanaren Punkten mit linearen Iterationen. Bei jedem Iterationsschritt werden die Objektpunkte schwachperspektivisch projiziert und der Skalierungsfaktor gegenüber den perspektivischen Bildpunkten aus dem gegebenen Kamerabild berechnet. Der dabei

gemessene Fehler dient der Konstruktion einer neuen Pose und damit einer korrigierten Perspektive. Wenn der Fehler konvergiert, entspricht damit die gefundene Pose einer Annäherung der perspektivischen Projektion.

Die schwach-perspektivische Projektion besteht aus einer orthographischen Projektion auf eine zur Bildebene parallelen Ebene mit Tiefe z_0 und anschließender perspektivischer Projektion von dieser Ebene aus auf die Bildebene. Dies entspricht einer skaliert-orthographischen Projektion, bei der auf die orthographische Projektion eine Skalierung folgt:

$$\begin{pmatrix} f\frac{x}{z_0} \\ f\frac{y}{z_0} \\ 1 \end{pmatrix} = \begin{pmatrix} fx \\ fy \\ z_0 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & z_0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Als Bedingung für die Konstruktion der schwach-perspektivischen Projektion wird davon ausgegangen, dass die Tiefe des abgebildeten Objekts im Verhältnis zur Entfernung von der Kamera gering ist und das Objekt möglichst zentral abgebildet wird. Der Skalierungsfaktor $s = \frac{f}{z_0}$ bestimmt dabei das Verhältnis zwischen orthographischer Abbildung der Objektpunkte auf z_0 und perspektivischer Abbildung von z_0 auf die Bildebene. Anders ausgedrückt ist s die Skalierung der Länge des senkrecht auf z_0 projizierten Vektors $p_w - p_{w0}$ zur Länge des Vektors $p'_i - p_{i0}$ in der Bildebene. Der Punkt p_{i0} ist die Abbildung des Objektursprungs p_{w0}, der Punkt p'_i die schwach-perspektivische Abbildung des Objektpunktes p_w (Abbildung 69).

Für die schwach-perspektivische Projektion von Welt- in Bildkoordinaten gilt

$$u = f\frac{p_w \cdot i + t_x}{p_w \cdot k + z_0}$$

$$v = f\frac{p_w \cdot j + t_y}{p_w \cdot k + z_0}.$$

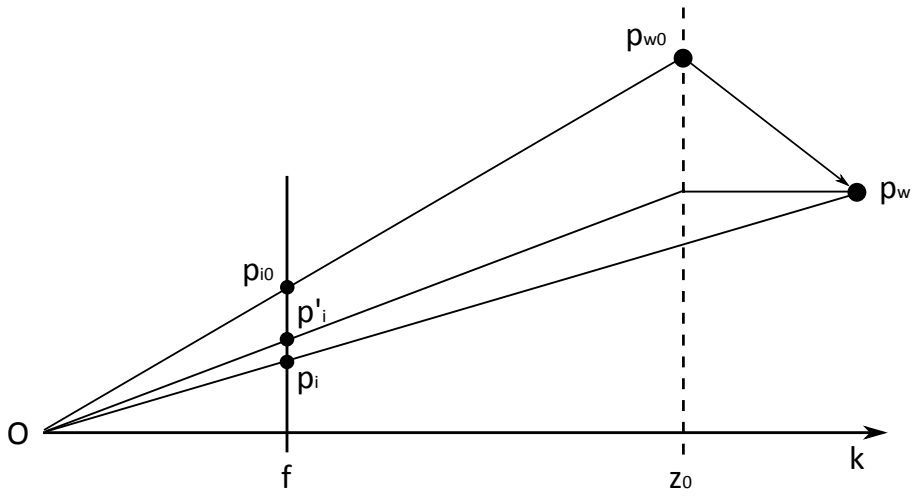


Abbildung 69: Schwach-perspektivische Projektion

Division durch z_0 und Einsetzung des Skalierungsfaktors s ergibt

$$u = \frac{p_w \cdot si + u_0}{1 + \epsilon}$$

$$v = \frac{p_w \cdot sj + v_0}{1 + \epsilon}$$

mit dem abgebildeten Objektursprung

$$u_0 = f\frac{t_x}{z_0}, v_0 = f\frac{t_y}{z_0}$$

und mit dem Verhältnis der Tiefe des Objektpunktes zur Entfernung der Ebene z_0 von der Kamera

$$\epsilon = \frac{p_w \cdot k}{z_0}.$$

Für die bekannten Korrespondenzen zwischen Objekt- und Bildpunkten

gilt durch Umformung daher

$$p_w \cdot si = u(1+\epsilon) - u_0$$

$$p_w \cdot sj = v(1+\epsilon) - v_0$$

wobei die linke Seite die schwach-perspektivische Abbildung der gegebenen Objektpunkte beschreibt und die rechte Seite die korrespondierenden schwach-perspektivischen Bildpunkte u' und v' durch Skalierung der perspektivischen Koordinaten u und v annähert:

$$u' = u(1+\epsilon)$$

$$v' = v(1+\epsilon).$$

Epsilon ist dabei als homogene Komponente zu verstehen, die einer Verschiebung der Bildpunkte in Richtung Projektionsstrahlen entspricht. Zu Beginn des Algorithmus sind die perspektivischen und schwach-perspektivischen Bildpunkte identisch ($\epsilon = 0$). Die Tiefenebene z_0 liegt im Objektursprung. Aus diesen Gleichungen wird ein lineares Gleichungssystem für alle Korrespondenzen aufgestellt, indem zunächst einmalig eine Objektmatrix A mit den Koordinaten der Objektpunkte als Zeilen gebildet und deren Pseudoinverse A^+ errechnet wird. Für jedes Bild werden die korrespondierenden Bildpunkte B bezüglich des abgebildeten Objektursprungs $p_i - p_{i0}$ bestimmt und diese mit der vorherberechneten Pseudoinversen der Objektpunkte multipliziert:

$$A = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ ... & ... & ... \\ x_n & y_n & z_n \end{pmatrix} \quad l = \begin{pmatrix} si_x & sj_x \\ si_y & sj_y \\ si_z & sj_z \end{pmatrix} \quad B = \begin{pmatrix} u'_1 - u_0 & v'_1 - v_0 \\ u'_2 - u_0 & v'_2 - v_0 \\ ... & ... \\ u'_n - u_0 & v'_n - v_0 \end{pmatrix}$$

$$Al = B$$

$$l = A^{+}B.$$

Das Ergebnis sind die Vektoren i und j der Rotationsmatrix und nach Normierung k aus ihrem Kreuzprodukt. Damit ist die Approximation der Rotation vom Objekt- in das Kamerakoordinatensystem bestimmt. Der Skalierungsfaktor s der Projektion berechnet sich aus der gemittelten Norm der beiden Vektoren

$$s_1 = \sqrt{i * i}, \;\; s_2 = \sqrt{j * j}, \;\; s = \frac{s_1 + s_2}{2}.$$

Als Nächstes wird aus dem Skalierungsfaktor s der aktuellen Projektion die Tiefe der Projektionsebene z_0 bestimmt

$$z_0 = \frac{f}{s}$$

und damit ein neues ϵ berechnet, welches als Fehlermaß dient. Liegt die Änderung des Fehlers $|\epsilon_i - \epsilon_{i-1}|$ über einem Schwellwert, so wird in der folgenden Iteration die Rotation mit einer durch ϵ korrigierten schwachperspektivischen Projektion im Bild wiederholt, bis die Änderung des Fehlers konvergiert.

Die Translation ergibt sich durch Skalierung der Abbildung des Vektors

vom Kameraursprung O zum Ursprung des Objekts p_{w0} mit s

$$t_x = \frac{u_0}{s}, \;\; t_y = \frac{v_0}{s}, \;\; t_z = z_0.$$

Die Nutzung vieler Korrespondenzen erhöht die Genauigkeit des Ergebnisses, sofern sie sich nicht in planarer Konfiguration befinden. Die Anzahl der Iterationen bis zur Konvergenz beträgt laut den Autoren 4-5 Schritte und wird damit als echtzeitfähig bezeichnet. Ein initialer Startwert wird nicht benötigt. Der Algorithmus konvergiert langsam oder gar nicht, wenn die schwach-perspektivische Bedingung nicht hinreichend erfüllt ist, also wenn sich das Objekt nahe der Kamera befindet oder weit von der optischen Achse entfernt ist. In diesem Fall treten verstärkt perspektivische Effekte auf, die sich nicht mit der schwachen Perspektive annähern lassen. Es existiert eine Erweiterung, die auch auf koplanaren Punktekonfigurationen arbeitet [ODD96]. In [HCDL95] wird die Pose mit paraperspektivischer Projektion beschrieben, welche weniger Einschränkungen gegenüber der schwachen Perspektive hat und besser konvergiert.

Ein von Christy und Horaud [CH99][HDLC97] vorgestelltes Verfahren dient der iterativen Posefindung aus koplanaren oder nicht koplanaren Konfigurationen von 3D Linien unter Zuhilfenahme schwach-perspektivischer und paraperspektivischer Darstellung. In Anlehnung an POSIT konvergiert das Verfahren mit drei bis fünf Iterationen sehr schnell, unabhängig von der Anzahl der Korrespondenzen. Der paraperspektivische Fall benötigt weniger Iterationen, ist aber schwerer zu implementieren, weswegen die schwach-perspektivische Anwendung von den Autoren bevorzugt wird. Gegenüber der Verwendung von Punktkorrespondenzen (POSIT) erzielt die iterativ schwach-perspektivische Berechnung mit Linien die exaktere Pose.

Eine Linie im 3D ist gegeben durch den Referenzpunkt $\mathbf{O}_i$ und die

Richtung der Linie $\mathbf{D}_i$. Jeder Punkt $\mathbf{P}_i$ auf der Linie kann durch den Parameter $\lambda \in [0, \infty]$ erreicht werden ($\mathbf{P}_i = \mathbf{O}_i + \lambda \mathbf{D}_i$). Die Projektion eines Linienpunktes in ideale Bildkoordinaten wird beschrieben als

$$\mathbf{P}_i = \begin{pmatrix} I_x & I_y & I_z & x_0 \\ J_x & J_y & J_z & y_0 \\ K_x & K_y & K_z & 1 \end{pmatrix} \begin{pmatrix} O_x \\ O_y \\ O_z \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} I_x & I_y & I_z & x_0 \\ J_x & J_y & J_z & y_0 \\ K_x & K_y & K_z & 1 \end{pmatrix} \begin{pmatrix} D_x \\ D_y \\ D_z \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{I} \cdot \mathbf{O}_i + x_0 \\ \mathbf{J} \cdot \mathbf{O}_i + y_0 \\ 1 + \eta_i \end{pmatrix} + \lambda \begin{pmatrix} \mathbf{I} \cdot \mathbf{D}_i \\ \mathbf{J} \cdot \mathbf{D}_i \\ \mu_i \end{pmatrix}$$

mit

$$\mathbf{I} = \frac{\mathbf{i}}{t_z} \qquad \mathbf{J} = \frac{\mathbf{j}}{t_z} \qquad \mathbf{K} = \frac{\mathbf{k}}{t_z} \qquad x_0 = \frac{t_x}{t_z} \qquad y_0 = \frac{t_y}{t_z}$$

und

$$\eta_i = \mathbf{K} \cdot \mathbf{O}_i \quad \text{und} \quad \mu_i = \mathbf{K} \cdot \mathbf{D}_i.$$

Eine korrespondierende Linie im Bild ist durch ihre Normale in der impliziten Geradengleichung

$$a_i x + b_i y + c_i = 0$$

charakterisiert. Durch Einsetzen in die Projektionsgleichung und Umformung erhält man für n Linienkorrespondenzen $2n$ Gleichungen

$$a_i \mathbf{I} \cdot \mathbf{O}_i + b_i \mathbf{J} \cdot \mathbf{O}_i + a_i x_0 + b_i y_0 + c_i (1 + \eta_i) = 0$$

$$a_i \mathbf{I} \cdot \mathbf{D}_i + b_i \mathbf{J} \cdot \mathbf{D}_i + c_i \mu_i = 0$$

deren Unbekannte $\mathbf{I}, \mathbf{J}, x_0, y_0, \eta_i$ und μ_i die Poseparameter enthalten. Entfällt der perspektivische Effekt durch $\eta_i = \mu_i = 0$, dann erhält man die Gleichungen für die schwach-perspektivische Projektion

$$a_i \mathbf{I} \cdot \mathbf{O}_i + b_i \mathbf{J} \cdot \mathbf{O}_i + a_i x_0 + b_i y_0 + c_i = 0$$

$$a_i \mathbf{I} \cdot \mathbf{D}_i + b_i \mathbf{J} \cdot \mathbf{D}_i = 0.$$

Diese Gleichungen sind linear und beinhalten 8 Unbekannte, sodass zur Lösung des Poseproblems mindestens 4 Lininenkorrespondenzen benötigt werden. Da sich der Algorithmus über die schwach-perspektivische Darstellung der korrekten perspektivischen Abbildung annähert, wird im ersten Schritt der perspektivische Effekt mit $\eta_i = 0$ und $\mu_i = 0$ für alle Gleichungen vernachlässigt. Durch iteratives Lösen des linearen Gleichungssystems

$$\begin{pmatrix} \dots & \dots & \dots & \dots \\ a_i \mathbf{O}_i^T & b_i \mathbf{O}_i^T & a_i & b_i \\ a_i \mathbf{D}_i^T & b_i \mathbf{D}_i^T & 0 & 0 \\ \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} \mathbf{I}^T \\ \mathbf{J}^T \\ x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} \dots \\ -c_i(1+\eta_i) \\ -c_i \mu_i \\ \dots \end{pmatrix}$$

erhält man nun eine Annäherung von perspektivischem Translationsvektor (t_x, t_y, t_z) und den Zeilen $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ der Rotationsmatrix:

$$t_z = \frac{1}{2}\left(\frac{1}{|\mathbf{I}|} + \frac{1}{|\mathbf{J}|}\right), \quad t_x = x_0 t_z, \quad t_y = y_0 t_z.$$

Nach Orthogonalisierung der Rotationsmatrix werden für alle Linienkorrespondenzen neue

$$\eta_i = \frac{\mathbf{k} \cdot \mathbf{O}_i}{t_z} \quad \text{und} \quad \mu_i = \frac{\mathbf{k} \cdot \mathbf{D}_i}{t_z}$$

berechnet. Sie korrigieren die schwach-perspektivische Darstellung für die nächste Iteration in Richtung korrekte Perspektive und dienen als Konvergenzkriterium. Wenn die Änderung zum Ergebnis der vorherigen Iteration gering genug ist oder die maximale Iterationszahl erreicht ist, bricht der Algorithmus ab.

Die Auswertung der Autoren ergab, dass die Fehlerrate bei der Poseberechung mit Linien auf ein Drittel gegenüber der Verwendung von Punktkorrespondenzen gesenkt werden konnte. Voraussetzung ist, dass die Gleichungen der Matrix **A** linear unabhängig sind. Dies ist nicht der Fall, wenn sich bei einem Minimum von vier nicht koplanaren Linien drei in einem gemeinsamen Punkt schneiden oder drei von ihnen parallel sind. Ist die Linienkonfiguration koplanar, reichen drei Linien zur Poseberechnung, welche weder einen gemeinsamen Schnittpunkt haben, noch parallel angeordnet sein dürfen.

9.1.6 Fluchtpunkte

Werden parallele Geraden im 3D Raum durch perspektivische Projektion abgebildet, so schneiden sie sich in einem Punkt der Abbildung, dem Fluchtpunkt. Für Geraden, die parallel zur Bildebene liegen, befindet sich der Fluchtpunkt im Unendlichen. Fluchtpunkte sind nur von der Orientierung der Kamera und den internen Parametern abhängig, jedoch nicht von der Translation. Diese kann im Gegensatz zur Rotation folglich nicht direkt mit Hilfe von Fluchtpunkten bestimmt werden.

Der Schnittpunkt einer Geraden mit der Bildebene, die parallel zu einer Weltgeraden verläuft und im Kamerazentrum liegt, definiert den Fluchtpunkt **v**. Diese Gerade entspricht dem Richtungsvektor **d** aller Weltgeraden, deren Abbildungen sich in diesem Fluchtpunkt schneiden (Abbildung 70).

Der Fluchtpunkt ist die Projektion eines Punktes, der unendlich in Richtung **d** verschoben ist: $\mathbf{v} = \mathbf{Kd}$.

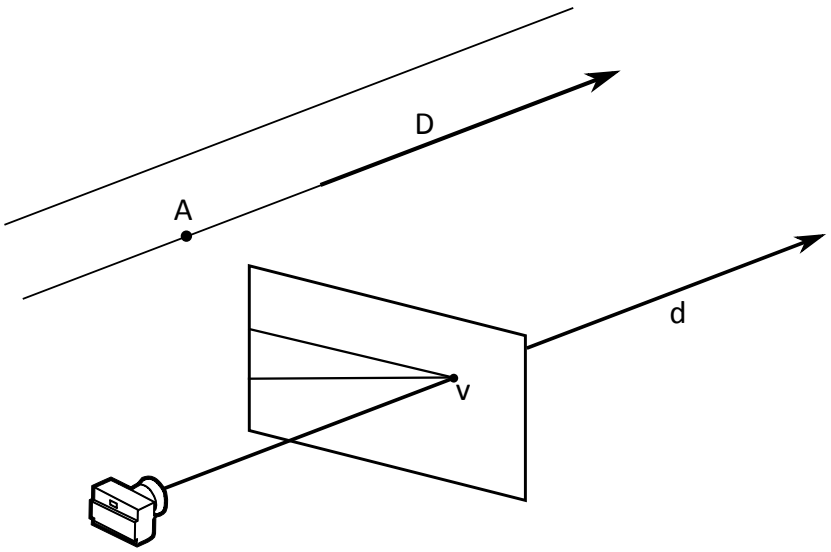


Abbildung 70: Konstruktion des Fluchtpunktes

Bei drei orthogonalen Weltgeraden, die sich in einem Punkt schneiden, nimmt man den Ursprung des Weltkoordinatensystems in genau diesem Schnittpunkt an. Die Abbildung der Geraden definiert ihre Fluchtpunkte im Bild. Die normierten Richtungsvektoren d_1, d_2, d_3 der Geraden vom Kamerazentrum durch die Fluchtpunkte bilden eine ebenfalls orthogonale Basis des Kamerakoordinatensystems und damit der Rotationsmatrix R [Ech90]:

$$R = (D_1, D_2, D_3)$$

mit

$$D_1 = \frac{d_1}{|d_1|}, \;\; D_2 = \frac{d_2}{|d_2|}, \;\; D_3 = \frac{d_3}{|d_3|}.$$

Da die Bedingungen von Parallelität und Orthogonalität gelten, eignet

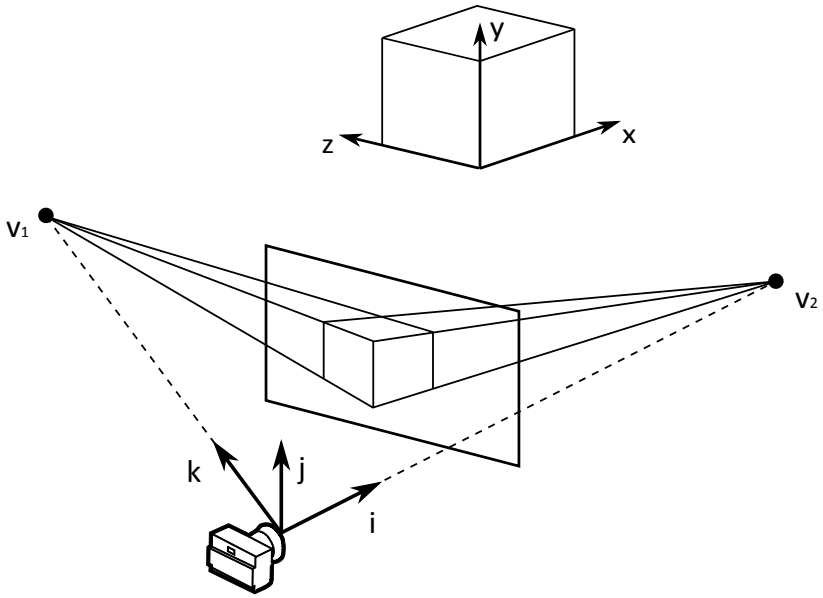


Abbildung 71: Pose aus Fluchtpunkten (Vereinfacht nur zwei dargestellt).

sich das Fluchtpunktverfahren besonders für architektonische Szenen. Die Rotationsmatrix zwischen zwei Bildern kann mit drei nicht kollinearen korrespondierenden Fluchtpunkten berechnet werden (Abbildung 71). Aus den Korrespondenzen der normierten Fluchtpunktrichtungen d und d' werden lineare Gleichungen aufgestellt, deren Lösung die Rotationsmatrix R ist. Die Spalten der orthogonalen 3x3 Matrizen D und D' bestehen aus den Vektoren d und d' der drei Fluchtpunktrichtungen:

$$D' = RD$$

$$R = D'D^T$$

$$d = \frac{K^{-1}v}{|K^{-1}v|}.$$

Die benötigten Richtungsvektoren durch die Fluchtpunkte können durch

Rückprojektion mit der Kalibriermatrix K ermittelt werden. Die Translation kann bei bekannter Rotation durch Punktkorrespondenzen in beiden Bildern berechnet werden [HZ10][CT90].

9.1.7 Pose aus Marker

Exemplarisch soll anhand des ARToolKit [KB99] die markerbasierte Poseberechnung vorgestellt werden. Sowohl die internen Kameraparameter der Kalibriermatrix K als auch die Größe eines quadratischen Markers sind bekannt. Der Ursprung des Markerkoordinatensystems befindet sich in seinem Zentrum und liegt in der Ebene $z = 0$. Der Marker wird erkannt durch Binarisierung des Kamerabildes. Regionen im Bild, die durch vier umschließende Geraden beschrieben werden können, werden als potentielle Marker angenommen. Die gefundenen Regionen im Bild werden normalisiert und das Label jeder Region wird durch Mustererkennung (Pattern Matching) mit gespeicherten Daten verglichen, der Marker wird so eindeutig identifiziert. Die Parameter der vier Geraden sowie die Koordinaten ihrer Schnittpunkte, welche die Markereckpunkte darstellen, werden in Bildkoordinaten ermittelt. Die Projektion je zweier paralleler Seiten des Markers in das Kamerabild ergibt Bildgeraden mit den Gleichungen

$$a_1 x + b_1 y + c_1 = 0, \quad a_2 x + b_2 y + c_2 = 0.$$

Durch Projektion mit der bekannten Kameramatrix K lassen sich so zwei Ebenen in Kamerakoordinaten definieren, die zwei Seiten eines Frustums zwischen Kamera und Marker aufspannen und deren Schnittgeraden mit der Bildebene genau die Projektionen der parallelen Seiten des Markers darstellen:

$$a_1K_{11}x + (a_1K_{12} + b_1K_{22})y + (a_1K_{13} + b_1K_{23} + c_1)z = 0$$

$$a_2K_{11}x + (a_2K_{12} + b_2K_{22})y + (a_2K_{13} + b_2K_{23} + c_2)z = 0$$

mit

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ 0 & K_{22} & K_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

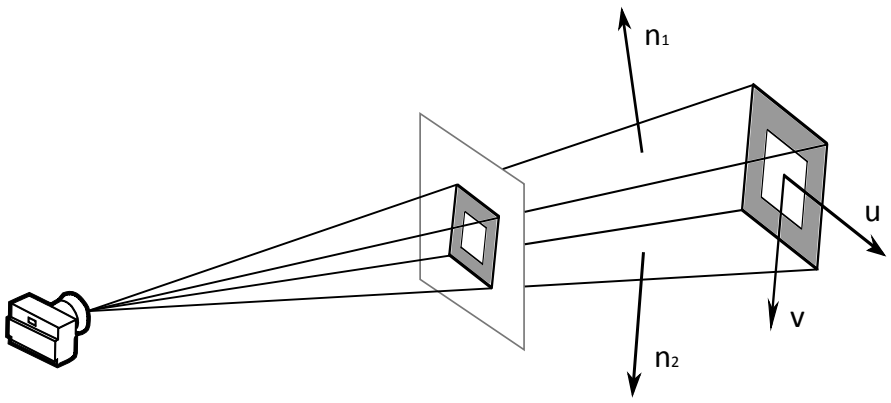


Abbildung 72: Pose aus Marker

Die Normalen $\mathbf{n}_1 = (a_1, b_1, c_1)^T$ und $\mathbf{n}_2 = (a_2, b_2, c_2)^T$ dieser beiden Ebenen des Frustums entsprechen denen der Geraden im Bild. Ihr Kreuzprodukt ergibt den Richtungsvektor der zwei parallelen Seiten des Markers. Aus den vier umschließenden Geraden werden so zwei Richtungsvektoren **u** und **v** gewonnen, die zwei Koordinatenachsen des Markers bilden und

orthogonalisiert werden müssen (Abbildung 72). Die dritte Raumachse **k** bildet ihr Kreuzprodukt, somit ist die Rotationsmatrix **R** bekannt.

Aus den 4 Korrespondenzen der Eckpunkte des Markers und ihren Abbildungen in Bildkoordinaten lassen sich mit Hilfe von **K** und **R** acht lineare Gleichungen aufstellen, aus deren Lösung der Translationsvektor t bestimmt wird. Die Parameter der Rotation werden in Folge über die Minimierung der Summe der Differenzen zwischen Markereckpunkten **u** und **v** im Bild und den von der geschätzten Pose aus rückprojizierten Markereckpunkten $\hat{\mathbf{u}}$ und $\hat{\mathbf{v}}$ optimiert:

$$err = \frac{1}{4} \sum_{i=1}^{4} \Big((\mathbf{u}_i - \hat{\mathbf{u}})^2 + (\mathbf{v}_i - \hat{\mathbf{v}})^2 \Big).$$

9.2 RANSAC

Ein mögliches Vorgehen zur Minimierung der fehlerhaften Daten in einer Menge ist die Bestimmung der initialen Modellparameter aus einer möglichst großen Gesamtmenge von Messungen mit anschließender iterativer Verkleinerung der Messdaten. Dazu werden in jedem Schritt diejenigen Daten eliminiert, die am weitesten entfernt vom Modelloptimum liegen. Dies wird wiederholt, bis die größte festgestellte Abweichung unter einen Schwellwert fällt oder die restliche Datenmenge zu klein wird. Fischler und Bolles [FB81] haben jedoch gezeigt, dass dieser Ansatz nicht in allen Fällen zu den optimalen Modellparametern führt. Sie schlagen stattdessen einen eigenen Algorithmus zur Bildung einer ausreißerfreien Menge von Messdaten vor - den Random Sample Consensus (RANSAC). Anstatt die komplette Datenmenge um die Ausreißer zu reduzieren, wird eine kleine initiale Menge von Messwerten um konsistente Daten erweitert, bis ein Optimum gefunden ist. Gesucht ist eine Untermenge der Messdaten, die durch das ihnen zugrunde liegende Modell am besten beschrieben wird. Diejenigen Daten, welche von dem Modell abweichen, werden *Outlier* genannt. Alle dem Modell entsprechenden Daten *Inlier* (Abbildung 73). Bei der Berechnung einer Kamerapose gilt es, falsche und schlecht lokalisierte Punktkorrespondenzen als Outlier zu filtern. Durch dieses Vorgehen wird eine Berechnung auf den Korrespondenzen robuster, da fehlerhafte Daten keinen Einfluss auf das Ergebnis haben.

Begonnen wird mit einer möglichst kleinen, zufällig verteilten Anzahl von Werten (*random sample*), die als Probe aus der Menge aller Korrespondenzen ausgesucht wird. Es wird angenommen, dass diese fehlerfrei sei. Die Größe der Probe hängt dabei von der Parametrierung ab. Mit der Probe werden die Poseparameter berechnet und für alle gemessenen Korrespondenzen

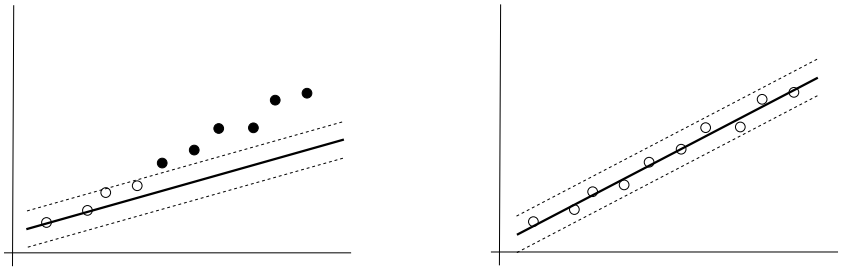


Abbildung 73: RANSAC: links Probe mit 4 Inliern, rechts Probe mit 10 Inliern

der Fehler zu den ermittelten Modelldaten bestimmt. Ist dieser größer als ein vorgegebener Schwellwert, wird die entsprechende Korrespondenz als Outlier eingestuft, ansonsten als Inlier. Eine nur aus Inliern bestehende Probe wird *consensus set* genannt. Dieses Vorgehen wird mit weiteren Proben wiederholt und jeweils die Anzahl der Inlier, der *Support* der Probe, verglichen. Der Algorithmus kann entweder abgebrochen werden, wenn eine Probe mit ausreichend großer Anzahl an Inliern gefunden ist, oder es wird nach einer festgelegten Anzahl von Iterationen die Probe mit der bis dahin maximalen Menge an Inliern ausgewählt, welche dementsprechend idealerweise keine Outlier mehr enthält. Mit dieser optimalen Probe wird anschließend die Poseberechnung durchgeführt.

Literatur- und Quellenverzeichnis

[Ach08] S. Achilles. Markerloses Tracking unter Verwendung von Analyse durch Synthese auf Basis von Featuredetektoren. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2008.

[AD03] A. Ansar and K. Daniilidis. Linear Pose Estimation from Points or Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):578–589, 2003.

[Age89] Defense Mapping Agency. Technical Manual 8358.2 The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS), 1989.

[Alt02] Walter Alt. *Nichtlineare Optimierung*. vieweg, 2002.

[ATea20] R. Azzam, T. Taha, and S. Huang et al. Feature-based visual simultaneous localization and mapping: a survey. *SN Applied Sciences 2, 224 (2020)*, 2(224), 2020.

[Azu97] Ronald T. Azuma. A Survey of Augmented Reality. *Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[BAM08] J. Baerz, O. Abert, and S. Mueller. Interactive particle tracing in dynamic scenes consisting of NURBS surfaces. In *IEEE/EG Symposium on Interactive Ray Tracing*, 2008.

[BB95] S.S. Beauchemin and J. L. Barron. The computation of optical flow. In *ACM Computing Surveys 27(3)*, pages 433–466, 1995.

[BC12] J.A. Brown and D.W. Capson. A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle

Filter. In *IEEE Transactions on Visualization and Computer Graphics 18*, pages 68–80, 2012.

[BETG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Computer Vision and Image Understanding (CVIU) 110(3)*, pages 346–359, 2008.

[BFH+61] C. G. Bell, H. Fujisaki, J. M. Heinz, K. N. Stevens, and A. S. House. Reduction of Speech Spectra by Analysis-by-Synthesis Techniques. In *The Journal of the Acoustical Society of America*, volume Vol. 33, 1961.

[BK08] Gary R. Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly, 2008.

[BM11] A. Braun and S. Müller. GPU-assisted 3D Pose Estimation Under Realistic Illumination. In *18th WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2011.

[Bou89] P. Bouthemy. A Maximum Likelihood Framework for Determining Moving Edges. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 11, pages 499–511, 1989.

[Bre65] J.E. Bresenham. Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[Can86] J. Canny. A Computational Approach To Edge Detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence 8(6)*, pages 679–689, 1986.

[CC13] C. Choi and H. I. Christensen. RGB-D Object Tracking: A Particle Filter Approach on GPU. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[CCC+16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[CH99] S. Christy and R. Horaud. Iterative pose computation from line correspondences. *Computer Vision and Image Understanding*, 73(1):137–144, 1999.

[CH19] Kevin Christensen and Martial Hebert. Edge-Direct Visual Odometry. *CoRR*, abs/1906.04838, 2019.

[CLL05] R. Collins, Y. Liu, and M. Leordeanu. On-Line Selection of Discriminative Tracking Features. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(1):1631–1643, 2005.

[CM06] A.B. Carman and P.D. Milburn. Determining rigid body transformation parameters from ill-conditioned spatial marker co-ordinates. *Journal of Biomechanics*, 39(10):1778–1786, 2006.

[CMC03] A. I. Comport, E. March, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR03)*, pages 36–45, 2003.

[CT90] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4:127–140, 1990.

[DC02] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24(7), pages 932–946, 2002.

[DD95] Daniel F. DeMenthon and Larry S. Davis. Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, 15:123–141, 1995.

[DG99] F. Dornaika and C. Garcia. Pose Estimation using Point and Line Correspondences. *Real-Time Imaging*, 5:215–230, 1999.

[DH72] R.O. Duda and P.E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[DMR18] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[DRP+19] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8084–8093, 2019.

[DST08] A. Dell'Acqua, A. Sarti, and S. Tubaro. 3D Motion from structures of points, lines and planes. *Image and Vision Computing*, 26:529–549, 2008.

[ea94] R. Tönjes et al. Analyse durch Synthese Modellierung von 3D-Objekten in Stereobildfolgen. In *1. Workshop visual computing*, 1994.

[Ech90] T. Echigo. A camera calibration technique using three sets of parallel lines. *Machine Vision and Applications*, 3(3):159–167, 1990.

[ESC13] J. Engel, J. Sturm, and D. Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[ESC14] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*, 2014.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Fer04] R. Fernando. *GPU Gems. Programming Techniques, Tips, and Tricks for Real-Time Graphics*. Addison-Wesley, 2004.

[FG87] W. Förstner and E. Gülch. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features. In *Proceedings of ISPRS Intercommission*

Conference on Fast Processing of Photogrammetric Data, pages 281–305, 1987.

[Fia04] M. Fiala. ARTag, An Improved Marker System Based on ARToolkit. Technical report, National Research Council Canada, Publication Number: NRC 47166, 2004.

[FON09] W.T. Fong, S.K. Ong, and A.Y.C. Nee. Computer vision centric hybrid tracking for augmented reality in outdoor urban environments. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, VRCAI 09, pages 185–190, 2009.

[Gai11] C. Gaida. Untersuchung von Verfahren zur Pose-Schätzung im Hinblick auf Analyse durch Synthese. *Bachelorarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2011.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GEM07] T. Grosch, T. Eble, and S. Mueller. Consistent interactive augmentation of live camera images with correct near-field illumination. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2007.

[Gem12] A. Gemmel. Markerloses Tracking unter Verwendung eines hybriden ADS/SLAM-Ansatzes. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2012.

[Gen02] Y. Genc. Markerless Tracking for AR: A Learing-Based Approach. In *International Symposium on Augmented Reality (ISMAR02)*, pages 295–304, 2002.

[GL04] I. Gordon and D. G. Lowe. Scene Modelling, Recognition and Tracking with Invariant Image Feaures. In *3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 110–119, 2004.

[GOMZN+19] R. Gomez-Ojeda, F. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez. PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments. *IEEE Transactions on Robotics*, 35(3):734–746, 2019.

[GT06] X. Gao and J. Tang. On the Probability of the Number of Solutions for the P4P Problem. *Journal of Mathematical Imaging and Vision*, 25(1):79–86, 2006.

[Hab09] T. Habelitz. Markerloses Tracking unter Verwendung von Analyse durch Synthese auf Basis der Ähnlichkeitsbestimmung photorealistischer Bilder. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2009.

[HCDL95] R. Horaud, S. Christy, F. Dornaika, and B. Lamiroy. Object pose: links between paraperspective and perspective. *IEEE International Conference on Computer Vision*, pages 426–433, 1995.

[HCLL89] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, 47(1):33–44, 1989.

[HDLC97] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.

[HLON94] R.M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.

[HS88] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HS90] C. Harris and C. Stennet. RAPID - A Video Rate Object Tracker. In *Proceedings of the British Machine Vision Conference*, pages 73–77, 1990.

[HS12] K. Hirose and H. Saito. Fast Line Description for Line-based SLAM. In *Proceedings of the British Machine Vision Conference*, pages 83.1–83.11, 2012.

[HSK+05] V. Havran, M. Smyk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Importance sampling for video environment maps. In *ACM SIGGRAPH Eurographics Symposium on Rendering*, 2005.

[HW02] Z.Y. Hu and F.C. Wu. A Note on the Number of Solutions of the Noncoplanar P4P Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 2002.

[HZ10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2. Edition, 2010.

[JS04] F. Jarre and J. Stoer. *Optimierung*. Springer, 2004.

[KB99] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR)*, 1999.

[KBK07] K. Koeser, B. Bartczak, and R. Koch. An analysis-by-synthesis camera tracking approach based on free-form surfaces. In *29th annual pattern recognition symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)*, pages 122–131, 2007.

[KBM+15] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 954–962, 2015.

[KH94] R. Kumar and A.R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *Computer Vision Graphics and Image Processing: Image Understanding*, 60(3):313–342, 1994.

[KM06] G. Klein and D. Murray. Full-3D Edge Tracking with a Particle Filter. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1119–1128, 2006.

[KM07] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[KSvA+08] M. Korn, M. Stange, A. von Arb, L. Blum, M. Kreil, K.J. Kunze, J. Anhenn, T. Wallrath, and T. Grosch. Interactive augmentation of live images using a HDR stereo camera. *Journal of Virtual Reality and Broadcasting (JVRB)*, 2008.

[Lev44] K. Levenberg. A Method for the Solution of Certain Problems in Least Squares. *The Quarterly of Applied Mathematics 2*, pages 164–168, 1944.

[LHM00] C.-P. Lu, G.D. Hager, and E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[LMNF09] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[Low91] D.G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13(5), 1991.

[Low92] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. In *International Journal of Computer Vision*, volume 8(2), pages 113–122, 1992.

[Low99] D.G. Lowe. Object Recognition From Local Scale-Invariant Features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

[LT88] R. K. Lenz and R. Y. Tsai. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology. In *IEEE Transactions on Pattern Analysis and Machine Intelligence 10(5)*, 1988.

[MABT11] D. Marimon, T. Adamek, A. Bonnin, and T. Trzcinski. Enhancing global positioning by image recognition, 2011.

[Mar63] D.W. Marquardt. An Algorithm for Least-Square Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics 11*, pages 431–441, 1963.

[MBCM99] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 262–268, 1999.

[Moe99] T.B. Moeslund. The Analysis-by-Synthesis Approach in Human Motion Capture: A Review. In *The 8th Danish conference on pattern recognition and image analysis*, 1999.

[Mor77] H. Moravec. Towards Automatic Visual Obstacle Avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 1977.

[Noh12] M. Nohn. Kameratracking auf Basis eines Partikelfilters nach dem Ansatz Analyse durch Synthese. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2012.

[ODD96] D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Computer vision and image understanding*, 63(3):495–511, 1996.

[OR13] E. Oyallon and J. Rabin. An analysis and implementation of the SURF method, and its comparison to SIFT. In *Image Processing On Line*, 2013.

[PSH09] L. Priese, F. Schmitt, and N. Hering. Grouping of Semantically Similar Image Positions. *Proceedings of the 16th Scandinavian Conference on Image Analysis*, pages 726–734, 2009.

[PTVF92] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[PVA+17] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. PL-SLAM: Real-time monocular visual SLAM with points and lines. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4503–4508, 2017.

[QL99] L. Quan and Z. Lan. Linear N-Point Camera Pose Determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7), 1999.

[RD05] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, pages 1508–1511, 2005.

[RD06] G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings*

of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pages 109–118, 2006.

[RD07] G. Reitmayr and T. Drummond. Initialisation for Visual Tracking in Urban Environments. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–9, 2007.

[RDB01] J. P. Rolland, L. D. Davis, and Y. Baillot. A Survey of Tracking Technology for Virtual Environments. *Fundamentals of Wearable Computers and Augmented Reality*, Chapter 3:67–112, 2001.

[Rei11] B. Reinert. Untersuchung nichtlinearer Methoden zur Berechnung der Kamerapose aus Punkt- und Linienmerkmalskorrespondenzen im Kontext der Analyse durch Synthese. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2011.

[RG06] T. Ritschel and T. Grosch. On-line estimation of diffuse materials. In *3rd Workshop Virtual and Augmented Reality of the GI-Group VR/AR*, 2006.

[RGKM07] T. Ritschel, T. Grosch, J. Kauz, and S. Mueller. Interactive illumination with coherent shadow maps. In *Eurographics Symposium on Rendering (EGSR07)*, 2007.

[Rot03] C. Rother. Linear Multi-View Reconstruction of Points, Lines, Planes and Cameras using a Reference Plane. In *9th IEEE International Conference on Computer Vision (ICCV)*, pages 1210–1217, 2003.

[RSM12a] B. Reinert, M. Schumann, and S. Müller. Combined Non-Linear Pose Estimation from Points and Lines. In *Arbeitsberichte aus dem Fachbereich Informatik, 09/2011, Universität Koblenz-Landau, ISSN (Online) 1864-0850*, 2012.

[RSM12b] B. Reinert, M. Schumann, and S. Müller. Parameter and Configuration Analysis for Non-Linear Pose Estimation with Points and Lines. In *7th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2012.

[SAM07] F. Scheer, O. Abert, and S. Mueller. Towards using realistic ray tracing in augmented reality applications with natural lighting. In *4th Workshop Virtual and Augmented Reality of the GI-Group VR/AR*, 2007.

[SAM09] M. Schumann, S. Achilles, and S. Müller. Analysis by Synthesis Techniques for Markerless Tracking. In *6th Workshop on Virtual and Augmented Reality, GI Workgroup VR/AR*, 2009.

[SB97] S.M. Smith and J.M. Brady. A New Approach to Low Level Image Processing. In *International Journal of Computer Vision*, volume 23(1), pages 45–78, 1997.

[Sch08] M. Schumann. Markerloses Tracking unter Verwendung von Analyse durch Synthese auf Basis der Ähnlichkeitsbestimmung stilisierter Bilder. *Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz*, 2008.

[SEC14] T. Schöps, J. Engel, and D. Cremers. Semi-Dense Visual Odometry for AR on a Smartphone. In *International Symposium on Mixed and Augmented Reality*, 2014.

[SFPG06] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *Workshop on Edge Computing Using New Commodity Architectures (EDGE 2006)*, 2006.

[SHM12] M. Schumann, J. Hoppenheit, and S. Müller. A Matching Shader Technique for Model-Based Tracking. In *20th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2012.

[SHM14] M. Schumann, J. Hoppenheit, and S. Müller. Feature Evaluation and Management for Camera Pose Tracking on 3D Models. In *9th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014.

[SJBP99] J. Strom, T. Jebara, S. Basu, and A. Pentland. Real time tracking and modeling of faces: An EKF-based analysis by synthesis approach. In *IEEE International Workshop on Modelling People*, page 55, 1999.

[SKM12] M. Schumann, S. Kowalczyk, and S. Müller. Initialization of Model-Based Camera Tracking with Analysis-by-Synthesis. In *8th International Symposium on Visual Computing (ISVC)*, 2012.

[SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. In *International Journal of Computer Vision 37(2)*, pages 151–172, 2000.

[SRD06] Paul Smith, Ian Reid, and Andrew Davison. Real-Time Monocular SLAM with Straight Lines. In *British Machine Vision Conference*, volume 1, pages 17–26, 2006.

[ST94] J. Shi and C. Tomasi. Good Features to Track. In *Computer Vision and Pattern Recognition (CVPR94)*, pages 593–600, 1994.

[Str01] D. Stricker. Tracking with Reference Images: A Real-Time and Markerless Tracking Solution for Out-Door Augmented Reality Applications. In *Virtual Reality, Archaeology, and Cultural Heritage (VAST)*, 2001.

[TM08] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, 2008.

[Tsa87] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. In *IEEE Journal of Robotics and Automation 3(4)*, 1987.

[TSR15] P. Tiefenbacher, T. Schulze, and G. Rigoll. Off-the-Shelf Sensor Integration for mono-SLAM on Smart Devices. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 15–20, 2015.

[TV98] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[VLF04] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking.

In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 48–57, 2004.

[WIS+18] Williem, Andre Ivan, Hochang Seok, Jongwoo Lim, Kuk-Jin Yoon, Ikhwan Cho, and In Kyu Park. Visual-Inertial RGB-D SLAM for Mobile Augmented Reality. In *Advances in Multimedia Information Processing – PCM 2017*, pages 928–938. Springer International Publishing, 2018.

[WMSM91] W.J. Wolfe, D. Mathis, C.W. Sklair, and M. Magee. The perspective view of three points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):66–73, 1991.

[WPS07] H. Wuest, A. Pagani, and D. Stricker. Feature Management for Efficient Camera Tracking. In *Proceedings of the 8th Asian conference on Computer vision (ACCV)*, pages 769–778, 2007.

[WS07a] D. Wagner and D. Schmalstieg. ARToolKitPlus for Pose Tracking on Mobile Devices. *Proceedings of 12th Computer Vision Winter Workshop (CVWW)*, pages 139–146, 2007.

[WS07b] H. Wuest and D. Stricker. Tracking of industrial objects by using CAD models. In *Journal of Virtual Reality and Broadcasting 4(1)*, 2007.

[WVS05] H. Wuest, F. Vial, and D. Stricker. Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality*, pages 62–69, 2005.

[WWS07] H. Wuest, F. Wientapper, and D. Stricker. Adaptable Model-Based Tracking Using Analysis-by-Synthesis Techniques. In

Proceedings of the 12th international conference on Computer analysis of images and patterns, pages 20–27, 2007.

[ZDHB08] F. Zhou, F. Duh, B.L. Henry, and M. Billinghurst. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. *Proceedings of ISMAR*, 2008.

[ZH05] C. Zhang and Z. Hu. A general sufficient condition of 4 positive solutions of the P3P problem. *Journal of Computer Science and Technology*, 20(6):836–842, 2005.

[ZH06] C. Zhang and Z. Hu. Why is the Danger Cylinder Dangerous in the P3P Problem? *Acta Automatica Sinica*, 32(4):504–511, 2006.

[ZSS20] Z. Zhang, T. Sattler, and D. Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis. *arXiv preprint arXiv:2005.05179*, 2020.

Eigene Veröffentlichungen

Feature Evaluation and Management for Camera Pose Tracking on 3D Models. Martin Schumann, Jan Hoppenheit and Stefan Müller. 9th International Conference on Computer Vision Theory and Applications (VISAPP). Lissabon, Portugal, 5.-8. Januar 2014

Initialization of Model-Based Camera Tracking with Analysis-by-Synthesis. Martin Schumann, Sebastian Kowalczyk, Stefan Müller. 8th International Symposium on Visual Computing (ISVC). Rethymnon, Kreta, Griechenland, 16.-18. Juli 2012

A Matching Shader Technique for Model-Based Tracking. Martin Schumann, Jan Hoppenheit, Stefan Müller. 20th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG). Pilsen, Tschechische Republik, 25.-28. Juni 2012

Parameter and Configuration Analysis for Non-Linear Pose Estimation with Points and Lines. Bernhard Reinert, Martin Schumann, Stefan Müller. 7th International Conference on Computer Vision Theory and Applications (VISAPP). Rom, Italien, 24.-26. Februar 2012

Combined Non-Linear Pose Estimation from Points and Lines. Bernhard Reinert, Martin Schumann, Stefan Müller. Arbeitsberichte aus dem Fachbereich Informatik, 09/2011, Universität Koblenz-Landau, ISSN (Online) 1864-0850

Analysis by Synthesis Techniques for Markerless Tracking. Martin Schumann, Sabine Achilles, Stefan Müller. In: Virtuelle und Erweiterte Realität, 6. Workshop der GI Fachgruppe VR/AR. Braunschweig, Deutschland, 18.-19. November 2009

Lebenslauf

seit 04/2017	**CA Digital GmbH, Hilden** Senior Softwareentwickler
10/2014 - 03/2017	**OrthoSetup GmbH, Neumarkt i.d. Oberpfalz** Projektorganisation, Softwareentwickler
10/2013 - 09/2014	**SOVAmed GmbH, Koblenz** Softwareentwickler
10/2009 - 09/2012	**Wissenschaftlicher Mitarbeiter** Arbeitsgruppe Computergraphik am Institut für Computervisualistik, Universität Koblenz-Landau Leitung und Organisation des DFG-Forschungsprojekts „Einsatz und Untersuchung von Analyse-durch-Synthese Techniken im markerlosen Tracking“
10/2008 - 06/2009	**Wissenschaftliche Hilfskraft mit Abschluss** Arbeitsgruppe Computergraphik am Institut für Computervisualistik, Universität Koblenz-Landau Akquise von Fördergeldern für Forschungsprojekte, Unterstützung der Lehre
04/2003 - 09/2008	**Studium der Computervisualistik** Universität Koblenz-Landau, Campus Koblenz Hochschulabschluss: Diplom-Informatiker