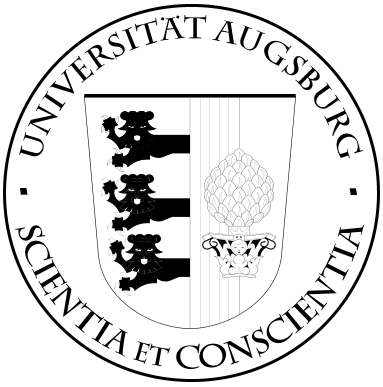
Universität Augsburg

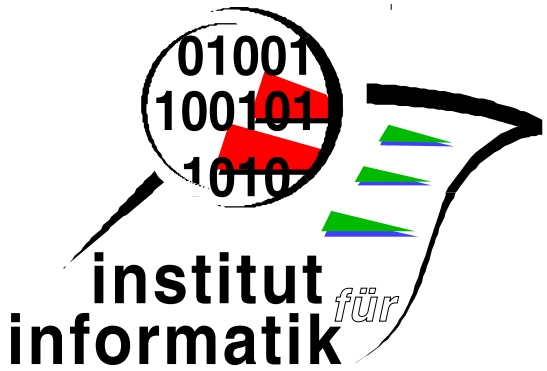


Optisches Kameratracking anhand natürlicher Merkmale

Diplomarbeit im Studiengang
Angewandte Informatik

Jonas Eschenburg

Report 2006-12 April 2006



Institut für Informatik
D-86135 Augsburg

Copyright © Jonas Eschenburg
Institut für Informatik
Universität Augsburg
D–86135 Augsburg, Germany
http://www.Informatik.Uni-Augsburg.DE
— all rights reserved —

Zusammenfassung

Systeme zur räumlichen Interpretation fotografischer Abbildungen liefern eine Grundlage für Anwendungen der *Erweiterten Realität*, welche die Kombination reeller und virtueller Bilder zum Ziel hat. Es sollen Systeme zum Kameratracking untersucht werden, die aus einer Bildsequenz die räumliche Bewegung der Kamera ableiten und damit eine perspektivisch korrekte Einbettung virtueller Objekte in das Bild erlauben. Im Gegensatz zu früheren Verfahren soll dies ohne die Anbringung spezieller Markierungen in der Szene funktionieren. Durch drei vorgestellte Systeme wird die Bandbreite existierender Ansätze repräsentiert. Eine Untersuchung der Geometrie des Kamerabilds folgt. Die erarbeiteten mathematischen Modelle finden anschließend in den für Tracking-Systeme relevanten Verfahren Anwendung. Abschließend soll der STAGE-Algorithmus vorgestellt werden, der den Kern einer Software bildet, welche zu dieser Diplomarbeit entwickelt wurde.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Dem Computer das Sehen beizubringen ist ein Ziel, das von Forschung und Industrie seit Jahren verfolgt wird. Viele interessante Entwicklungen haben seitdem zu ernsthaften und weniger ernsthaften Anwendungen geführt. Seien es die automatische Qualitätskontrolle bei der Produktion von Mikrochips, der automatisch navigierende Mars Rover der NASA oder der Tyrannosaurus Rex im Film *Jurassic Park* – hinter der Fassade halten die Techniken aus dem Computersehen Einzug. Dabei steht das bekannte Urzeitgeschöpf aus dem Spielberg-Klassiker stellvertretend für die wohl sichtbarste Anwendung dieses Forschungsgebiets, das im Schnittpunkt von Informatik, Physik und Biologie liegt. Techniken aus dem Computersehen ermöglichen dem Computer die dreidimensionale Interpretation fotografischer Abbildungen und damit das Verschmelzen von Realität und *Virtualität* – nicht nur im Kino.

Als Vorbild für diese Art der Bilderkennung dient hierbei unser eigenes visuelles Zentrum. Der Mensch ist in der Lage, in Sekundenbruchteilen das Signal seiner Augen zu verarbeiten und im Gehirn ein dreidimensionales Modell seiner Umwelt entstehen zu lassen. Aus optischen Merkmalen wie Ecken, Kanten, Farben und Schattierungen werden Informationen über Form, Lage und Bewegung der erkannten Objekte gesammelt. Es werden sowohl erlernte, als auch unbekannte Objekte erfasst.

Für den Computer stellt diese Aufgabe unter den zur Verfügung stehenden technischen Mitteln nach wie vor eine unüberwundene Hürde dar, wenn man die menschliche Leistungsfähigkeit zum Maßstab nimmt. Von einer vollautomatischen Verarbeitung des Bildes kann man nur in Teilproblemen sprechen. In vielen Fällen wird die Aufgabe erleichtert, entweder durch menschliche Hilfe oder durch spezielle Präparierung der Umwelt. So genannte Marker, das sind für den Computer leicht erkenn- und interpretierbare Objekte, werden an verschiedenen Punkten platziert. Ein großer Teil kommerziell verfügbarer Anwendungen basiert auf dem Einsatz dieser Marker.

Wie uns unser eigenes Sehvermögen beweist, ist ein räumliches Verstehen des Bildes auch ohne solche Marker möglich. Allein durch die Verwendung von natürlichen Merkmalen können Objekte im Bild wiedererkannt und ihre räumliche Lage bestimmt werden.

1.1 Zielsetzung

Diese Diplomarbeit möchte den Leser mit einer Reihe von Ansätzen vertraut machen, die bei der Implementierung von Bilderkennungs-Systemen verwendet wurden und werden. Es soll insbesondere auf die geometrische Interpretation fotografischer Abbildungen eingegangen werden. Diese stellt das wichtigste Instrument für ein erfolgreiches Bildverständnis dar. Weiterhin sollen die Algorithmen der Bildverarbeitung vorgestellt werden, welche den Einsatz der mathematischen Modelle erst ermöglichen.

Es war Teil der Aufgabenstellung dieser Arbeit, eine Software zu entwickeln, die auf den zu erlernenden Methoden aufsetzt. Die Entwicklung dieser Software dauerte einige Monate und hat dem Autor viele praktische Erkenntnisse geliefert. Ziel der Arbeit ist deshalb auch, die gesammelten Erfahrungen weiterzugeben, wobei jedoch nicht die quantitative Analyse im Vordergrund steht.

1.2 Überblick

In Kapitel 2 soll zunächst eine Klärung des Begriffs „Tracking“ und seiner engen Verwandten erfolgen, bevor auf die zu erwartende Funktionsweise aus einer *High-Level*-Sicht eingegangen wird.

Kapitel 3 stellt Systeme vor, welche an den Universitäten Oxford und Manchester sowie an der Technischen Hochschule von Lausanne entstanden sind. Letzteres lieferte den roten Faden bei der Erstellung der Software für diese Arbeit.

In Kapitel 4 soll die projektive Geometrie des Kamerabilds auf relativ ausführliche Weise behandelt werden. Es stehen insbesondere die mathematischen Modelle im Vordergrund, welche bei der Beschreibung von Bildinhalten dienlich sind. Ebenso wird auf die Ermittlung der Modellparameter eingegangen.

Das Kapitel 5 liefert die notwendigen Instrumente, um die mathematischen Modelle des vorherigen Kapitels auf Bilder anwenden zu können. Es werden die Themen Merkmalsextraktion und -matching sowie robuste Parameterschätzung behandelt.

Unter Berufung auf die vorgestellten Methoden wird in Kapitel 6 der Algorithmus STAGE unter die Lupe genommen, welcher das Kernstück der vom Autor entwickelten Software darstellt.

Kapitel 2

Beschreibung eines Tracking-Systems

Ein *Tracking-System* ist ein Algorithmus, welcher einen Strom aufeinander folgender Kamerabilder erhält und daraus die Perspektivenparameter jedes Bildes ermittelt. Dies geschieht, indem die abgebildete *Szene* durch ein räumliches Koordinatensystem beschrieben wird, welches jedem Punkt eine eindeutige Koordinate zuordnet. Die Perspektive des Bildes kann dann als Relation zwischen Welt- und Bildkoordinaten ausgedrückt werden.

Die Aufgabe, ein Weltkoordinatensystem möglichst gut auf ein Bildkoordinatensystem abzubilden, wird *Registrierung* genannt. Zu einer gelungenen Registrierung gehören korrekt ermittelte Daten über die Lage der Kamera im Weltkoordinatensystem sowie die Kalibrationsparameter, welche die optischen Vorgänge in der Kamera genau beschreiben.

Beim *Tracking* wird die Aufgabe dahingehend erweitert, die Registrierung über einen kontinuierlichen Strom von Bildern aufrechtzuerhalten. Dabei wird insbesondere die Ähnlichkeit aufeinander folgender Bilder ausgenutzt.

Aus dem Vergleich zweier Bilder derselben Szene lassen sich wesentliche Informationen über die Lage der Kamera in beiden Bildern ziehen. Dazu ist jedoch erforderlich, korrespondierende Punkte zu finden, was ein schwieriges Problem darstellt. Dieser Verfahrensschritt wird *Matching* genannt. Dabei wird nach leicht zu erkennenden *Merkmalen* in beiden Bildern gesucht und diese miteinander gepaart.

Matching und Registrierung sind eng miteinander Verknüpft. Ein gutes Matching ist nur über eine Registrierung zu erreichen und eine gute Registrierung benötigt ein Matching. In diesem Problem liegt die Schwierigkeit der Aufgabe. Nur durch die gleichzeitige Betrachtung beider Aufgaben ist das Problem zu lösen, weshalb die Begriffe zu einem gewissen Grad austauschbar sind.

2.1 Unterscheidungen

Man unterscheidet grundsätzlich zwischen zwei Kategorien von Tracking-Systemen: so genannten *Online*- und *Offline*-Systemen.

Online-Systeme sind auf Echtzeitbedingungen ausgelegt. Sie müssen ein Bild in möglichst kurzer Zeit verarbeiten. Dazu stehen ihnen nur das aktuelle und die vorhergegangenen Bilder zur Verfügung. Üblicherweise verwenden Online-Systeme ein gespeichertes Vorwissen über die Szenengeometrie, obwohl es auch Gegenbeispiele gibt. Weiterhin wird zumeist vorausgesetzt, dass die Kalibrationssparameter der Kamera bekannt sind. Die Software *StageDesigner*, welche begleitend zu dieser Arbeit geschrieben wurde, beinhaltet einen Online-Tracker. Die im Folgenden vorgestellten Methoden orientieren sich deshalb an den Bedürfnissen dieser Kategorie von Tracking-Systemen.

Offline-Systeme sind auf größtmögliche Genauigkeit ausgelegt. Die Rechenzeit ist zweitrangig. Sie beziehen ihre Bilder nicht direkt über eine Kamera, sondern aus einer gespeicherten

Filmdatei. So brauchen sie nicht Bild für Bild vorzugehen, sondern können global arbeiten, also auf der gesamten Bildsequenz. Großer Beliebtheit erfreuen sich kommerziell verfügbare Offline-Tracker in der Filmindustrie. Dort wird für die Verknüpfung von computergeneriertem mit gefilmtem Bildmaterial eine exakte (virtuelle) Replikation der (reellen) Kamerabewegung benötigt. Das Kameratracking wird deshalb in der Filmindustrie auch *Matchmoving* genannt. Da Offline-Systeme ein Vielfaches der Rechenzeit von Online-Systemen benötigen dürfen, können sie auf deutlich aufwändigere Verfahren zurückgreifen. Dies ermöglicht die Arbeit ohne gespeichertes Vorwissen über die Szenengeometrie, weil sich aus dem Vergleich von drei oder mehr Bildern die vollständige dreidimensionale Geometrie des Bildes rekonstruieren lässt. Ebenso kann auf eine Kalibration der Kamera verzichtet werden, da die Kalibrationsparameter direkt aus der Bildsequenz ermittelt werden.

2.2 Lernphase

Benötigt ein Tracking-System Vorkenntnisse über die Szenengeometrie, so nennt man das Sammeln dieser Informationen *Lernphase*, in Anlehnung an die Verfahren aus der Künstlichen Intelligenz. Jedoch muss dieser Arbeitsschritt keineswegs automatisch erfolgen. Üblich sind gespeicherte Informationen, die die Assoziation von Bildmerkmalen mit räumlichen Elementen ermöglichen sowie Polygonmodelle der Szene. Ersteres wären beispielsweise im Vorfeld gemachte Aufnahmen der Szene, so genannte *Keyframes*, für die eine Registrierung bereits stattgefunden hat. Hierbei wird zumeist ein manueller Eingriff erfordert.

2.3 Merkmalsextraktion

In der Form, in der sie im Computer gespeichert werden, macht es wenig Sinn, nach Gemeinsamkeiten zwischen Bildern zu suchen. Die gewaltige Masse an Daten macht es notwendig, Wichtiges von Unwichtigem zu unterscheiden. Deshalb ist der erste Schritt, den die meisten Tracking-Systeme bei einem Bild unternehmen, dieses auf eine Menge möglichst markanter Merkmale zu reduzieren. In den meisten Fällen handelt es sich dabei um Punkte (genau genommen kleine Bildausschnitte), die bestimmte Eigenschaften besitzen, welche sie leicht identifizierbar machen, beispielsweise starke Kontraste in der Bildhelligkeit. Es sind aber auch andere Arten von Merkmalen möglich.

2.4 Matching und Registrierung

Ziel des Tracking-Systems ist es, das aktuelle Bild mit einem dreidimensionalen Modell zu registrieren. Das heißt, eine mathematische Transformation zu finden, die das Modell annähernd deckungsgleich auf das tatsächliche Kamerabild projiziert. Der Grad der Übereinstimmung kann anhand der zuvor extrahierten Merkmale gemessen werden. Dazu werden Methoden der Bildverarbeitung eingesetzt.

Um überhaupt eine sinnvolle Hypothese über die Bildperspektive aufstellen zu können, ist die eindeutige Identifikation von Bildmerkmalen erforderlich. Eine Hauptschwierigkeit dabei ist, dass optisch vergleichende Methoden leicht auf Mehrdeutigkeiten im Bild hereinfallen. Beispielsweise kann das Fenster eines Gebäudes zwar als leicht zu lokalisierendes (weil kontrastreiches) Merkmal angesehen werden. Ist es jedoch ein Fenster unter vielen gleichartigen,

womöglich einer ganzen Fensterfront, so wird es beim Matching zweier Bilder oft falsch zugeordnet. Falsch zugeordnete Merkmale können den Versuch, die Parameter für die Registrierung zu bestimmen, zunichte machen, da sie große Fehler in die Berechnungen einbringen. Es gilt also, „Ausreißer“ in den Daten zu erkennen und zu unterdrücken.

2.5 Tracking

Ist eine gute Registrierung erst einmal für ein Bild gelungen, ist es verhältnismäßig leicht, ebenso eine Registrierung für nachfolgende Bilder zu finden. Dabei macht man sich die relative Ähnlichkeit benachbarter Bilder einer Sequenz zunutze. Diese ermöglicht eine exakte Ermittlung der Bewegung, also der Änderung der Kamerapose im Bezug zum vorherigen Bild. Diese Art des Trackings nennt sich *relatives Tracking*, da ab einem bestimmten Zeitpunkt keine absoluten (also in der Lernphase ermittelten) Modellinformationen mehr in die Berechnung einfließen. Relatives Tracking hat die wünschenswerte Eigenschaft, Kamerabewegungen sehr exakt nachzuvollziehen. Problematisch ist jedoch, dass sich, wie bei jedem relativen Verfahren, kleine Messfehler akkumulieren. Dieser Effekt wird *Drift* genannt.

Dem im Abschnitt zuvor beschriebenen absoluten Tracking haftet dieser Makel nicht an. Dadurch, dass die Registrierung zu jedem Bild unabhängig ermittelt wird, ergibt sich jedoch ein Zittereffekt, der so genannte *Jitter*. Gute Tracking-Systeme kombinieren daher beide Verfahren und erhalten exakte Lösungen mit wenig Jitter.

Ob sich die Kamera in Relation zur Szene bewegt oder umgekehrt, also die Kamera stillsteht und sich die Szene bewegt, ist prinzipiell egal. In dieser Arbeit wird immer vom ersten Fall ausgegangen. Damit eine Bewegung der Kamera überhaupt die Vorgänge in einer Bildsequenz adäquat beschreiben kann, darf sich die dreidimensionale Geometrie der Szene während der Aufnahmezeit nicht ändern. Die meisten Trackingverfahren treffen deshalb eine *Starrheitsannahme*. In der Praxis kann jedoch ein kleiner Anteil an Bewegungen im Bild, beispielsweise Passanten vor einem Gebäude, heraus gefiltert werden. Es gibt des Weiteren spezielle Trackingverfahren für die Verfolgung dynamischer Szenen mit mehreren bewegten Objekten.[Bar02]

Kapitel 3

Bestehende Systeme

Es sollen nun drei Tracking-Systeme untersucht werden, die jeweils exemplarisch für eine Kategorie stehen. Das erste System wurde aufgrund seiner Einfachheit ausgewählt und weil es mit relativer Posebestimmung arbeitet, das heißt nach einer Initialisierung keine absoluten Informationen über die Szenengeometrie mehr verwendet. Es handelt sich um den Tracker der Universität Oxford, vorgestellt in [SFZ00].

Das zweite System ist an der Eidgenössischen Technischen Hochschule in Lausanne (EPFL) entstanden und wird in [VLF04b, und weiteren] vorgestellt. Es verwendet ein hybrides Verfahren, bei dem sowohl relative als auch absolute Informationen in die Posebestimmung eingehen.

Als Letztes soll ein System vorgestellt werden, welches – im Gegensatz zu den beiden vorherigen – als Offline-System konzipiert wurde und deutlich aufwändigere Verfahren einsetzen kann.

3.1 Das System der Universität Oxford

Bei dem Tracking-System der Universität Oxford [SFZ00] handelt es sich um einen verhältnismäßig einfach aufgebauten Online-Tracker, der so konzipiert wurde, dass er ohne Vorkenntnisse über die Szenengeometrie und mit minimalem Aufwand für den Benutzer einsetzbar ist. Der einzige manuelle Eingriff findet zur Initialisierung statt. Das anschließende Tracking der Kameraposition erfolgt vollautomatisch.

Auch auf eine genaue Vermessung der Kamerakalibrationsparameter kann verzichtet werden. Eine Abschätzung lässt sich aus den anfangs getätigten Eingaben ermitteln.

Bei der Posebestimmung macht sich das System den Umstand zunutze, dass sich in vielen Situationen planare Strukturen im Bild finden lassen. Dies können der Fußboden, die Wände eines Zimmers oder die Fassade eines Gebäudes sein.

Initialisierung Als ersten Schritt der Initialisierung markiert der Benutzer einen möglichst großen Bildbereich, der jedoch vollständig innerhalb der gewählten Ebene liegen sollte. Dies gibt dem nachfolgenden Matchingschritt einen Hinweis darauf, welche Ebene im Bild verfolgt werden soll und kann weggelassen werden, wenn die zu verfolgende Ebene auch die größte Ebene im Bild ist.

Eine Posebestimmung mittels einer Ebene im Bild ist dann möglich, wenn die Kalibrationsparameter der Kamera sowie die Lage der Ebene im Raum bekannt sind. Da das Weltkoordinatensystem jedoch frei gewählt werden kann, wird die Ebene als XY-Ebene festgelegt. Der Benutzer markiert nun vier Punkte im Bild mit der Maus. Dabei muss es sich um Punkte handeln, die in der (tatsächlichen, dreidimensionalen) Szene ein Rechteck auf der gewählten Ebene bilden. Es kann sich beispielsweise um die vier Ecken eines Fensters oder eines großen Pflastersteins handeln. Der erste auf diese Weise markierte Punkt wird zum Ursprung des

Abbildung 3.1: Die Initialisierung des Tracking-Systems der Universität Oxford ist besonders einfach. Zuerst wird ein Bildbereich markiert, der auf der zu verfolgenden Ebene liegen muss(links). Darauf setzt das Tracking ein (Mitte). Der Benutzer muss nun vier Punkte in rechteckiger Anordnung markieren(rechts). Damit ist die Initialisierung abgeschlossen. Entnommen aus [SFZ00] mit freundlicher Genehmigung von Gilles Simon.

Weltkoordinatensystems, die darauf folgenden Punkte definieren zusammen mit dem ersten die X- und Y-Koordinatenachsen.

Vier Punkte genügen, um die zweidimensionale projektive Abbildung von Punkten aus der XY-Ebene des Weltkoordinatensystems in die Bildebene zu bestimmen. Eine solche Abbildung nennt sich *Homographie* und wird als Matrix geschrieben. Diese spezielle Homographie $\boldsymbol{H_w^0}$ wird „Welthomographie für Bild 0“ genannt.Aus dieser Homographie kann, ähnlich zum Verfahren in *Pose aus Homographie* in Abschnitt 4.3.2, S.34, die Pose der Kamera bestimmt werden.

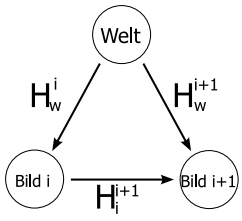
Voraussetzung ist dabei jedoch ein Vorwissen über die Kalibrationsparameter der Kamera. Diese lassen sich jedoch in vielen Fällen in ausreichender Genauigkeit schätzen. Es fehlt nur noch die Brennweite, die sich aber aus den markierten Punkten ergibt.

Es werden nun für dieses und alle darauf folgenden Bilder mittels des *Harris Corner Detectors* (vgl. 5.1.3, S.38) Merkmalspunkte extrahiert.

Matching aufeinander folgender Bilder Für jedes $\boldsymbol{i+1}$-te Bild, das nach dieser Initialisierung von der Kamera eintrifft, wird nun versucht, Merkmale zu finden, die mit jenen des vorherigen Bildes korrespondieren. Dabei wird die Suche auf ein Suchfenster um den Merkmalspunkt eingeschränkt. Der optische Vergleich zweier Merkmale geschieht, indem im Suchfenster nach Merkmalen mit hoher Korrelation Ausschau gehalten wird. Mittels des „robusten“ *RANSAC*-Algorithmus (vgl. 5.3.1, S. 53) wird gleichzeitig nach einem guten Matching zwischen den Merkmalen und einer Homographie $\boldsymbol{H_i^{i+1}}$ gesucht, welche die projektive Abbildung zwischen den beiden Merkmalsmengen beschreibt.

Durch Verkettung dieser Homographie mit der Welthomographie des Vorgängerbildes kann auch für dieses Bild eine Welthomographie berechnet werden: $\boldsymbol{H_w^{i+1} = H_i^{i+1} H_w^i}$.

Das folgende Diagramm veranschaulicht die Rechnung:



Posebestimmung Auf diese Weise steht zu jedem Bild $\boldsymbol{i}$ eine Welthomographie $\boldsymbol{H_w^i}$ zur Verfügung, die die Punkte auf der XY-Ebene des Weltkoordinatensystems auf Punkte in der Bildebene abbildet. Aus $\boldsymbol{H_w^i}$ lässt sich die Projektionsmatrix $\boldsymbol{P_i}$ bestimmen, welche die vollständige Information zu Pose und Projektion der Kamera enthält.

Erweiterungen Damit ist die „Basisvariante“ des Trackers bereits beschrieben. Über längere Strecken wird die Welthomographie allerdings immer ungenauer, da sich kleine Fehler akkumulieren. Es werden deshalb weitere Verfahren verwendet, um die Genauigkeit über längere Sequenzen zu steigern. Zum einen wird versucht, eine Homographie $\boldsymbol{H_0^i}$ zwischen dem Bild 0 und dem aktuellen Bild zu berechnen. Dabei kann die aktuelle, relativ ermittelte Welthomographie verwendet werden: $\boldsymbol{\hat{H}_0^i = H_w^i (H_w^0)^{-1}}$ liefert einen Schätzwert. Dieser hilft bei der Suche nach der exakteren Homographie $\boldsymbol{H_0^i}$, indem er

- Schätzwerte für die Merkmalspositionen vorgibt und
- die Merkmale pixelweise so transformiert, dass sie annähernd deckungsgleich liegen und somit die Korrelation erhöhen.

Um einen weiteren Genauigkeitszuwachs zu erreichen, werden neben den Punkten auf der Ebene auch noch einige Punkte außerhalb der Ebene verfolgt. Deren Position im Raum ist jedoch unbekannt und muss aus dem Vergleich verschiedener Ansichten geschätzt werden.

Es kann sein, dass die getrackte Ebene das Bild verlässt. In diesem Fall ist es notwendig, eine andere Ebene zu verfolgen. Deshalb wird ständig nach einer alternativen Ebene gesucht. Diese liegt allerdings in unbekannter räumlicher Lage. Befinden sich jedoch mindestens drei der oben genannten Punkte auf der Ebene, so kann ihre Lage ermittelt werden. Es findet dann eine „Ebenenübergabe“ statt, in der in ein neues räumliches Koordinatensystem gewechselt wird.

3.2 Das System der EPFL

Das an der Eidgenössischen Technischen Hochschule in Lausanne entstandene und in mehreren Veröffentlichungen [VLF04b, VLF04a, LPF04, LVTF03] vorgestellte System ist zurzeit einer der am weitesten fortgeschrittenen Ansätze auf dem Gebiet der Augmented Reality. Es vereint erfolgreich relative und absolute Informationen. Dafür benötigt es allerdings in der Lernphase viel manuelle Unterstützung.

Konkret wird gefordert, dass die zu trackende Szene als dreidimensionales Polygonmodell vorliegt und eine Reihe von Beispielansichten (Keyframes) vorhanden sind. Zu diesen Keyframes muss eine gültige Registrierung vorliegen. Weiterhin müssen die Kalibrationsparameter der Kamera bekannt sein.

Damit das Tracking nicht auf die durch die Keyframes grob vorgegebenen Kamerapositionen reduziert ist, sieht das System die Erstellung von Keyframes im laufenden Betrieb vor. Diese so genannten *Online-Keyframes* ermöglichen ein unterbrechungsfreies Tracking, solange sich ein Objekt im Bild befindet, welches als Polygonmodell vorliegt.

Das Polygonmodell der Szene (oder Objekten darin) ermöglicht zusätzlich, ein Keyframe aus einer anderen Perspektive zu berechnen. Über diese Neuberechnung kann die Korrelation der Merkmale des aktuellen Bildes mit denen des Keyframes maximiert werden, indem eine Ansicht nahe der aktuellen Kameraposition erzeugt wird. Zusätzlich erlaubt das Polygonmodell auf einfache Weise, Punkte in die Registrierung aufzunehmen, deren 3D-Positionen unbekannt sind.

Die Lernphase Der Tracker der EPFL ist ein System, das Vorwissen über die Szene enthält. Dieses besteht zum einen aus Keyframes. In diesen sucht ein Harris Corner Detector nach Merkmalspunkten. Mittels des kommerziellen Programms ImageModeler™ von RealViz™ wird die Kamera für dieses Bild registriert sowie die dreidimensionale Lage der Merkmalspunkte bestimmt.[1]

Als weitere Information wird ein Polygonmodell der Szene beziehungsweise einiger darin vorkommender Objekte benötigt. Dieses kann mit ImageModeler oder einer beliebigen anderen CAD-Anwendung erstellt werden. Die Verwendung eines Polygonmodells erhöht die Anforderungen an den Anwender beträchtlich. Vacchetti *et al.* rechtfertigen diesen Aufwand damit, dass für viele Anwendungen aus dem Bereich der Augmented Reality in jedem Fall ein Modell benötigt wird.[VLF04b] Modell und Merkmalspunkte müssen dasselbe Weltkoordinatensystem verwenden.

Das Trackingverfahren Wir gehen für den Moment davon aus, dass eine Registrierung des Vorgängerbilds $\boldsymbol{i}$ bereits stattgefunden hat und kommen später auf das Problem der Initialisierung zurück.

Das System der EPFL nutzt bereits während der Registrierung der Kamerapose für das aktuelle Bild eine Kombination aus absoluten Informationen aus der Lernphase und relativen Informationen aus dem Vergleich mit dem Vorgängerbild. Wie die meisten anderen Algorithmen verwendet auch dieses System korrespondierende Punktmerkmale, die über einen Korrelationsvergleich bestimmt werden. Eine Besonderheit des Verfahrens besteht darin, dass die so gefundenen Merkmalspaare nicht zu 100% fehlerfrei sein müssen. Durch die Verwendung eines robusten M-Schätzers (vgl. 5.3.2, S.54) kann eine gute Bestimmung der Kamerapose auch unter fehlerhaften Korrespondenzen stattfinden.

Es treten hierbei zwei Typen von Korrespondenzen auf: 2D-2D und 2D-3D. Erstere stammen aus gefundenen Korrespondenzen zwischen aktuellem Bild und Vorgängerbild. Letztere werden durch Korrespondenzenfindung mit dem aktuellen Keyframe ermittelt und sollen eine Fehlerakkumulation verhindern, die bei rein relativem Tracking auftreten würde.

Sei $\boldsymbol{q}$ der zu optimierende Parametervektor für die Pose der Kamera im aktuellen Bild. Es ist wünschenswert, eine Kamerapose zu finden, die den Abstand zwischen projizierten 3D- und gemessenen 2D-Punkten minimiert. Dieser lässt sich durch die folgende Summe über alle 3D-2D-Korrespondenzen $\boldsymbol{M} \leftrightarrow \boldsymbol{m}$ des Kamerabilds beschreiben:

$$\boldsymbol{r_q} = \sum_{\boldsymbol{j}} \boldsymbol{\rho} \left(\|\boldsymbol{\Phi}(\boldsymbol{q}, \boldsymbol{M_j}) - \boldsymbol{m_j}\|^{\boldsymbol{2}} \right)$$

[1] `http://www.realviz.com`

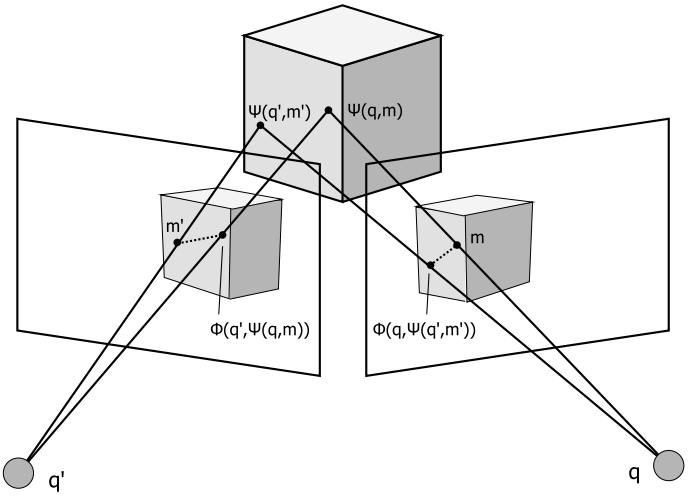


Abbildung 3.2: Transferfehler zwischen Kameras mit Poseparametern $\boldsymbol{q}$ und $\boldsymbol{q'}$ sowie Punkten $\boldsymbol{m}$ und $\boldsymbol{m'}$. Die gestrichelte Linie stellt die zu minimierende Distanz da.

Dabei ist $\boldsymbol{\rho}$ der M-Schätzer von Tukey (s. [Zha95]) und $\boldsymbol{\Phi(q)}$ die Projektion von Weltkoordinaten in die Bildebene unter Kameraparametern $\boldsymbol{q}$.

Die Verwendung von Polygonmodellen macht den Einsatz von 2D-2D-Korrespondenzen zur Posebestimmung möglich, da sie eine Rückprojektion eines 2D-Punktes auf die Modelloberfläche, also einen Punkt im Weltkoordinatensystem ermöglichen. Diese Rückprojektion sei durch $\boldsymbol{\Psi(q, m)}$ bezeichnet und projiziert einen Punkt $\boldsymbol{m}$ auf der Bildebene der Kamera mit Pose $\boldsymbol{q}$ auf einen Punkt $\boldsymbol{M}$ im Weltkoordinatensysten. Somit lässt sich der folgende *Transferfehler* als Summe über alle 2D-2D-Korrespondenzen $\boldsymbol{m' \leftrightarrow m}$ zwischen vorherigem und aktuellem Kamerabild definieren:

$$\boldsymbol{s = \sum_j \rho \left(\|\Phi(q, \Psi(q', m')) - m\|^2 + \|m' - \Phi(q', \Psi(q, m))\|^2 \right)}$$

Abbildung 3.2 stellt die Situation anschaulicher dar. Die Terme $\boldsymbol{r_q}$ und $\boldsymbol{s}$ werden nun zusammen minimiert. Da die berechnete Pose $\boldsymbol{q'}$ des Vorgängerbildes fehlerhaft gewesen sein und weil sich dieser Fehler auch auf die neue Pose $\boldsymbol{q}$ auswirken könnte, wird $\boldsymbol{q'}$ noch einmal mit in die Minimierung einbezogen:

$$\boldsymbol{\min_{q,q'} \quad r_q + r_{q'} + s}$$

Als Initialwert für $\boldsymbol{q}$ und $\boldsymbol{q'}$ wird jeweils das bereits berechnete $\boldsymbol{q'}$ verwendet. Für weitere Ausführungen sei auf [LVTF03] verwiesen.

Eine Erweiterung auf die Verfolgung von Linienmerkmalen wird in [VLF04a] vorgestellt.

Matching mit dem Keyframe Zum Vergleich der Merkmale des aktuellen Bildes mit denen des Keyframes wird die normalisierte Kreuzkorrelation (NCC, s.S. 46) verwendet. Da es sich bei dieser jedoch um einen einfachen Pixel-für-Pixel-Vergleich in einem kleinen Fenster um den Punkt handelt, reagiert der NCC-Operator äußerst empfindlich auf Änderungen im Betrachtungswinkel. Dies würde sich sehr negativ auf das Matching mit den Keyframes auswirken.

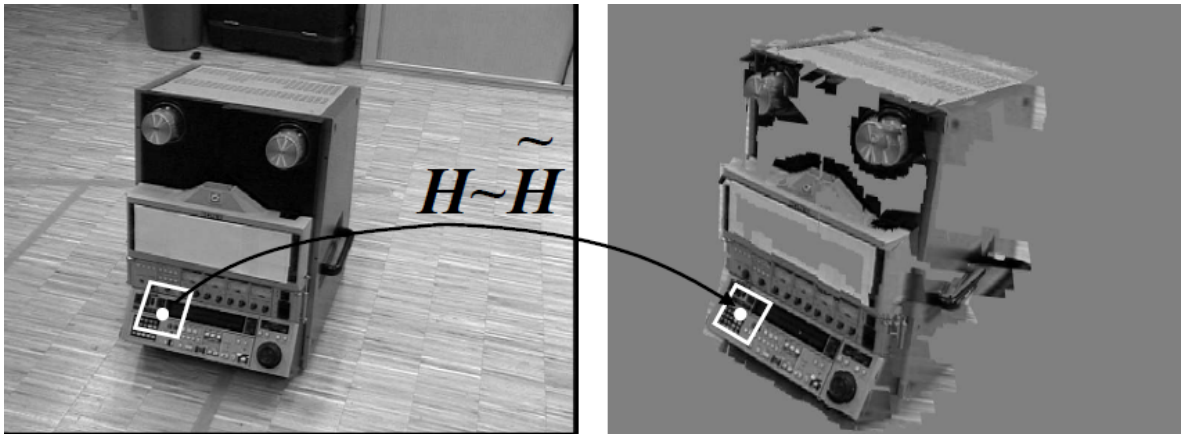


Abbildung 3.3: Neu erzeugte Ansicht eines Keyframes. Für jedes Merkmal wird durch eine Homographie ein kleiner, quadratischer Bildausschnitt transformiert. Auf diese Weise wird die Korrelation maximiert. Entnommen aus [VLF04b] mit freundlicher Genehmigung von Pascal Fua und Vincent Lepetit.

Da jedoch ein Polygonmodell der Szene zur Verfügung steht, ist es möglich, eine Version des Keyframes aus einer anderen Perspektive zu erzeugen. Dies geschieht vor dem eigentlichen Matching-Vorgang. Es steht also noch keine aktuelle Kamerapose zur Verfügung. Als Ersatz wird die Pose $\boldsymbol{q'}$ des Vorgängerbildes verwendet, die in den meisten Fällen der aktuellen (jedoch noch unbekannten) Pose $\boldsymbol{q}$ ähnlich ist. Die eigentliche Neuberechnung der Keyframe-Ansicht geschieht, indem zu jedem Merkmalspunkt des Keyframes eine Homographie berechnet wird, die ein kleines rechteckiges Fenster um den Punkt in das neue Bild überträgt. Die Homographie wird dabei durch die jeweilige Facette des Polygonmodells bestimmt. Abbildung 3.3 zeigt eine solche Neuberechnung.

Initialisierung Frühere Versionen des Systems bedurften zur Initialisierung der manuellen Auswahl eines Keyframes. Dann wurde die Kamera in eine der Keyframeperspektive ähnliche Stellung gebracht. War die Position ähnlich genug, stieg die Anzahl verfolgter Merkmale und das Tracking begann.[VLF03]

Im Unterschied zum oben beschriebenen Matching mit dem Keyframe steht zum Zeitpunkt der Initialisierung noch keine Kameraposition zur Verfügung. Es kann also keine Ansicht des Keyframes generiert werden, die dem aktuellen Kamerabild ähnelt. Weiterhin ist auch zu diesem Zeitpunkt ohne manuelle Hilfe oder Zufallsauswahl noch nicht klar, *welches* Keyframe betrachtet werden soll.

In neueren Ausbaustufen des Trackers wurde eine *Eigenbild*-basierte Merkmalserkennung integriert [LPF04, LVTF03]. Hierbei handelt es sich um eine Methode, das Bild anhand von Filtern untersucht. Diese Filter sind darauf zugeschnitten, ein spezifisches, vorher „eintrainiertes“ Merkmal unter geometrischer Transformation zu erkennen. Es handelt sich um ein Verfahren aus der statistischen Datenanalyse, dem eine Hauptkomponentenanalyse zugrunde liegt (s. Abschnitt 5.2.3, S.49).

Mit den so gewonnenen Korrespondenzen wird ein robuster RANSAC-Algorithmus initialisiert, der versucht, gleichzeitig eine gute Registrierung *und* ein gutes Matching zu ermitteln. Wird eine Lösung gefunden, ist die Initialisierung abgeschlossen.

3.3 Das System der Universität von Manchester

Online-Verfahren bilden den Schwerpunkt dieser Arbeit. Dennoch soll ein Blick über den Tellerrand gewagt werden. Das Tracking-System ICARUS wurde an der Universität von Manchester entwickelt und durch durch Gibson *et al.* in [GCH+02] vorgestellt. Mittlerweile ist ICARUS zur kommerziellen Software PFTRACK™ weiterentwickelt worden.[2] Damit steht die ursprüngliche Software nicht mehr zum kostenlosen Download zur Verfügung.

ICARUS wurde mit dem Ziel entwickelt, eine stabile Rekonstruktion der Szenenstruktur und der Kamerabewegung einer Bildsequenz unter minimalen Voraussetzungen zu ermöglichen. Weder wird Vorwissen jeglicher Art über die Szene verlangt, noch muss eine Kalibration der Kamera erfolgt sein. Jedoch kann bestehendes Vorwissen über die Kalibrationsparameter zur Verbesserung der geometrischen Berechnungen einfließen.

Auch dieses System setzt Keyframes ein, allerdings handelt es sich nicht um im Voraus erfolgte Aufnahmen, sondern um tatsächliche Bilder innerhalb der Bildsequenz. Diese werden so ausgesucht, dass je zwei Keyframes die Szene aus möglichst verschiedenen Winkeln abbilden. So wird die anschließende Berechnung der Szenenstruktur und Kamerabewegung in den dazwischen liegenden Bildern stabilisiert.

Aus der so entstandenen Aneinanderreihung rekonstruierter Bildsequenzen wird durch ein Verschmelzungsverfahren zwischen je zwei benachbarten Abschnitten eine zusammengeführte Bildsequenz rekonstruiert. In mehreren Durchgängen werden immer mehr überlappende Teilstücke miteinander verschmolzen. Abschließend wird die gesamte Sequenz aus den verschmolzenen Teilstücken zusammengeführt.

Verfolgung von Merkmalspunken Der erste Schritt des Systems ist die Verfolgung von Merkmalspunkten. ICARUS legt viel Wert darauf, jedes Merkmal über eine möglichst lange Bildsequenz hinweg zu verfolgen. Dazu werden Merkmalspunkte im ersten Bild der Sequenz über einen Harris Corner Detector ermittelt. Im Unterschied zu den vorangegangenen Tracking-Systemen wird zur Verfolgung der Merkmale kein Matching mit der Merkmalsmenge des nachfolgenden Bildes durchgeführt. Vielmehr wird jedes einzelne Merkmal Bildpunkt für Bildpunkt in einem Ausschnitt des nachfolgenden Bildes gesucht. Dies geschieht durch ein erweitertes KLT-Verfahren[ST94].

Es wird darauf geachtet, dass in jedem Bild eine bestimmte Mindestzahl von Merkmalen vorhanden ist. Dies wird durch Hinzufügen weiterer Punkte erreicht. Als Besonderheit von ICARUS werden die hinzugefügten Merkmale *in beide Richtungen* verfolgt. Dies geschieht in einem anschließend durchgeführten Rückwärtsverfolgungsschritt.

Auswahl von Keyframes Nun wird nach Paaren von Bildern $\boldsymbol{i}$ und $\boldsymbol{j}$ gesucht, die sich besonders gut für die anschließende Rekonstruktion eignen. Es wird darauf geachtet, dass

- mindestens 50% ihrer jeweiligen Merkmale korrespondieren
- $\boldsymbol{i}$ und $\boldsymbol{j}$ eine signifikante Parallaxe aufweisen.

Der letzte Punkt bedeutet, dass die Bilder perspektivisch weit genug auseinander liegen, um eine stabile Rekonstruktion der verbundenen Bildgeometrie zu erlauben. Speziell soll vermieden

[2] `http://www.thepixelfarm.co.uk/`

werden, dass die Bildsequenz zwischen i und j vor allem eine Ebene abbildet, also nur zweidimensionale Struktur beinhaltet. Dies geschieht, indem geprüft wird, wie gut sich die Transformation der Merkmalskoordinaten zwischen i und j als Ebenen-zu-Ebenen-Transformation (Homographie) darstellen lässt. Ein hoher Fehler deutet auf eine große Parallaxe hin.

Projektive Rekonstruktion Die Bildparallaxe ist notwendig, um eine gute Rekonstruktion der Teilsequenz zwischen den Keyframes zu erlauben. Dies geschieht unabhängig voneinander für alle Paare benachbarter Keyframes. Bei diesem so genannten *Bündelblockausgleich*, einem Standardverfahren der Photogrammetrie, werden in einer gemeinsamen Optimierung die Koordinaten der Merkmalspunkte im Weltkoordinatensystem M_i (mit Projektion $m_i^{(j)}$ im Bild j) und die zu jedem Bild der Teilsequenz gehörigen Kameraparameter q_j bestimmt, sodass ein *Reprojektionsfehler* minimiert wird:

$$\underset{q_1,\ldots,q_k,M_1,\ldots,M_n}{min} \sum_{j=1}^{k} r_j$$

$$r_j = \sum_{i=1}^{n} \parallel \Psi(q_j, M_i) - m_i^{(j)} \parallel^2$$

Dabei bezeichnet Ψ wieder die Projektion vom Welt- ins Bildkoordinatensystem. Einen guten Überblick über das Verfahren des Bündelblockausgleichs gibt [TMHF00].

Verschmelzung In einem Verschmelzungsverfahren werden die so ermittelten Rekonstruktionen zusammengeführt. Es müssen die ermittelten Weltkoordinaten der Kameraparameter und Merkmalspunkte in ein gemeinsames Koordinatensystem übertragen werden. Eine Verschmelzung wird zwischen allen benachbarten Sequenzen durchgeführt. Dabei überlappen sich benachbarte Sequenzen in jeweils einem Bild, dem gemeinsamen Keyframe. Den Merkmalspunkten, die innerhalb dieser Überlappung auftreten, sind in beiden Sequenzen Weltkoordinaten zugeordnet. Dadurch ist es möglich, über einen robusten RANSAC-Algorithmus all jene Punkte herauszufiltern, die sich *nicht* durch eine Transformation des Koordinatensystems ineinander überführen lassen. Diese lassen auf schlecht lokalisierte Merkmale schließen und können somit heraus gefiltert beziehungsweise repariert werden.

In weiteren Verschmelzungsschritten werden die Überlappungen sogar noch größer. Angenommen, es seien i_1, i_2, i_3, i_4 benachbarte Keyframes. Dann werden im ersten Schritt folgende Verschmelzungen durchgeführt:

$$(i_1, i_2) + (i_2, i_3) \rightarrow (i_1, i_3)$$

$$(i_2, i_3) + (i_3, i_4) \rightarrow (i_2, i_4)$$

und im zweiten Schritt:

$$(i_1, i_3) + (i_2, i_4) \rightarrow (i_1, i_4)$$

Dabei wächst der überlappende Teil jedes Mal an.

In einem letzten Durchlauf findet eine Verschmelzung sämtlicher überlappender Teilsequenzen statt und der Algorithmus ist abgeschlossen.

3.4 Weitere Trackingverfahren

Es wurden drei Systeme vorgeführt, die einen Ausschnitt aus dem Spektrum der Augmented Reality Tracking-Systeme zeigen.

Nicht berücksichtigt wurden Verfahren, die zur Registrierung nicht oder nicht ausschließlich die Kamera verwenden. Es ist eine Vielzahl an Sensoren verfügbar, die physikalische Eigenschaften messen. Vom trägheitsbasierten Tracking mittels optischen Kreiselinstrumenten und Beschleunigungsmessern (Gravimetern) über ultraschallbasierte Ortungsverfahren bis hin zur globalen Positionsbestimmung mittels Satellitennavigation gibt es eine ganze Klasse weiterer Trackingverfahren, die jedoch mit dem Erkennen natürlicher Merkmale nichts mehr zu tun haben.

Die Tabelle auf S.20 fasst die behandelten Systeme zusammen.

	Oxford	**EPFL**	**Icarus**
Kategorie	Online	Online	Offline
Vorwissen über Szenengeometrie	Nein	Ja, Keyframes und Polygonmodell	Nein
Merkmalstyp	Punktmerkmale	Punktmerkmale (Linienmerkmale)	Punktmerkmale
Merkmalsextraktion	Harris Corner Detector	Harris Corner Detector	Harris Corner Detector
Merkmalsgröße	Fest	Fest	Fest
Merkmalsverfolgung	Merkmal-zu-Merkmal-Matching	Merkmal-zu-Merkmal-Matching	Explizite Suche nach Merkmal(KLT)
Ähnlichkeitsmaß →aufeinander folgende Bilder →entfernte Bilder	 NCC NCC unter projektiver Transformation	 NCC NCC unter projektiver Transformation, Eigenbildanalyse (Initialisierung)	 NCC unter affiner Transformation —
Posebestimmung	Pose aus Homographie	POSIT (Initialisierung), allgemeine Posebestimmung	Bündelblockausgleich
Robuste Parameterbestimmung	RANSAC	RANSAC (Initialisierung), M-Schätzer	M-Schätzer (Bündelblockausgleich), RANSAC (Verschmelzungsschritt)

Kapitel 4

Die Projektive Geometrie des Kamerabilds

Um eine erfolgreiche Bilderkennung durchzuführen, ist das Verständnis des durch eine Kamera aufgezeichneten Bildes und der möglichen geometrischen Transformationen von elementarer Wichtigkeit. Deshalb beschäftigt sich Abschnitt 4.1 mit dem Aufbau einer Kamera und den Möglichkeiten, ihre Projektion mathematisch zu beschreiben. Als besonderes Hilfsmittel wird der Begriff des *Projektiven Raums* eingeführt. In Abschnitt 4.2 wird das spezielle Problem der 2-Bild-Geometrie untersucht, das eine besondere Rolle bei dem Vergleich eines Bildes mit einem zuvor angefertigten „Gedächtnisbild“ spielt. Im letzten Abschnitt geht es um die Bestimmung der *Projektionsmatrix*, die die 3D-zu-2D-Abbildung einer Kamera beschreibt.

4.1 Die Geometrie des Kamerabilds

4.1.1 Das Lochkameramodell

In der Geschichte der Fotografie gehen Berichte über die Verwendung der Lochkamera bis in die Antike zurück. Die *Camera Obscura*, wie sie auch genannt wird, besteht aus einem lichtundurchlässigen Behälter, der auf einer Seite ein Loch besitzt. Durch dieses Loch kann Licht ins Innere des Behälters fallen, wo es auf eine Rückwand trifft. Tatsächliche Lochkameras waren groß genug, um einen Menschen aufnehmen zu können, der nun das auf die Rückwand projizierte Bild betrachten konnte. Später wurden kleinere Lochkameras gebaut, die eine halbtransparente Rückwand oder eine Spiegelvorrichtung enthielten, sodass die Projektion auch von außen wahrgenommen werden konnte.[Wik05]

Für unsere Zwecke besitzt die Lochkamera die Eigenschaft, sich gut mathematisch abstrahieren zu lassen: Stellen wir uns das Loch als das Kamerazentrum $\boldsymbol{C} \in \mathbb{R}^3$ vor. Die Rückwand der Kamera definiert eine Ebene $\boldsymbol{\mathcal{P}}$, die $\boldsymbol{C}$ nicht enthält. Jeder Punkt $\boldsymbol{M} \neq \boldsymbol{C}$ erzeugt eine Gerade $\overline{\boldsymbol{MC}}$. Ist $\overline{\boldsymbol{MC}}$ *nicht parallel* zu $\boldsymbol{\mathcal{P}}$, so existiert ein Schnittpunkt $\boldsymbol{m}$ mit der Ebene.

Wir können durch Wahl eines Koordinatensystems erreichen, dass $\boldsymbol{C} = \boldsymbol{0}$ und $\boldsymbol{\mathcal{P}} = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{f})^{\boldsymbol{T}} |\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}\}$. Es gibt einen Punkt $\boldsymbol{p} \in \boldsymbol{\mathcal{P}}$, der durch die Eigenschaft $\overline{\boldsymbol{Cp}} \perp \boldsymbol{\mathcal{P}}$ ausgezeichnet ist. Dieser Punkt wird *Bildhauptpunkt* genannt. Der Abstand $\boldsymbol{f}$ zwischen $\boldsymbol{p}$ und $\boldsymbol{C}$ heißt *Brennweite*. In dieser Konfiguration ist leicht zu sehen, dass $\boldsymbol{m} = (\boldsymbol{fx/z}, \boldsymbol{fy/z}, \boldsymbol{f})^{\boldsymbol{T}}$ für $\boldsymbol{M} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})^{\boldsymbol{T}}$.

Bisher haben wir nur den Punkt $\boldsymbol{X}$ auf die Ebene $\boldsymbol{\mathcal{P}}$ projiziert. Für eine 3D-2D-Transformation fehlt noch die Definition eines 2D-Koordinatensystems auf der Bildebene. Dies erreichen wir auf einfachste Weise durch Weglassen der dritten Koordinate: $\boldsymbol{m'} = (\boldsymbol{fx/z}, \boldsymbol{fy/z})^{\boldsymbol{T}}$.

Die beschriebene Projektion lässt sich durch die Einführung *homogener Koordinaten* in 4.1.2 sehr einfach darstellen.

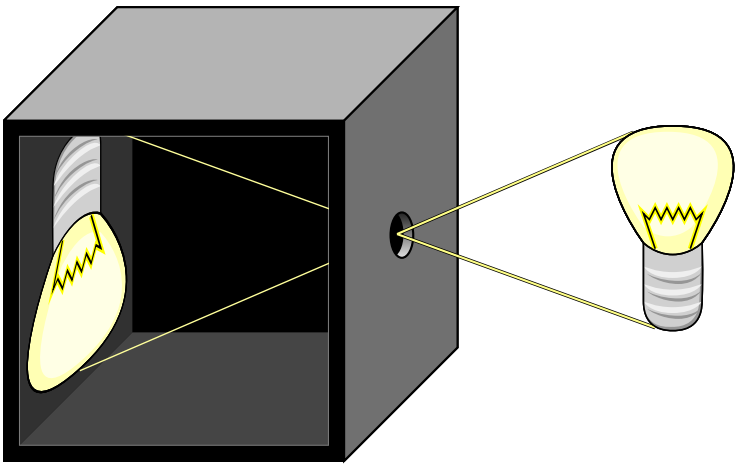


Abbildung 4.1: Illustration einer Lochkamera im Aufschnitt. Licht fällt von der Vorderseite durch ein Loch und erzeugt auf der Rückwand der Kamera eine Projektion. Diese erscheint um 180° verdreht, da sich alle Strahlen in einem Punkt treffen.

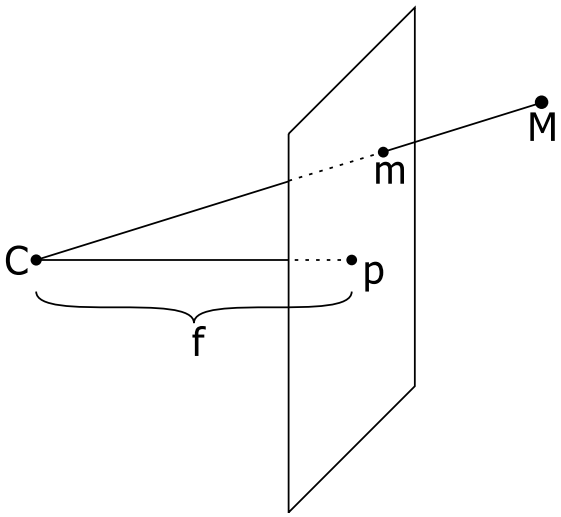


Abbildung 4.2: Strahlengang einer idealisierten Lochkamera. Es ist üblich, die Projektionsebene *vor* die Kamera zu setzen, damit die Orientierung des Bildes erhalten bleibt.

4.1.2 Einführung homogener Koordinaten

In diesem Abschnitt sollen der *Projektive Raum* und *homogene Koordinaten* eingeführt werden. Der *Projektive Raum der Dimension* $\boldsymbol{n}$, $\mathbb{P}^{\boldsymbol{n}}$, ist die Menge der eindimensionalen Untervektorräume des $\mathbb{R}^{\boldsymbol{n+1}}$. Seine Elemente werden durch Repräsentanten dargestellt. Jeder Vektor $\boldsymbol{v} \in \mathbb{R}^{\boldsymbol{n+1}}, \boldsymbol{v} \neq \boldsymbol{0}$ ist ein Repräsentant des Untervektorraums $\mathbb{R}\boldsymbol{v}$. Somit sind zwei Vektoren $\boldsymbol{a}, \boldsymbol{b} \neq \boldsymbol{0} \in \mathbb{R}^{\boldsymbol{n+1}}$ Repräsentanten desselben homogenen Vektors genau dann wenn $\boldsymbol{a} = \boldsymbol{\lambda b}$. [1]

Ein Vorteil dieser Koordinaten ist, dass sie den euklidischen Raum um weitere Transformationen bereichern. Wir betten den $\mathbb{R}^{\boldsymbol{n}}$ in den $\mathbb{P}^{\boldsymbol{n}}$ ein, indem jedes $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ mit dem Repräsentant $[\boldsymbol{x}, \boldsymbol{1}]^{\boldsymbol{T}}$ den $\mathbb{P}^{\boldsymbol{n}}$ identifiziert wird. Somit können wir die *projektiven* Abbildungen (die die *affinen* Abbildungen mit einschließen) als lineare Abbildungen darstellen.

Die affinen Abbildungen sind vom Typ $\boldsymbol{x'} = \boldsymbol{Ax} + \boldsymbol{b}$ (für eine Matrix $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m \times n}}$ und einen Vektor $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{n}}$) und können in homogenen Koordinaten wie folgt dargestellt werden:

$$\tilde{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{1} \end{bmatrix}$$

Die projektiven Abbildungen sind vom Typ $\boldsymbol{x'} = (\boldsymbol{Ax} + \boldsymbol{b})/(\boldsymbol{c^T x} + \boldsymbol{d})$. In homogenen Koordinaten lassen sie sich so ausdrücken:

$$\tilde{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{c^T} & \boldsymbol{d} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{1} \end{bmatrix}$$

Die Division geschieht hier erst bei der „Rückkonvertierung“ in den $\mathbb{R}^{\boldsymbol{n}}$, indem man den homogenen Vektor $\tilde{\boldsymbol{x}}'$ durch Division mit der letzten Komponente wieder in die Form $\tilde{\boldsymbol{x}}' = [\boldsymbol{x^T}, \boldsymbol{1}]^{\boldsymbol{T}}$ bringt. Was ist, wenn diese Rückkonvertierung nicht möglich ist, weil die letzte Komponente $\boldsymbol{0}$ ist? In diesem Fall beschreibt $\tilde{\boldsymbol{x}}'$ immer noch einen zulässigen homogenen Vektor, der allerdings kein *inhomogenes* Äquivalent besitzt. Solche Vektoren nehmen die wichtige Rolle der *idealen Punkte* ein. Eine weiter gehende Behandlung der homogenen Koordinaten findet z.B. in [HZ00] statt.

4.1.3 Die Projektionsmatrix P

In 4.1.1 haben wir die Projektion in einer Kamera mit folgender Gleichung beschrieben:

$$\begin{pmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \end{pmatrix} = \begin{pmatrix} \boldsymbol{fx/z} \\ \boldsymbol{fy/z} \end{pmatrix}$$

Mittels homogener Koordinaten können wir diese projektive Abbildung als Matrixmultiplikation darstellen:

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ & \boldsymbol{f} & \boldsymbol{0} & \boldsymbol{0} \\ & & \boldsymbol{1} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{1} \end{bmatrix}$$

Die $\boldsymbol{3 \times 4}$-Matrix, die einen Punkt aus dem $\mathbb{P}^{\boldsymbol{3}}$ auf einen Punkt im $\mathbb{P}^{\boldsymbol{2}}$ abbildet, wird Projektionsmatrix $\boldsymbol{P}$ genannt. In der zuvor gezeigten Form tritt sie allerdings nur unter den

[1] Um darzustellen, dass es sich um homogene Vektoren handelt, werden in dieser Arbeit Vektorsymbole mit einem Tildezeichen (z.B. $\tilde{\boldsymbol{x}}$) versehen. Als Zeilen- oder Spaltenvektoren geschriebene Repräsentanten stehen in eckigen Klammern.

Annahmen auf, die in 4.1.1 gemacht wurden: $\boldsymbol{C}$ ist der Koordinatenursprung, der Bildhauptpunkt $\boldsymbol{p}$ hat die Koordinaten $\boldsymbol{(0, 0, f)}$. Im allgemeinen Fall lässt sich $\boldsymbol{P}$ darstellen als:

$$\boldsymbol{P = K[R \mid t\,]} \tag{4.1}$$

wobei $\boldsymbol{K} \in \mathbb{R}^{\mathbf{3\times3}}$ eine rechte obere Dreiecksmatrix ist, deren Einträge eine geometrische Bedeutung zukommt, die in Abschnitt 4.1.4 erläutert wird. Sie wird *Kalibrationsmatrix* genannt. $\boldsymbol{R} \in \mathbb{R}^{\mathbf{3\times3}}$ ist eine Rotationsmatrix (also $\boldsymbol{R^{-1} = R^T}$ und $\mathbf{det}\,\boldsymbol{R} = \mathbf{1}$) und $\boldsymbol{t} \in \mathbb{R}^{\mathbf{3}}$ ein beliebiger Translationsvektor. Hartley demonstriert in [HZ00, A3.1.1 S.552], wie man mit Hilfe einer RQ-Zerlegung jede Projektionsmatrix in dieser Form darstellen kann.

Diese Trennung stellt die Zerlegung der Projektionsmatrix in *extrinsische* und *intrinsische* Kalibrationsparameter dar. Die Kalibrationsmatrix $\boldsymbol{K}$ enthält die *intrinsischen* Kalibrationsparameter. Das sind diejenigen Parameter, die nur von Mechanismen innerhalb der Kamera beeinflusst werden. Sie sind invariant unter Rotation und Translation der Kamera, welche durch die *extrinsischen* Kalibrationsparameter $\boldsymbol{R}$ und $\boldsymbol{t}$ ausgedrückt werden. [2]

4.1.4 Die Kalibrationsmatrix K

In diesem Abschnitt soll die Bedeutung der einzelnen Einträge der Kalibrationsmatrix $\boldsymbol{K}$ untersucht werden. Da homogene Koordinaten verwendet werden, ist $\boldsymbol{K}$ skalierungsinvariant, d.h. es lässt sich normieren zu einer Form, in der der letzte Eintrag gleich 1 ist:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{s} & \boldsymbol{p_x} \\ & \boldsymbol{\beta} & \boldsymbol{p_y} \\ & & \mathbf{1} \end{bmatrix} \tag{4.2}$$

$\boldsymbol{\alpha, \beta}$ Die Einträge $\boldsymbol{\alpha}$ und $\boldsymbol{\beta}$ ersetzen den einzelnen Parameter $\boldsymbol{f}$ für die Brennweite. Man benötigt zwei Parameter, wenn die Skalierung für die X- und die Y-Achse nicht identisch ist. Dies ist bei rechteckigen Bildpunkten der Fall. Im Videostandard PAL DV beträgt das Seitenverhältnis eines Pixels beispielsweise 59/54.

$\boldsymbol{p_x, p_y}$ In 4.1.1 waren wir davon ausgegangen, dass der Bildhauptpunkt $\boldsymbol{p}$ mit dem Ursprung des Bildkoordinatensystems übereinstimmt. Das ist oft nicht der Fall, beispielsweise wenn sich der Koordinatenursprung, wie in der Computergrafik üblich, in der linken unteren (oder oberen) Bildecke befindet. In diesem Fall liegt der Bildhauptpunkt in der Mitte des Bildes auf den Koordinaten (Breite/2, Höhe/2). Auch baulich bedingt kann es in realen Kameras zu einer Verschiebung des Bildhauptpunktes kommen.

$\boldsymbol{s}$ Der *skew*-Parameter $\boldsymbol{s}$ beschreibt eine parallelogrammartige Verzerrung des Bildes. Da diese in der Praxis selten vorkommt, wird oft $\boldsymbol{s = 0}$ angenommen. Ein realer Ursprung für eine solche Verzerrung wäre ein schief eingebauter Bildsensor in einer Digitalkamera oder eine Situation, in der ein Bild von einem Bild gemacht wurde.[HZ00, 5.1 S.143]

4.1.5 Linsenverzeichnung

Bisher sind wir immer davon ausgegangen, dass sich die Projektion einer Kamera hinreichend gut durch ein projektives Modell beschreiben lässt. Für eine Lochkamera mag dies gelten.

[2] Genau genommen beschreiben $\boldsymbol{R}$ und $\boldsymbol{t}$ nicht die Rotation und Translation der *Kamera* im *Weltkoordinatensystem*, sondern die Rotation und anschließende Translation der *Welt* im *Kamerakoordinatensystem*.

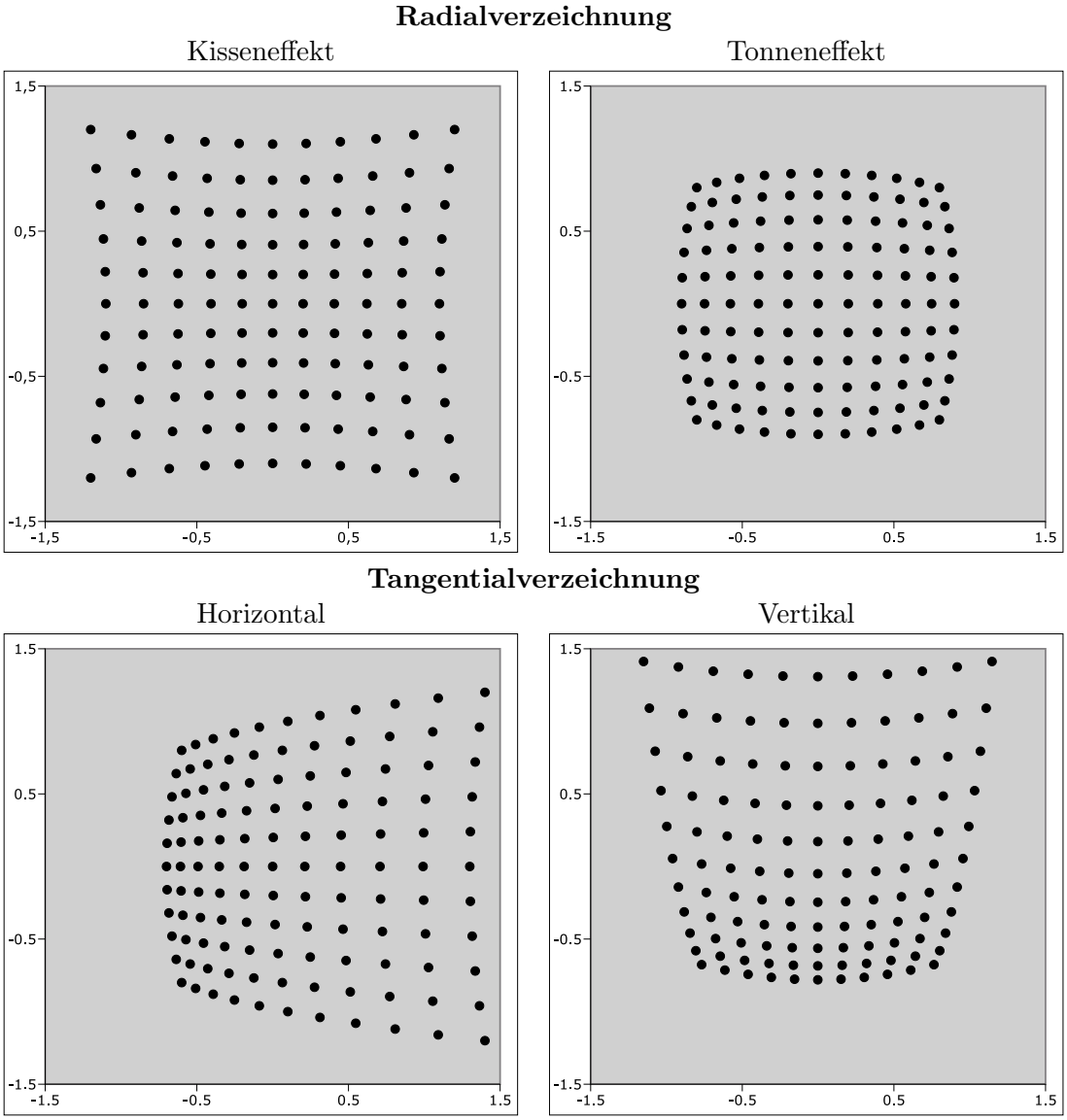


Abbildung 4.3: Linsenverzeichnungseffekte, die mit den Heikkilä-Parametern modelliert werden können

Die Linsenoptik in heutigen Kameras erzeugt jedoch eine deutliche Verzerrung, besonders in Weitwinkelobjektiven. In *Fischaugenobjektiven* ist dieser Effekt sogar erwünscht und ermöglicht ein besonders großes Sichtfeld. Für praktische Zwecke in der Bilderkennung muss die Linsenverzeichnung jedoch ausgeglichen werden.

Die wichtigsten Verzeichnungseffekte sind die radiale und die tangentiale Verzeichnung, wobei der größte Anteil bei herkömmlichen Objektiven auf die radiale Verzeichnung entfällt. Eine Radiale Verzeichnung nach außen löst einen *Nadelkisseneffekt* aus, eine Verzeichnung nach innen einen *Tonneneffekt*. Die Verzeichnung wird als Funktion $\boldsymbol{\Delta : \mathbb{P}^2 \to \mathbb{P}^2}$ definiert und die Projektionsgleichung erweitert zu:

$$\tilde{\boldsymbol{m}} = \boldsymbol{K\Delta}(\ [\boldsymbol{R}|\boldsymbol{t}]\ \tilde{\boldsymbol{M}}\ ) \tag{4.3}$$

Je nach Modellierung der Linsenverzeichnung nimmt $\boldsymbol{\Delta}$ eine andere Form an. [HZ00, 6.4] verwendet nur die radiale Verzerrung $\boldsymbol{L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \ldots}$:

$$\boldsymbol{\Delta} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}(\|\boldsymbol{X}\|)\boldsymbol{X} \\ \boldsymbol{1} \end{bmatrix}$$

Dabei werden Parameter bis zu $\boldsymbol{\kappa_4}$ verwendet.

Heikkilä und Silven definieren in [HS97] die Linsenverzeichnung als Funktion

$$\boldsymbol{\Delta}[\boldsymbol{X}, \boldsymbol{1}]^{\boldsymbol{T}} = (\boldsymbol{X} + \boldsymbol{\delta}^{(r)}(\boldsymbol{X}) + \boldsymbol{\delta}^{(t)}(\boldsymbol{X}),\ \boldsymbol{1})^{\boldsymbol{T}}$$

mit radialer Verzeichnung

$$\boldsymbol{\delta}^{(r)}(\boldsymbol{X}) = (\boldsymbol{k_1 r^2 + k_2 r^4 + \ldots})\ \boldsymbol{X}$$

und tangentialer Verzeichnung

$$\boldsymbol{\delta}^{(t)} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{2p_1 xy + p_2(r^2 + 2x^2)} \\ \boldsymbol{p_1(r^2 + 2y^2) + 2p_2 xy} \end{pmatrix}$$

wobei $\boldsymbol{X = (x, y)^T}$ und $\boldsymbol{r^2 = \|X\|^2 = x^2 + y^2}$. $\boldsymbol{p_1, p_2}$ sowie $\boldsymbol{k_1, k_2, \ldots}$ (häufig bis $\boldsymbol{k_2}$ oder $\boldsymbol{k_3}$) sind die Parameter. [3]

Mit der eingeführten Linsengeometrie wurde das lineare Kameramodell um eine nichtlineare Komponente erweitert. Müssen nun alle mathematischen Verfahren, die mit Bildkoordinaten arbeiten, diesen Umstand berücksichtigen? Die Antwort ist „Nein," denn es ist möglich, durch Entzerrung des Bildes diese nichtlineare Komponente herauszufiltern. Dies kann geschehen, bevor geometrische Eigenschaften des Bildes ermittelt werden.

4.2 Die Geometrie von zwei Bildern

Von elementarer Bedeutung im Computersehen ist das Verständnis über den Zusammenhang zweier verschiedener Bilder der gleichen zugrunde liegenden Szene. In den folgenden Abschnitten soll beschrieben werden, durch welche Einflüsse sich zwei Bilder unterscheiden und welche Schlüsse man aus den Unterschieden zweier Bilder ziehen kann.

[3] Tatsächlich operiert die Funktion $\boldsymbol{\Delta}$ in [HS97] nicht auf $\boldsymbol{X}$, sondern auf $\boldsymbol{fX}$. Die Multiplikation mit der Brennweite wird also in die Verzerrung miteinbezogen. Bis auf Parameterwerte ändert dies aber nichts am Modell.

4.2.1 Spezialfall: Planare Geometrie

Zuerst soll ein Spezialfall der Zweibildgeometrie untersucht werden. Es handelt sich um eine Klasse von Fällen, in denen zwei Bilder durch eine zweidimensionale projektive Transformation zur Deckung gebracht werden können, ohne die darunter liegende dreidimensionale Geometrie zu kennen. Genauer soll für jeden Punkt $\tilde{\boldsymbol{m}}$ des einen Bildes ein korrespondierender Punkt $\tilde{\boldsymbol{m}}'$ des anderen Bildes existieren mit:

$$\tilde{\boldsymbol{m}}' = \boldsymbol{H}\tilde{\boldsymbol{m}} \tag{4.4}$$

wobei $\boldsymbol{H}$ eine Matrix aus $\mathbb{R}^{\mathbf{3}}$ ist. Eine lineare Abbildung im projektiven Raum nennt man *Homographie*.

Die wichtigsten Situationen, in denen die Beziehung zweier Bilder durch eine Homographie beschrieben ist, sind:

1. Die Bilder werden aus der gleichen Position, aber mit unterschiedlicher Rotation und/oder Kameraeinstellungen erzeugt.

2. Es handelt sich um Bilder einer Ebene.

Fall 1 lässt sich wie folgt formalisieren. Durch freie Koordinatenwahl des Weltkoordinatensystems können wir festlegen, dass das Zentrum der Kamera im ersten Bild auf dem Ursprung und die Bildhauptachse auf der Z-Achse liegt. Auch in der Bildebene besteht freie Wahl des Koordinatensystems. Wir bestimmen, dass die Kalibrationsmatrix des ersten Bildes gleich der Identität ist. Somit ist die Projektionsmatrix $\boldsymbol{P_1} = \boldsymbol{I}[\boldsymbol{I}|\boldsymbol{0}] = [\boldsymbol{I}|\boldsymbol{0}]$. Die Projektionsmatrix des zweiten Bildes hat dann die Form $\boldsymbol{P_2} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{0}]$. Sei $\tilde{\boldsymbol{M}} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{1}]^{\boldsymbol{T}}$ ein 3D-Punkt:

$$\begin{array}{lllll} \tilde{\boldsymbol{m}} & = \boldsymbol{P_1}\tilde{\boldsymbol{M}} = & [\boldsymbol{I}|\boldsymbol{0}]\tilde{\boldsymbol{M}} = & [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{1}]^{\boldsymbol{T}} \\ \tilde{\boldsymbol{m}}' & = \boldsymbol{P_2}\tilde{\boldsymbol{M}} = & \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{0}]\tilde{\boldsymbol{M}} = & \boldsymbol{KR}[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{1}]^{\boldsymbol{T}} \end{array}$$

Also ist $\tilde{\boldsymbol{m}}' = \boldsymbol{KR}\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{m}}$.

Für Fall 2 wählen wir unser Koordinatensystem wieder wie oben: $\boldsymbol{P_1} = [\boldsymbol{I}|\boldsymbol{0}]$, $\boldsymbol{P_2} = [\boldsymbol{A}|\boldsymbol{a}]$. Die Ebene sei die Menge aller Punkte $\tilde{\boldsymbol{x}}$ mit $[\boldsymbol{v}, \boldsymbol{1}]\tilde{\boldsymbol{x}} = \boldsymbol{0}$. Durch Rückprojektion des Punktes $\tilde{\boldsymbol{m}}$ auf die Ebene erhält man $\tilde{\boldsymbol{M}} = [\tilde{\boldsymbol{m}}^{\boldsymbol{T}}, -\boldsymbol{v}^{\boldsymbol{T}}\tilde{\boldsymbol{m}}]^{\boldsymbol{T}}$ und somit $\tilde{\boldsymbol{m}}' = [\boldsymbol{A}|\boldsymbol{a}]\tilde{\boldsymbol{M}} = (\boldsymbol{A} - \boldsymbol{av}^{\boldsymbol{T}})\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{m}}$ [HZ00, 12.1 S.312f].

Bestimmung der Homographie (DLT)

Die Homographie $\boldsymbol{H}$ zwischen zwei Bildern kann aus 4 Punkt-zu-Punkt-Korrelationen mit Punkten in allgemeiner Lage berechnet werden. Bei mehr als 4 Korrespondenzen ist das Problem überbestimmt und es existiert eine exakte Lösung im Allgemeinen nicht mehr. Jedoch kann eine Approximation ermittelt werden. Für jede Korrespondenz gilt die Bedingung $\tilde{\boldsymbol{m}}' = \boldsymbol{H}\tilde{\boldsymbol{m}}$ und somit:

$$\begin{array}{rcl} \tilde{\boldsymbol{m}}' \times \boldsymbol{H}\tilde{\boldsymbol{m}} & = & \boldsymbol{0} \\ \Leftrightarrow \quad \tilde{\boldsymbol{m}}' \times \begin{bmatrix} \boldsymbol{h_1^T} \\ \boldsymbol{h_2^T} \\ \boldsymbol{h_3^T} \end{bmatrix} \tilde{\boldsymbol{m}} & = & \boldsymbol{0} \end{array}$$

was sich umformen lässt zu:

$$[\tilde{\boldsymbol{m}}']_{\times} \begin{bmatrix} \tilde{\boldsymbol{m}}^{\boldsymbol{T}} & & \\ & \tilde{\boldsymbol{m}}^{\boldsymbol{T}} & \\ & & \tilde{\boldsymbol{m}}^{\boldsymbol{T}} \end{bmatrix} \begin{pmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \\ \boldsymbol{h_3} \end{pmatrix} = \boldsymbol{0} \tag{4.5}$$

$[\boldsymbol{a}]_\times$ bezeichnet die schiefsymmetrische Matrix

$$\begin{bmatrix} \mathbf{0} & -\boldsymbol{a_3} & \boldsymbol{a_2} \\ \boldsymbol{a_3} & \mathbf{0} & -\boldsymbol{a_1} \\ -\boldsymbol{a_2} & \boldsymbol{a_1} & \mathbf{0} \end{bmatrix}$$

sodass $[\boldsymbol{a}]_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$. Die Determinante der Matrix $[\tilde{\boldsymbol{m}}']_\times$ ist $\mathbf{0}$, somit sind die Zeilen linear abhängig. Die dritte Zeile ist also redundant und kann weggelassen werden. Sei $\tilde{\boldsymbol{m}} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}]^T$ und $\tilde{\boldsymbol{m}}' = [\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{w}']^T$, dann resultiert aus Gleichung 4.5 die Bedingung [HZ00, 3.1 S.72]:

$$\begin{bmatrix} \mathbf{0}^T & -\boldsymbol{w}'\tilde{\boldsymbol{m}}^T & \boldsymbol{y}'\tilde{\boldsymbol{m}}^T \\ \boldsymbol{w}'\tilde{\boldsymbol{m}}^T & \mathbf{0}^T & -\boldsymbol{x}'\tilde{\boldsymbol{m}}^T \end{bmatrix} \underbrace{\begin{pmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \\ \boldsymbol{h_3} \end{pmatrix}}_{\boldsymbol{h}} = \mathbf{0}$$

Die Matrix $\boldsymbol{H}$ ist hier zu einem 9-elementigen Spaltenvektor $\boldsymbol{h}$ transformiert worden. Jede Korrespondenz trägt auf diese Weise zwei Zeilen zu einem System $\boldsymbol{Ah} = \mathbf{0}$ bei. Mit vier Korrespondenzen hat $\boldsymbol{A}$ acht Zeilen, was zusammen mit $\|\boldsymbol{h}\|^2 = \mathbf{1}$ genügt, um die neun Einträge von $\boldsymbol{h}$ eindeutig zu bestimmen. Bei mehr als vier Korrespondenzen lässt sich ein Vektor $\hat{\boldsymbol{h}}$ bestimmen, der $\|\boldsymbol{A}\hat{\boldsymbol{h}}\|^2$ minmiert. [4]

Der beschriebene Algorithmus zur Schätzung einer Lösung über ein lineares Gleichungssystem wird *Direkte Lineare Transformation (DLT)* genannt.

Normalisierung

Der DLT-Algorithmus hat eine Eigenschaft, die ihn problematisch macht: Für mehr als vier Korrespondenzen ist er nicht invariant unter linearer Transformation. Sei durch Korrespondenzen $\tilde{\boldsymbol{m}} \leftrightarrow \tilde{\boldsymbol{m}}'$ mittels des DLT eine Homographie $\boldsymbol{H}$ bestimmt worden. Eine Homographie, die Korrespondenzen $\boldsymbol{U}\tilde{\boldsymbol{m}} \leftrightarrow \boldsymbol{V}\tilde{\boldsymbol{m}}'$ miteinander verbindet, ist somit durch $\boldsymbol{VHU^{-1}}$ gegeben. Allerdings wird der DLT-Algorithmus, angewandt auf das transformierte Problem, im Allgemeinen ein $\boldsymbol{H}' \neq \boldsymbol{VHU^{-1}}$ liefern. Es gibt also Unterschiede im Ergebnis durch die Transformation eines Koordinatensystems, und die Qualität des Ergebnisses hängt sehr stark von dieser Wahl ab.[HZ00, 3.4.2 S.89ff]

Als günstig hat sich deshalb folgende Normalisierungsstrategie erwiesen: Finde Matrizen $\boldsymbol{U}$ und $\boldsymbol{V}$ vom Typ

$$\begin{bmatrix} \boldsymbol{s} & & \boldsymbol{u} \\ & \boldsymbol{s} & \boldsymbol{v} \\ & & \mathbf{1} \end{bmatrix}$$

sodass für die Korrespondenzen $\boldsymbol{U}\tilde{\boldsymbol{m}}_{\boldsymbol{i}}$ und $\boldsymbol{V}\tilde{\boldsymbol{m}}'_{\boldsymbol{i}}$ gilt:

- Der Schwerpunkt liegt bei $[\mathbf{0}, \mathbf{0}, \mathbf{1}]^T$.
- Der durchschnittliche Abstand zum Ursprung beträgt $\sqrt{\mathbf{2}}$.

Der DLT-Algorithmus ermittelt nun eine Homographie $\hat{\boldsymbol{H}}$ mit $\boldsymbol{V}\tilde{\boldsymbol{m}}' \approx \hat{\boldsymbol{H}}\boldsymbol{U}\tilde{\boldsymbol{m}}$. Diese kann man nun in eine Homographie $\boldsymbol{H}$ für das ursprüngliche Koordinatensystem umwandeln: $\tilde{\boldsymbol{m}}' \approx \boldsymbol{V^{-1}}\hat{\boldsymbol{H}}\boldsymbol{U}\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{m}}$

[4] Die exakte Lösung im ersten Fall kann durch ein Gauss-Verfahren bestimmt werden. Die Approximation $\hat{\boldsymbol{h}}$ ist gegeben durch den *rechten Singulärvektor* zum *kleinsten Singulärwert* der Matrix $\boldsymbol{A}$. [HZ00, A3.4.2 S.563f]

4.2.2 Allgemeiner Fall: Epipolargeometrie

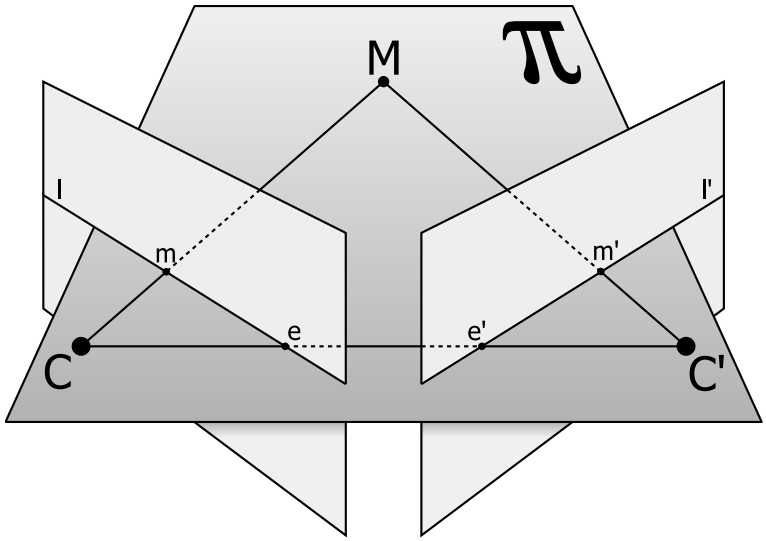


Abbildung 4.4: Epipolargeometrie – zwei Kamerazentren C und C' sowie ein Punkt M erzeugen eine Ebene $\boldsymbol{\pi}$. Die Projektionen von $\boldsymbol{M}$ und $\boldsymbol{C'}$ auf die Projektionsebene, $\boldsymbol{m}$ und $\boldsymbol{e}$, erzeugen eine Gerade $\boldsymbol{l}$.

Während sich 4.2.1 mit dem Spezialfall beschäftigt, dass der Zusammenhang zweier Bilder allein durch eine 2-dimensionale projektive Abbildung beschrieben ist, beschäftigt sich dieser Abschnitt mit der Frage, welche Gesetze im allgemeinen Fall gelten, in welchem eine solche Abbildung nicht existiert. Es wird sich zeigen, dass jeder Punkt des einen Bildes eine Gerade im anderen Bild festlegt, auf der sich der korrespondierende Punkt befinden muss.

Zwei Kameras mit Zentren $\boldsymbol{C}$ und $\boldsymbol{C'}$ und ein Punkt $\boldsymbol{M}$ in allgemeiner Lage definieren eine Ebene $\boldsymbol{\pi}$ (s. Abb. 4.4). Ebenfalls in dieser Ebene liegen die Projektionen $\boldsymbol{m}$ und $\boldsymbol{m'}$ von $\boldsymbol{M}$ auf die Bildebenen der Kameras. Seien nun $\boldsymbol{M}$ und $\boldsymbol{m'}$ unbekannt und $\boldsymbol{m'}$ der zu bestimmende Punkt. Über $\boldsymbol{M}$ ist nur bekannt, dass es sich auf der Geraden $\overline{\boldsymbol{Cm}}$ befindet, welche in $\boldsymbol{\pi}$ liegt. Die Ebene $\boldsymbol{\pi}$ wird auf eine Gerade $\boldsymbol{l'}$ in der Projektionsebene der zweiten Kamera projiziert, sodass sich $\boldsymbol{m'}$ ebenfalls in $\boldsymbol{l'}$ befinden muss. Die Projektion des Zentrums der einen Kamera auf die Projektionsebene der anderen Kamera heißt *Epipol* und wird mit $\boldsymbol{e}$ bzw. $\boldsymbol{e'}$ bezeichnet.

Ist die Situation umgekehrt, also $\boldsymbol{m'}$ bekannt und $\boldsymbol{m}$ zu bestimmen, so liegt $\boldsymbol{m}$ auf der Geraden $\boldsymbol{l}$, die durch Projektion von $\boldsymbol{\pi}$ auf die Projektionsebene der ersten Kamera zustande kommt.

Die Fundamentalmatrix F Es gibt eine Abbildung, die jeden Punkt $\boldsymbol{m}$ auf eine Gerade $\boldsymbol{l'}$ und jeden Punkt $\boldsymbol{m'}$ auf eine Gerade $\boldsymbol{l}$ projiziert. [5] Xu zeigt in [XZ96] (ebenfalls in [HZ00, 8.2.2 S.224]), dass diese Abbildung projektiv ist und durch

$$\tilde{\boldsymbol{m}}'^{\boldsymbol{T}}\boldsymbol{F}\tilde{\boldsymbol{m}} = \boldsymbol{0} \tag{4.6}$$

[5] Eine Gerade im $\mathbb{R}^{\boldsymbol{2}}$, deren Punkte $(\boldsymbol{x}, \boldsymbol{y})^{\boldsymbol{T}}$ die Gleichung $\boldsymbol{ax + by + c = 0}$ erfüllen, lässt sich als homogener Vektor $\tilde{\boldsymbol{l}} = [\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]^{\boldsymbol{T}}$ schreiben. Es gilt dann $\tilde{\boldsymbol{x}}^{\boldsymbol{T}}\tilde{\boldsymbol{l}} = \tilde{\boldsymbol{l}}^{\boldsymbol{T}}\tilde{\boldsymbol{x}} = \boldsymbol{0}$ für alle Punkte $\tilde{\boldsymbol{x}} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{1}]^{\boldsymbol{T}}$ auf der Geraden.

beschrieben ist. Dabei ist $\boldsymbol{F} \in \mathbb{R}^{\mathbf{3\times 3}}$ eine Rang-2-Matrix und wird *Fundamentalmatrix* genannt. Aus Gleichung 4.6 folgt direkt:

$$\boldsymbol{F}\tilde{\boldsymbol{m}} = \boldsymbol{l'}$$

$$\tilde{\boldsymbol{m}}'^{\boldsymbol{T}}\boldsymbol{F} = \boldsymbol{l}^{\boldsymbol{T}}$$

Bedeutung der Fundamentalmatrix Die Fundamentalmatrix ist außerordentlich wichtig bei der Suche nach Punkt-Korrespondenzen in zwei Bildern ohne vorhergehendes Wissen über die dreidimensionale Geometrie der Szene. Die Gleichung 4.6 ermöglicht bei einem gegebenen Punkt $\boldsymbol{m}$ den Suchraum für den korrespondierenden Punkt $\boldsymbol{m'}$ vom gesamten Bild auf eine Gerade einzuschränken.

Weiterhin ist es möglich, bei bekannter Fundamentalmatrix und bekannter Projektionsmatrix $\boldsymbol{P}$ die Projektionsmatrix $\boldsymbol{P'}$ des anderen Bildes zu bestimmen. [HZ00, 8.5.3 S.238]

8-Punkte Algorithmus

Der einfachste Algorithmus, um die Fundamentalmatrix zu bestimmen, soll nun eingeführt werden. Mit $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3}]^{\boldsymbol{T}}$ lässt sich Gleichung 4.6 umformen zu:

$$\tilde{\boldsymbol{m}}'^{\boldsymbol{T}} \begin{bmatrix} \tilde{\boldsymbol{m}}^{\boldsymbol{T}} & & \\ & \tilde{\boldsymbol{m}}^{\boldsymbol{T}} & \\ & & \tilde{\boldsymbol{m}}^{\boldsymbol{T}} \end{bmatrix} \begin{pmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \end{pmatrix} = \boldsymbol{0}$$

und somit für $\tilde{\boldsymbol{m}}' = [\boldsymbol{x'}, \boldsymbol{y'}, \boldsymbol{w'}]^{\boldsymbol{T}}$:

$$[\boldsymbol{x'}\tilde{\boldsymbol{m}}^{\boldsymbol{T}}, \boldsymbol{y'}\tilde{\boldsymbol{m}}^{\boldsymbol{T}}, \boldsymbol{w'}\tilde{\boldsymbol{m}}^{\boldsymbol{T}}] \underbrace{\begin{pmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \end{pmatrix}}_{\boldsymbol{f}} = \boldsymbol{0} \tag{4.7}$$

Dies liefert pro Korrespondenz $\tilde{\boldsymbol{m}}_{\boldsymbol{i}} \leftrightarrow \tilde{\boldsymbol{m}}_{\boldsymbol{i}}{}'$ eine Zeile für eine Matrix $\boldsymbol{A}$. Bei acht Korrespondenzen dieses Typs lässt sich das Gleichungssystem Typ $\boldsymbol{Af} = \boldsymbol{0}$ lösen, wenn zusätzlich die Skalierung von $\boldsymbol{F}$ durch die Bedingung $\|\boldsymbol{f}\|^{\boldsymbol{2}} = \boldsymbol{1}$ festgelegt wird. Dabei ist $\boldsymbol{f}$ ein Spaltenvektor aus $\mathbb{R}^{\boldsymbol{9}}$. Bei mehr als acht Korrespondenzen ist eine Lösung $\hat{\boldsymbol{f}}$ ermittelbar, die $\|\boldsymbol{Af}\|^{\boldsymbol{2}}$ minimiert. $\hat{\boldsymbol{f}}$ ist gegeben durch den *rechten Singulärvektor* zum *kleinsten Singulärwert* der Matrix $\boldsymbol{A}$. [HZ00, A3.4.2 S.563f]

Erzwingen einer singulären Lösung Wie bereits erwähnt ist die Fundamentalmatrix $\boldsymbol{F}$ singulär und hat Rang 2. Die durch den 8-Punkte-Algorithmus ermittelte Lösung ist jedoch im Allgemeinen nichtsingulär. Die Singularitätsbedingung ist nicht Teil des Lösungsverfahrens und muss nachträglich erzwungen werden. Darunter leidet die Qualität der Lösung.

Um die Singularität zu erzwingen, wird zuerst eine Singulärwertzerlegung durchgeführt: $\boldsymbol{F} = \boldsymbol{U}\,\mathbf{diag}(\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \boldsymbol{\sigma_3})\boldsymbol{V}^{\boldsymbol{T}}$, wobei $\boldsymbol{U}$ und $\boldsymbol{V}$ orthonormal und die $\boldsymbol{\sigma_i} \geq \boldsymbol{0}$ und absteigend sortiert sind. Dann ersetzt man $\boldsymbol{F}$ durch ein $\boldsymbol{F'} = \boldsymbol{U}\,\mathbf{diag}(\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \boldsymbol{0})\boldsymbol{V}^{\boldsymbol{T}}$.

Normalisierung 4.2.1 Wie auch schon der DLT-Algorithmus zur Bestimmung einer Homographie ist der 8-Punkte-Algorithmus nicht invariant unter beliebiger Transformation: Sei $\boldsymbol{F}$ das Ergebnis des 8-Punkte-Algorithmus angewandt auf Korrespondenzen $\tilde{\boldsymbol{m}}_i \leftrightarrow \tilde{\boldsymbol{m}}_i'$ und $\boldsymbol{F'}$ das Ergebnis desselben angewandt auf $\boldsymbol{S}\tilde{\boldsymbol{m}}_i \leftrightarrow \boldsymbol{T}\tilde{\boldsymbol{m}}_i'$. Dann ist zwar $\tilde{\boldsymbol{m}}'^{\boldsymbol{T}}\boldsymbol{T}^{\boldsymbol{T}}\boldsymbol{F'}\boldsymbol{S}\tilde{\boldsymbol{m}} \approx \boldsymbol{0}$, jedoch im allgemeinen $\boldsymbol{T}^{\boldsymbol{T}}\boldsymbol{F'}\boldsymbol{S} \neq \boldsymbol{F}$. [Har95]

Es stellt sich heraus, dass derselbe Normalisierungsmechanismus wie in 4.2.1(S.28) das Ergebnis des 8-Punkte-Algorithmus erheblich verbessern kann:

Finde Matrizen $\boldsymbol{U}$ und $\boldsymbol{V}$ vom Typ

$$\begin{bmatrix} \boldsymbol{s} & & \boldsymbol{u} \\ & \boldsymbol{s} & \boldsymbol{v} \\ & & \boldsymbol{1} \end{bmatrix}$$

sodass für die Korrespondenzen $\boldsymbol{U}\tilde{\boldsymbol{m}}_i$ und $\boldsymbol{V}\tilde{\boldsymbol{m}}_i'$ gilt:

- Der Schwerpunkt liegt bei $[\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{1}]^{\boldsymbol{T}}$.
- Der durchschnittliche Abstand zum Ursprung beträgt $\sqrt{\boldsymbol{2}}$.

Der 8-Punkte-Algorithmus ermittelt nun eine Fundamentalmatrix $\boldsymbol{F'}$ für Korrespondenzen $\boldsymbol{U}\tilde{\boldsymbol{m}}_i \leftrightarrow \boldsymbol{V}\tilde{\boldsymbol{m}}_i'$. Setze $\boldsymbol{F} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{F'}\boldsymbol{U}$ dann ist $\tilde{\boldsymbol{m}}'^{\boldsymbol{T}}\boldsymbol{F}\tilde{\boldsymbol{m}} = \tilde{\boldsymbol{m}}'^{\boldsymbol{T}}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{F'}\boldsymbol{U}\tilde{\boldsymbol{m}} \approx \boldsymbol{0}$.

7-Punkte Algorithmus

Die Fundamentalmatrix hat sieben Freiheitsgrade, also genügen sieben Korrespondenzen um $\boldsymbol{F}$ zu bestimmen. Somit hat die Matrix $\boldsymbol{A}$ die Größe $\boldsymbol{7 \times 9}$ und im Allgemeinen Rang 7. Ihr rechter Nullraum wird somit von zwei Vektoren $\boldsymbol{f_1}$ und $\boldsymbol{f_2} \in \mathbb{R}^{\boldsymbol{9}}$ aufgespannt, die zu Matrizen $\boldsymbol{F_1}$ und $\boldsymbol{F_2}$ gehören. Die Lösung $\boldsymbol{F}$ ist also eine Linearkombination dieser beiden Vektoren. Da $\boldsymbol{F}$ bis auf Skalierung definiert ist, genügt ein Parameter: $\boldsymbol{F} = \boldsymbol{\alpha F_1} + \boldsymbol{(1-\alpha)F_2}$.

Setzt man die Singularitätsbedingung $\boldsymbol{\det(\alpha F_1 + (1-\alpha)F_2) = 0}$ so erhält man eine Gleichung dritten Grades, die entweder genau eine oder drei reelle Lösungen besitzt. Welche der drei möglichen Lösungen die richtige ist, kann durch weitere Korrespondenzen getestet werden.

Weitere Lösungsmethoden

[HZ00, 10.3 S.266f] beschreibt einen Iterationsmechanismus um $\boldsymbol{Af}$ durch eine *singuläre* Matrix zu minimieren. Weiterhin ist es möglich, eine Fundamentalmatrix aus nur sechs Korrespondenzen zu bestimmen, wenn genau vier der Punkte komplanar sind [Zha96, 3.9.1 S.179]. [Zha96] gibt auch einen Überblick über die Möglichkeiten der nichtlinearen Optimierung der Fundamentalmatrix.

4.3 Die Bestimmung der Projektionsmatrix

Von zentraler Bedeutung für jede Anwendung, die versucht, ein Kamerabild mit virtuellen Objekten zu ergänzen, ist die Bestimmung des Kamera- bzw. Projektionsmatrix $\boldsymbol{P}$, eingeführt in 4.1.3. Ist die einem Bild zugrunde liegende dreidimensionale Geometrie bekannt, so kann $\boldsymbol{P}$ direkt aus 3D- zu 2D-Korrespondenzen bestimmt werden. Es muss also genügend viele Bildpunkte $\tilde{\boldsymbol{m}}_i$ geben, deren Position in Weltkoordinaten $\tilde{\boldsymbol{M}}_i$ bekannt ist.

4.3.1 Direkte Bestimmung mittels DLT

Die Situation ist hier analog zu 4.2.1, wo eine Homographie aus 2D-zu-2D-Korrespondenzen bestimmt wurde. Der Unterschied liegt nur in der Dimension des Problems. Für eine Korrespondenz $\tilde{\boldsymbol{m}} \leftrightarrow \boldsymbol{P}\tilde{\boldsymbol{M}}$ gilt die Bedingung:

$$\tilde{\boldsymbol{m}} \times \boldsymbol{P}\tilde{\boldsymbol{M}} = \boldsymbol{0}$$

und somit analog zu Gleichung 4.5 unter $\boldsymbol{P} = [\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}]^{\boldsymbol{T}}$:

$$[\tilde{\boldsymbol{m}}]_\times \begin{bmatrix} \tilde{\boldsymbol{M}}^{\boldsymbol{T}} & & \\ & \tilde{\boldsymbol{M}}^{\boldsymbol{T}} & \\ & & \tilde{\boldsymbol{M}}^{\boldsymbol{T}} \end{bmatrix} \underbrace{\begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \\ \boldsymbol{p_3} \end{pmatrix}}_{\boldsymbol{p}} = \boldsymbol{0} \tag{4.8}$$

Wieder ist die dritte Zeile der Gleichung redundant und kann weggelassen werden. Da P invariant unter Skalierung ist, sind 11 Parameter zu bestimmen. Jeder Punkt trägt zwei Zeilen zu einem Gleichungssystem $\boldsymbol{Ap} = \boldsymbol{0}$ bei, sodass $\boldsymbol{5\frac{1}{2}}$ Punkte genügen, um $\boldsymbol{P}$ unter $\|\boldsymbol{p}\| = \boldsymbol{1}$ auf gewohnte Weise zu bestimmen (s. 4.2.1). Der „halbe Punkt“ muss nur auf eine Koordinate bestimmt sein. Bei mehr Punkten lässt sich wie üblich eine Projektionsmatrix wählen, die $\|\boldsymbol{Ap}\|$ minimiert.

Auch in dieser Variante des DLT-Algorithmus ist eine vorherige Normalisierung der Eingabewerte notwendig. Siehe dazu 4.2.1.

Lage der Punkte Die errechnete Matrix $\boldsymbol{P}$ ist nur dann eine gute Approximation der tatsächlichen Abbildung durch die Kamera, wenn die Punkte $\tilde{\boldsymbol{M}}_{\boldsymbol{i}}$ in *allgemeiner Lage* zueinander sind. Insbesondere der Fall, dass alle Punkte (oder alle bis auf einen) auf einer Ebene liegen, erzeugt keine eindeutig bestimmte Matrix. Gleiches gilt für Fälle, die nur sehr geringfügig von der beschriebenen Konfiguration abweichen, beispielsweise Aufnahmen aus großer Höhe. Hier ist die geschätzte Matrix $\boldsymbol{P}$ sehr ungenau[HZ00, 6.1 S.168]. Abbildung 4.5 zeigt ein Beispiel für entartete Konfigurationen.

4.3.2 Bestimmung bei kalibrierter Kamera

Der DLT-Algorithmus ist anfällig für entartete Situationen und instabil, denn er optimiert alle 11 Freiheitsgrade der Projektionsmatrix. Ist man jedoch im Besitz der Kalibrationsmatrix $\boldsymbol{K}$, so sind bereits 5 Freiheitsgrade festgelegt. Es müssen nur noch die *extrinsischen Kalibrationsparameter* Rotation und Translation optimiert werden. Die Rotation besitzt drei Freiheitsgrade (beispielsweise als Euler-Winkel), ebenso wie die Translation. Durch die Reduktion der Freiheitsgrade reicht eine geringere Menge an bekannten Punkten aus, um das Problem zu lösen. Tatsächlich sind bereits 3 Punkte genug, um die Lösungsmenge auf bis zu vier Lösungen zu reduzieren[QL99]. Hat man vier Punkte, so ist die Lösung (im Allgemeinen) eindeutig bestimmt.

Die Aufgabe, zu einer gegebenen Projektion unter bekannten Projektionsparametern eine Rotation und Translation zu finden, die den Beobachtungen entspricht, heißt *Posebestimmung* (engl. *Pose Estimation*).

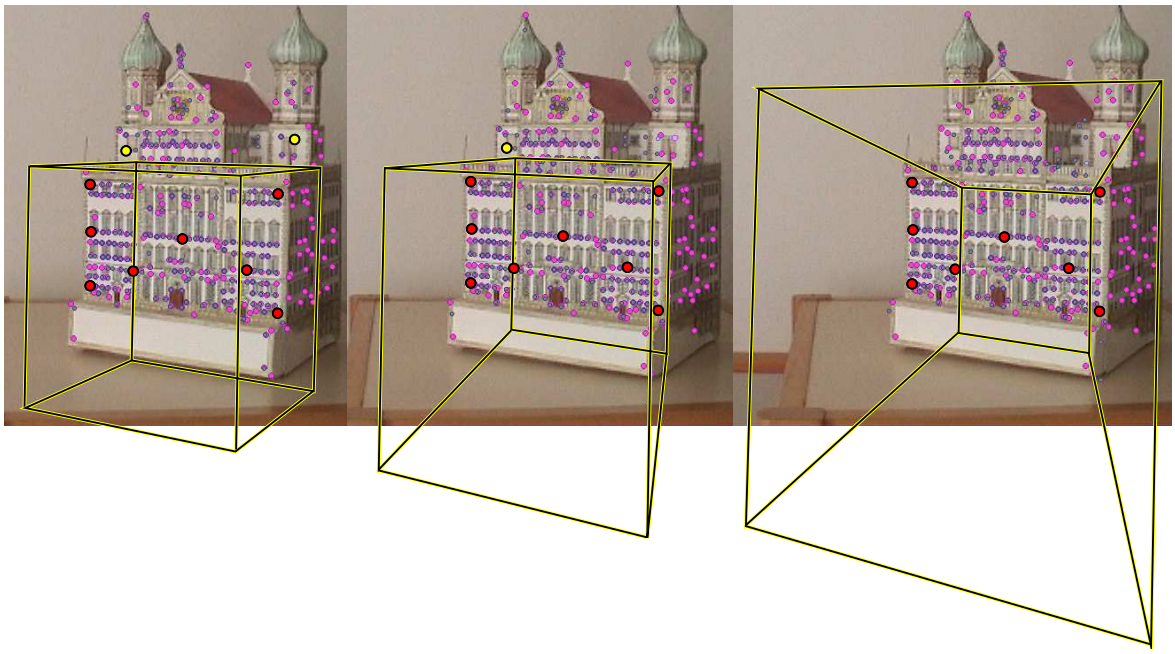


Abbildung 4.5: Beispiel für entartete Situationen bei der Ermittlung der Projektionsmatrix. Anhand von Punkten, deren Lage im Raum bekannt ist, wird eine Projektionsmatrix $\boldsymbol{P}$ ermittelt. Alle Bilder zeigen den Einheitswürfel im Weltkoordinatensystem projiziert durch $\boldsymbol{P}$. Das **linke Bild** zeigt eine korrekt ermittelte Projektion. Sämtliche Punkte liegen auf einer Ebene, bis auf die beiden gelben. Im **mittleren Bild** wurde einer der gelben Punkte, im **rechten Bild** beide gelben Punkte weggelassen. Während die Punkte auf der Ebene weiterhin korrekt projiziert werden, ist die Projektion anderer Punkte scheinbar willkürlich.

Spezialfall: Posebestimmung aus Homographie

Zuerst einmal soll ein Spezialfall des Posebestimmungsproblems behandelt werden, der in der Praxis eine große Rolle spielt. Gegeben seien zwei Bilder, die eine Ebene $\boldsymbol{\pi}$ zeigen. Wie in 4.2.1 gezeigt, lässt sich hier eine Homographie berechnen, sodass $\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{m}}'$ für Punkte $\tilde{\boldsymbol{m}}'$ aus Bild 1 und $\tilde{\boldsymbol{m}}$ aus Bild 2. Durch Wahl eines geeigneten euklidischen Weltkoordinatensystems lässt sich erreichen, dass $\boldsymbol{\pi}$ der XY-Ebene entspricht, also in homogenen Koordinaten $\boldsymbol{\pi} = \{[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{0}, \boldsymbol{w}]^{\boldsymbol{T}} \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w} \in \mathbb{R}, \boldsymbol{w} \neq \boldsymbol{0}\}$.

Seien nun die Projektionsmatrix $\boldsymbol{P}'$ des ersten Bildes bekannt und Position und Rotation $[\boldsymbol{R}|\boldsymbol{t}]$ des zweiten Bildes die zu bestimmenden Parameter. Auch bekannt sei die Kalibrationsmatrix des zweiten Bildes, $\boldsymbol{K}$. Dann gilt:

$$\begin{aligned} \tilde{\boldsymbol{m}} &= \boldsymbol{H}\tilde{\boldsymbol{m}}' \\ \Leftrightarrow \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}][\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{0}, \boldsymbol{w}]^{\boldsymbol{T}} &= \boldsymbol{H}\boldsymbol{P}'[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{0}, \boldsymbol{w}]^{\boldsymbol{T}} \\ \Leftrightarrow [\boldsymbol{R}|\boldsymbol{t}][\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{0}, \boldsymbol{w}]^{\boldsymbol{T}} &= \boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{H}\boldsymbol{P}'[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{0}, \boldsymbol{w}]^{\boldsymbol{T}} \end{aligned}$$

Sei $\boldsymbol{R} = [\boldsymbol{r_1}|\boldsymbol{r_2}|\boldsymbol{r_3}]$ und $\boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{H}\boldsymbol{P}' = [\boldsymbol{a_1}|\boldsymbol{a_2}|\boldsymbol{a_3}|\boldsymbol{a_4}]$. Durch Weglassen der Nullspalten und der homogenen Koordinaten erhalten wir:

$$\boldsymbol{\lambda}[\boldsymbol{r_1}|\boldsymbol{r_2}|\boldsymbol{t}] = [\boldsymbol{a_1}|\boldsymbol{a_2}|\boldsymbol{a_4}]$$

wobei $\boldsymbol{\lambda} \neq \boldsymbol{0} \in \mathbb{R}$ durch den Wegfall der homogenen Koordinaten zustande kommt. Da $\boldsymbol{r_1}$ ein Spaltenvektor einer Rotationsmatrix ist, muss gelten $\|\boldsymbol{r_1}\| = \boldsymbol{1}$. Somit ist $\boldsymbol{\lambda} = \pm\|\boldsymbol{a_1}\|$ und $[\boldsymbol{r_1}|\boldsymbol{r_2}|\boldsymbol{t}] = \pm\frac{\boldsymbol{1}}{\|\boldsymbol{a_1}\|}[\boldsymbol{a_1}|\boldsymbol{a_2}|\boldsymbol{a_4}]$. Die dritte Spalte einer Rotationsmatrix erfüllt $\boldsymbol{r_3} = \boldsymbol{r_1} \times \boldsymbol{r_2}$. Durch Einsetzen erhält man in beiden Fällen $\boldsymbol{r_3} = \frac{\boldsymbol{1}}{\|\boldsymbol{a_1}\|^{\boldsymbol{2}}}\boldsymbol{a_1} \times \boldsymbol{a_2}$.

Welche Lösung ist die richtige? Die beiden Lösungen stammen daher, dass sich aus den Koordinaten $\tilde{\boldsymbol{m}}$ nicht ergibt, ob sich die Punkte *vor* oder *hinter* der Kamera befinden. In projektiven Bildkoordinaten macht das keinen Unterschied: Ein homogener Vektor $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}]^{\boldsymbol{T}}$ repräsentiert den gleichen 1-Dimensionalen Unterraum wie $[-\boldsymbol{x}, -\boldsymbol{y}, -\boldsymbol{w}]^{\boldsymbol{T}}$. Dadurch ist auch nicht definiert, auf welcher Seite der Ebene $\boldsymbol{\pi}$ sich die Kamera befindet.

In der Praxis ist das Problem jedoch leicht lösbar, indem man dafür sorgt, dass sich die Kamera des zweiten Bildes auf derselben Seite der Ebene befindet wie die Kamera des ersten Bildes. Dazu wählt man die Lösung, für die $\mathbf{sign}(\boldsymbol{r_3^T t}) = \mathbf{sign}(\boldsymbol{a_3^T a_4})$.

Robustheit Durch schlechte Lokalisierung der $\tilde{\boldsymbol{m}}$ und $\tilde{\boldsymbol{m}}'$ (z.B. ausgelöst durch Bildrauschen) oder schlecht kalibrierte Kameras kann es zu dem Fall kommen, dass $\|\boldsymbol{a_1}\| \neq \|\boldsymbol{a_2}\|$ und $\boldsymbol{a_1^T a_2} \neq \boldsymbol{0}$. Der oben beschriebene Algorithmus erzeugt dann eine Matrix $\boldsymbol{R}$, die keine Rotationsmatrix mehr ist. Sturm zeigt in [Stu00], wie eine robustere Bestimmung mittels Singulärwertzerlegung möglich ist.

Allgemeine Posebestimmung

Die allgemeine Posenbestimmung befasst sich mit dem Problem, der Projektion von $\boldsymbol{n}$ Punkten mit bekannter Lage eine Pose zuzuordnen, die möglichst gut mit den Messungen übereinstimmt. Hier eine kleine Übersicht. Bei $\boldsymbol{n} = \boldsymbol{3}$ ist es möglich, bis zu vier verschiedene Lösungen zu finden. Dazu ist die Lösung einer Gleichung vierten Grades erforderlich.

Quan und Lan zeigen in [QL99] einen Algorithmus, um die Lösung für 4 und mehr Punkte mit linearen Methoden zu finden.

POSIT [DD92] ist ein iterativer Algorithmus zur Bestimmung der Pose ausgehend von einer skalierten Parallelprojektion.

Rosenhahn verwendet in [Ros03] zur Analyse des Posebestimmungsproblems eine Clifford-Algebra und erweitert die Aufgabe auf die Posebestimmung von Freiformobjekten.

4.3.3 Geometrisches vs. algebraisches Fehlermaß

In den bisher vorgestellten linearen Lösungsverfahren wurde immer ein Fehlerterm $\boldsymbol{d_{alg}(x) = \|Ax\|}$ minimiert. Dieser Fehler nennt sich *algebraischer Fehler*. Es ist nicht ohne weiteres möglich, dem algebraischen Fehler eine einfache geometrische Bedeutung beizumessen. Eine intuitivere Fehlerfunktion ist der geometrische Fehler $\boldsymbol{d^2_{geom}}$. Er ist definiert als die Summe der quadratischen euklidischen Abstände zwischen gemessenen und projizierten Punkten in der Bildebene. Für die Projektionsmatrix ist also $\boldsymbol{d^2_{geom}(P) = \sum_i d^2(\tilde{m}, P\tilde{M})}$ mit $\boldsymbol{d^2([x, y, 1]^T, [x', y', 1]^T) = (x - x')^2 + (y - y')^2}$. Ein Vorteil des geometrischen Fehlers ist, dass das Optimum invariant unter Winkel erhaltenden Operationen ist, solange die Bildpunkte quadratisch sind ($\boldsymbol{\alpha = \beta}$ und $\boldsymbol{s = 0}$ in $\boldsymbol{K}$).

4.3.4 Nichtlineare Minimierung

Der geometrische Fehler kann durch lineare Verfahren in den meisten Fällen nicht minimiert werden. Es ist jedoch möglich, durch nichtlineare Optimierungsverfahren eine sehr gute Annäherung zu erreichen. Voraussetzung ist, dass durch lineare Verfahren oder auf anderem Wege eine Initiallösung in der Nähe des Minimums ermittelt werden konnte.

Hierbei stellt man die zu optimierende Projektionsmatrix durch eine minimale Parametrisierung dar. Im Falle der reinen Posebestimmung mit bekannter Kalibrationsmatrix sind das sechs Parameter (jeweils drei für Rotation und Translation). Sei $\boldsymbol{q \in \mathbb{R}^6}$ der Parametervektor. Eine Funktion $\boldsymbol{g}$ bildet diesen auf eine $\mathbb{R}^{3\times4}$-Matrix der Form $[\boldsymbol{R}|\boldsymbol{t}]$ ab: $\boldsymbol{P = K[R|t] = K\, g(q)}$. Es muss also ein Parametervektor gefunden werden, der $\boldsymbol{d^2_{geom}(K\, g(q))}$ minimiert.

Dies ist beispielsweise mit dem *Levenberg-Marquardt-Verfahren* möglich, einem Standardverfahren aus der Optimierung.

Kapitel 5

Werkzeugkasten für Tracking-Systeme

Dieses Kapitel hat zum Ziel, den Leser mit Methoden vertraut zu machen, die in einem Tracking-System zum Einsatz kommen. Ziel ist es, die geometrischen Modelle aus Kapitel 4 auf zwei Bilder anzuwenden, die von einer Kamera geliefert werden. Dazu ist die Extraktion von Merkmalen aus dem Bildsignal notwendig, was in Abschnitt 5.1 geschieht. Diese können Mathematisch als Punkte dargestellt werden. Zuvor muss jedoch eine Verknüpfung der gefundenen Merkmale beider Bilder miteinander stattfinden. Diesem Thema widmet sich Abschnitt 5.2. Wie dabei das Problem überwunden wird, ein mathematisches Modell aus Merkmalspaaren abzuleiten und gleichzeitig Merkmalspaare anhand des mathematischen Modells zu finden, wird im letzten Abschnitt 5.3 erläutert.

5.1 Merkmalsextraktion

Ein Bild wird im Computer durch ein zweidimensionales Feld von diskreten Intensitätswerten dargestellt. Dieses Feld beinhaltet eine enorme Menge an Informationen. Die Merkmalsextraktion versucht, spezifische Eigenschaften des Bildes herauszufinden und durch möglichst wenige kennzeichnende Parameter zu beschreiben.

Zielsetzung der Merkmalsextraktion ist es, Gemeinsamkeiten in Bildern zu finden. Dazu ist es notwendig, in

- *verschiedenen* Bildern
- *unabhängig* voneinander
- *dieselben* Merkmale zu extrahieren.

In der Praxis ist es nicht möglich, exakt dieselben Merkmale in zwei Bildern mit schon geringen Unterschieden wiederzufinden. Allerdings kann mit den folgenden Methoden erreicht werden, dass möglichst viele gefundene Merkmale tatsächlich in beiden Bildern vorkommen.

Welche Bildeigenschaften kommen als Merkmal in Frage? In der Computer Vision wurden schon viele Ansätze erforscht, Merkmale aus Bildern zu extrahieren. Zu den erfolgreichen Versuchen gehören die folgenden:

Punkte sind die einfachste Merkmalsart, da sie nur durch eine $\boldsymbol{x}$- und eine $\boldsymbol{y}$-Koordinate beschrieben sind. In der Praxis werden den Merkmalspunkten Informationen zugefügt, um weitere Unterscheidungskriterien zu ermöglichen. Oft wird von Merkmalspunkten gesprochen, auch wenn eigentlich eine Umgebung des Bildes um den Punkt herum gemeint ist. Punkte sind auch die einfachste geometrische Struktur und deshalb mathematisch leicht verständlich.

Geraden und andere geometrische Formen können ebenso als Merkmale dienen. Besonders in markerbasierten Tracking-Systemen ist die Erkennung von Rechtecken (auch unter projektiver Verzerrung) eine Standardprozedur.

„Blobs“ sind zusammenhängende Flächen mit bestimmten Farbeigenschaften. So kann in Videoüberwachungssystemen gezielt nach hautfarbenen Flächen gesucht werden, um Personen zu erkennen. Auch ob ein Blob länglich ist oder welchen Flächeninhalt er besitzt wird als Merkmal verwendet.

Konturlinien sind ebenfalls wichtige Merkmale, da sich viele Objekte gut über ihre Silhouetten erkennen lassen.

Statistische Merkmale wie Farb- und Helligkeitsverteilung geben Aufschluss über Art und Inhalt des Bildes.

In dieser Arbeit wird vor allem auf Punktmerkmale eingegangen. Sie stellen die Basis für die meisten Trackingalgorithmen dar.

5.1.1 Gewünschte Eigenschaften

Die Merkmalsextraktion bildet die Grundlage für die darauf folgende Korrespondenzenfindung. Dieses Ziel bestimmt, welchen Anforderungen der Extraktionsalgorithmus genügen muss, um in verschiedenen Abbildungen derselben Szene möglichst viele gemeinsame Merkmalspunkte zu finden. Es ist deshalb wichtig, dass die Identifikation von Merkmalen invariant unter Transformationen ist, die in fotografischen Abbildungen typisch sind.

Invarianz unter Helligkeits- und Kontraständerung Durch unterschiedliche Belichtungszeiten oder Beleuchtungsverhältnisse können sich zwei Bilder derselben Szene, selbst wenn sie aus exakt der gleichen Lage fotografiert wurden, in ihrer Helligkeit und ihrem Kontrast unterscheiden. Dies soll, solange die Unterschiede nicht extrem sind (schwarzes Bild), keine Auswirkungen haben.

Invarianz unter Translation Durch Verschiebung der Kamera parallel zur Bildebene verschiebt sich auch die Position eines Merkmalspunkts auf dem Bild. Es ist wünschenswert, dass die Lage eines Merkmals im Bild keinen Einfluss auf seine Erkennung hat.

Invarianz unter Rotation Rotation der Kamera um ihre Hauptachse erzeugt eine Rotation des Bildes um seinen Hauptpunkt. Auch dieser Effekt soll möglichst wenig Einfluss auf die zu erkennenden Merkmale im Bild haben.

Invarianz unter Skalierung Wird ein Gegenstand mit einem Weitwinkelobjektiv fotografiert, so erscheint er im Bild größer als auf einem Teleobjektivbild. Dennoch zeigt er, bis auf Skalierung, dieselben Merkmale. Besonders die gleichmäßige Skalierung beider Koordinatenachsen sollte keine Auswirkungen haben.

Invarianz unter projektiver Transformation Texturierte Flächen bieten zumeist viele Anhaltspunkte für die Extraktion von Merkmalen. Ebenen sind in fotografischen Abbildungen projektiven Transformationen unterworfen. Um eine größtmögliche Wiedererkennungsrate zu erreichen, sollten diese möglichst wenig Einfluss auf die Merkmalsextraktion haben.

Die folgenden Abschnitte stellen Methoden vor, die diese Anforderungen zumindest teilweise erfüllen.

5.1.2 Notation

Zuvor jedoch eine kurze Einführung der verwendeten Notation. Wie eingangs erwähnt, sind Bilder als zweidimensionales Feld von Intensitätswerten gespeichert. Mathematisch stellen wir ein Bild als Funktion $\boldsymbol{I} : \mathbb{Z}^2 \rightarrow \mathbb{R}$ dar. Der Zugriff auf einen Bildpunkt geschieht über den Subskript: $\boldsymbol{I_p}$ ist die Intensität des Bildpunktes mit Koordinaten $\boldsymbol{p} \in \mathbb{Z}^2$. In der Realität verfügt ein Bild nur über endlich viele Bildpunkte. Deshalb wird die Funktion außerhalb des gespeicherten Bereichs durch die Nullfunktion fortgesetzt, $\boldsymbol{I_p}$ ist also für alle $\boldsymbol{p} \in \mathbb{Z}^2$ definiert.

Faltungsoperator Der Operator $\otimes$ definiert eine Faltung[1]:

$$(\boldsymbol{I} \otimes \boldsymbol{J})_{\boldsymbol{x}} = \sum_{\boldsymbol{u}} \boldsymbol{I_u} \boldsymbol{J_{u-x}} \tag{5.1}$$

Die Schreibweise $\boldsymbol{I} \otimes \boldsymbol{M}$, wobei $\boldsymbol{M}$ eine Matrix aus $\mathbb{R}^{(2m+1)\times(2n+1)}$ ist, bezeichnet eine Faltung mit einer Funktion $\boldsymbol{J}$, die um den Koordinatenursprung die Werte der Matrix annimmt:

$$\boldsymbol{J_{(x,y)^T}} = \left\{ \begin{array}{c} \boldsymbol{M_{y+m+1,\ x+n+1}} \\ \boldsymbol{0} \end{array} \middle| \begin{array}{c} \boldsymbol{-n \leq x \leq n, -m \leq y \leq m} \\ \boldsymbol{ansonsten} \end{array} \right.$$

5.1.3 Harris Corner Detector

Die von Harris und Stevens in [HS88] vorgestellte Methode findet Merkmale in Bildern, indem sie nach *Ecken* sucht. Welche Idee hinter diesem Prinzip steckt, zeigt Abbildung 5.1: Wenn man ein Bild durch ein kleines „Fenster“ hindurch betrachtet, also das Bild bis auf einen kleinen Ausschnitt abdeckt, so stellt man fest, dass dieses Fenster oft ein Stück weit verschoben werden kann, ohne dass es zu nennenswerten Änderungen des Inhalts kommt. Dies funktioniert, solange sich nicht eine Kante oder Ecke im Fenster befindet. Ist Letzteres der Fall, führt eine kleine Bewegung des Fensters in jede beliebige Richtung sofort zu einer Änderung des Inhalts.

Diese Änderung lässt sich mathematisch in folgender Weise beschreiben. Wie eingangs erwähnt, besteht das Bild aus diskreten Helligkeitswerten. Es sei $\boldsymbol{I_u}$ die Helligkeit des Bildpunkts an den Koordinaten $\boldsymbol{u}$ ($\boldsymbol{u}$ ist ein Vektor der Dimension 2, beinhaltet also auch die y-Komponente). Wir betrachten eine Verschiebung des Fensters um einen Vektor $\boldsymbol{x}$ (ebenfalls zweidimensional). Nun können wir die Änderung des Fensterinhalts als mathematischen Fehlerterm beschreiben:

$$\boldsymbol{E_x} = \sum_{\boldsymbol{u} \in \boldsymbol{Fenster}} (\boldsymbol{I_{u+x}} - \boldsymbol{I_u})^2$$

Anstatt über alle $\boldsymbol{u} \in \boldsymbol{Fenster}$ zu summieren, können wir auch einfach über *alle* $\boldsymbol{u}$ summieren, wenn wir einen „Fensterterm“ $\boldsymbol{W_{u-z}}$ mit Zentrum $\boldsymbol{z}$ einführen, der an allen Punkten $\boldsymbol{u}$,

[1] Streng mathematisch gesehen handelt es sich nicht um eine Faltung, da bei dieser eine der beiden Funktionen gespiegelt wird. In der Bildverarbeitung hat sich jedoch die Darstellung der Faltung durch eine Faltungsmatrix durchgesetzt, welche *nicht* gespiegelt wird. Um die Sache nicht noch komplizierter zu machen, wird diese Definition verwendet.

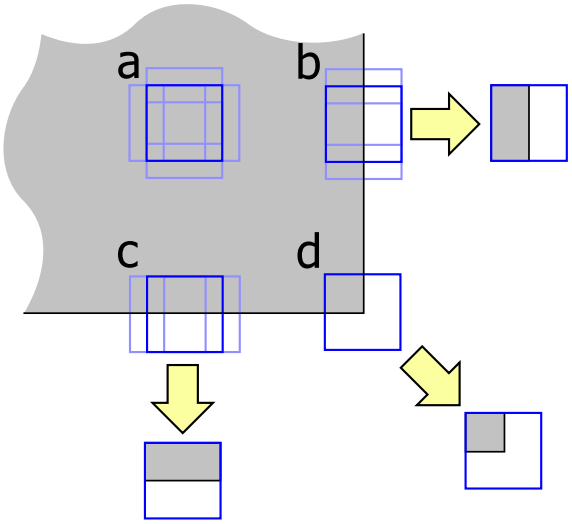


Abbildung 5.1: Das Prinzip der Corner Detection. Das Fenster **a** liegt weder auf einer Kante, noch auf einer Ecke. Es kann in beide Richtungen verschoben werden, ohne dass sich der Inhalt ändert. Die Fenster **b** und **c** liegen auf einer Kante. Sie lassen sich nur noch in eine Richtung bewegen, ohne dass sich ihr Inhalt ändert. Fenster **d** liegt auf einer Ecke und lässt sich nicht bewegen, ohne dass sich der Inhalt ändert.

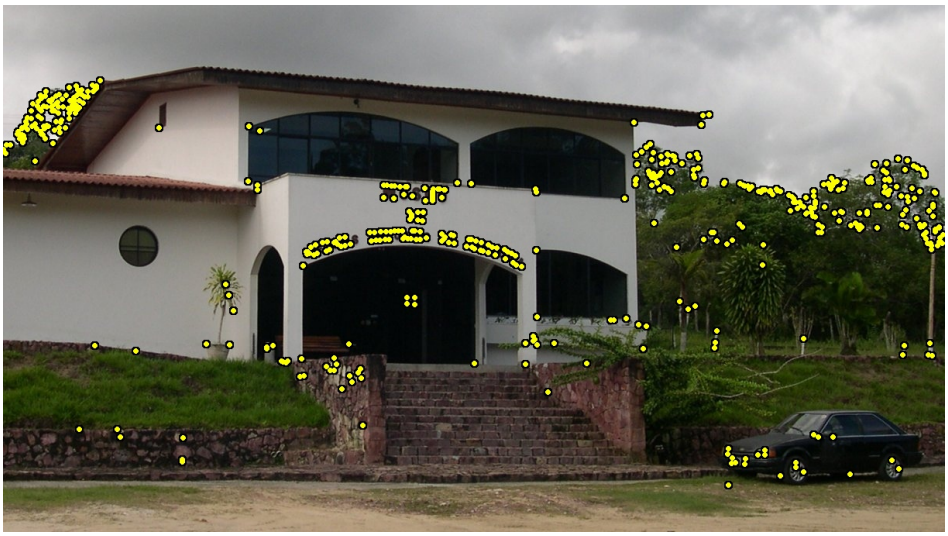


Abbildung 5.2: Es wurden die Harris-Merkmale mit den 500 höchsten Cornerness-Werten aus einem Foto extrahiert. Wie leicht zu erkennen ist, entspricht die Wahl der Merkmale oft nicht dem menschlichen Wichtigkeitsempfinden. In diesem Fall spricht der Detector zu sehr auf kleine Hintergrunddetails an. Teil des Problems ist die festgelegte Größe des Fensters.

an denen das Fenster „offen“ ist, den Wert 1 einnimmt und ansonsten die 0.

$$\boldsymbol{E_{z,x}} = \sum_{\boldsymbol{u}} \boldsymbol{W_{u-z}}(\boldsymbol{I_{u+x}} - \boldsymbol{I_u})^{\mathbf{2}}$$

Der „Trick“ des Harris Corner Detectors ist nun, den Term in der Klammer als Funktion $\boldsymbol{f_u(x)}$ mit Gradient $\boldsymbol{g_u} := \boldsymbol{\nabla f_u}$ linear zu approximieren. Wir definieren eine neue Fehlerfunktion:

$$\boldsymbol{E_z(x)} = \sum_{\boldsymbol{u}} \boldsymbol{W_{u-z}}(\boldsymbol{g_u^T x})^{\mathbf{2}} \tag{5.2}$$

Diese Fehlerfunktion approximiert den Term $\boldsymbol{E_{z,x}}$ für kleine $\boldsymbol{x}$. Was *klein* bedeutet, hängt von der Glattheit der Bilddaten ab. Den Gradienten $\boldsymbol{g_u}$ approximieren wir aus den Nachbarn des Bildpunkts bei $\boldsymbol{u}$:

$$\begin{aligned} \boldsymbol{g_u} &= \boldsymbol{(X_u, Y_u)^T} \\ \boldsymbol{X_u} &= \boldsymbol{I_{u+(1,0)^T} - I_{u+(-1,0)^T}} \\ \boldsymbol{Y_u} &= \boldsymbol{I_{u+(0,1)^T} - I_{u+(0,-1)^T}} \end{aligned}$$

Um die Schreibweise zu vereinfachen, verwenden wir den Faltungsoperator $\otimes$:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{I \otimes (-1, 0, 1)} \\ \boldsymbol{Y} &= \boldsymbol{I \otimes (-1, 0, 1)^T} \end{aligned}$$

In 5.2 lässt sich die Variable $\boldsymbol{x}$ aus der Summe ziehen:

$$\boldsymbol{E_z(x)} = \boldsymbol{x^T} \left(\sum_{\boldsymbol{u}} \boldsymbol{W_{u-z}}[\boldsymbol{g_u g_u^T}] \right) \boldsymbol{x} = \boldsymbol{x^T} \left(\sum_{\boldsymbol{u}} \boldsymbol{W_{u-z}} \begin{bmatrix} \boldsymbol{X_u^2} & \boldsymbol{X_u Y_u} \\ \boldsymbol{X_u Y_u} & \boldsymbol{Y_u^2} \end{bmatrix} \right) \boldsymbol{x}$$

und unter Anwendung von Definition 5.1 die Gleichung in eine einfache Form bringen mit

$$\begin{aligned} \boldsymbol{E_z(x)} &= \boldsymbol{x^T M_z x} \\ \boldsymbol{M_z} &= \begin{bmatrix} \boldsymbol{A_z} & \boldsymbol{C_z} \\ \boldsymbol{C_z} & \boldsymbol{B_z} \end{bmatrix} \\ \boldsymbol{A} &= \boldsymbol{X^2 \otimes W} \\ \boldsymbol{B} &= \boldsymbol{Y^2 \otimes W} \\ \boldsymbol{C} &= \boldsymbol{(XY) \otimes W} \end{aligned}$$

Zwischenergebnis Wir ordnen also jedem Punkt $\boldsymbol{z}$ des Bildes eine Matrix $\boldsymbol{M_z}$ zu. Diese kann mit 5.3 zum Berechnen der Änderung des Fensterinhalts verwendet werden, die auftritt, wenn man das Fenster um einen Vektor $\boldsymbol{x}$ verschiebt. Nun soll aus der Matrix $\boldsymbol{M_z}$ ein sinnvolles Kriterium dafür abgeleitet werden, wann es sich bei dem Punkt bei $\boldsymbol{z}$ um eine „Ecke“ handelt.

Die Matrix $\boldsymbol{M_z}$ heißt *Lokale Strukturmatrix*. Sie ist symmetrisch und positiv semidefinit und hat damit zwei reelle Eigenwerte $\geq \mathbf{0}$. Diese beschreiben die Größe des *Fehleranstiegs* in der Richtung des stärksten bzw. des schwächsten Fehleranstiegs. Sind beide Eigenwerte gleich 0, so befinden wir uns in einer „flachen“ Region wie bei Fenster **a** in Abbildung 5.1. Ist ein Eigenwert größer als $\mathbf{0}$, befindet sich das Fenster über einer Kante wie in den Fällen **b** und **c**. Sind beide Eigenwerte größer als $\mathbf{0}$, ist Fall **d** eingetreten und das Fenster befindet sich über einer Ecke.

Um das Berechnen der Eigenwerte einzusparen, machen wir uns zunutze, dass

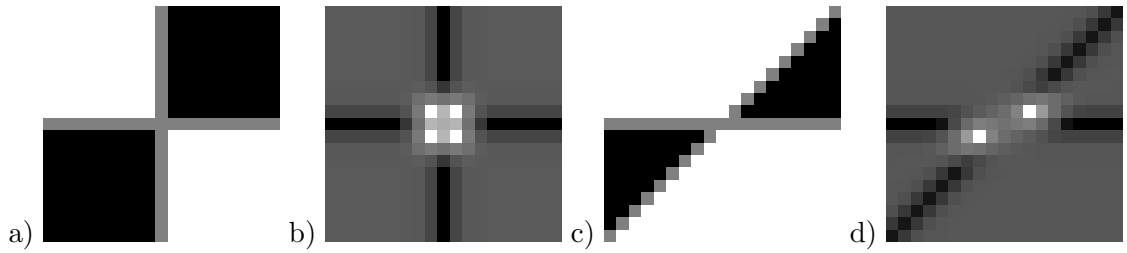


Abbildung 5.3: Bild a) zeigt ein Schachbrettmuster mit Werten 0 (schwarz), 1 (weiß) und 0,5(grau). Bild b) ist die zugehörige Cornerness. Es zeigt 4 Maxima um das Zentrum in symmetrischer Anordnung. Dies ist typisch bei der Verwendung von Gaußfenstern. Bild c) Zeigt das Schachbrettmuster unter Scherung mit Scherparameter 1. Bild d) zeigt die zugehörige Cornerness. Die zwei Maxima in den weißen und nun stumpfen Ecken sind verschwunden.

- Die Determinante von $\boldsymbol{M_z}$ gleich dem *Produkt der Eigenwerte* und
- die Spur von $\boldsymbol{M_z}$ gleich der *Summe der Eigenwerte* ist.

Wir definieren die *Cornerness* („Eckigkeit“) $\boldsymbol{R}$ an der Stelle $\boldsymbol{z}$ als

$$\boldsymbol{R_z = \det(M_z) - k\,\mathrm{tr}(M_z)^2}$$

und somit

$$\boldsymbol{R = AB - C^2 - k(A + B)^2}$$

wobei $\boldsymbol{k}$ eine gewählte Konstante ist (Harris schlägt 0,04 vor). Die Merkmalspunkte des Harris Corner Detector sind dann die *lokalen Maxima von* $\boldsymbol{R}$. Es kann ein Schwellwert verwendet werden, um die Anzahl der Merkmalspunkte einzuschränken.

Bisher haben wir das Fenster immer als rechteckig angenommen. Durch die Einführung der Fensterfunktion $\boldsymbol{W}$ ist es jedoch möglich, andere Formen zu wählen. Tatsächlich hat es sich als vorteilhaft herausgestellt, eine Gaußfunktion zu verwenden, also

$$\boldsymbol{W_x = e^{-\frac{x^T x}{2\sigma^2}}}$$

mit Standardabweichung $\boldsymbol{\sigma}$. Dies hat zwei Gründe:

1. *Rotationsinvarianz.* Die Gaußfunktion hat die Eigenschaft, vollkommen rotationssymmetrisch zu sein. Dadurch gewinnt auch die Merkmalsextraktion an Rotationstoleranz.
2. *Stetigkeit.* Eine Rechtecksfunktion gewichtet alle Punkte im Rechteck gleich, einschließlich der Randpunkte. Dies kann bei kleinen Verschiebungen des Fensters zu großen Änderungen der Fehlerfunktion führen. Durch die Gewichtungsabnahme zum Rand des Gaußfensters wird dieser Effekt unterdrückt.

Diskussion der Invarianzeigenschaften

Es soll nun untersucht werden, inwieweit der Harris Corner Detector den in 5.1.1 aufgestellten Anforderungen genügt.

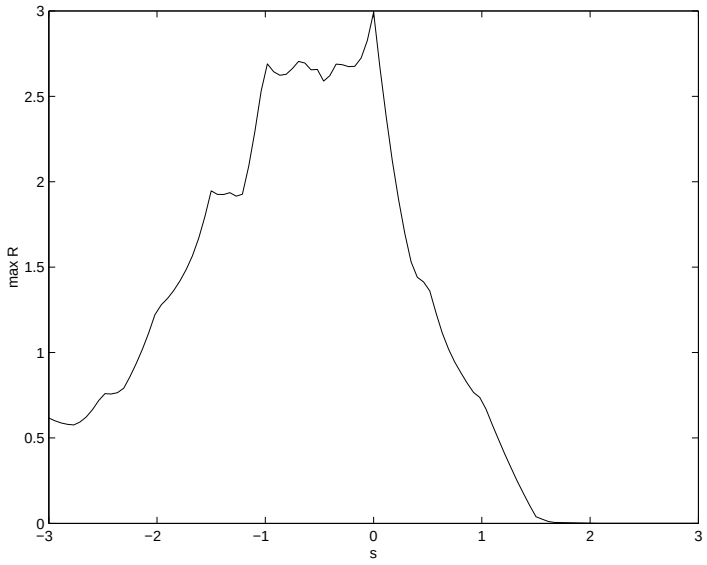


Abbildung 5.4: Die Cornernessfunktion der Ecke eines Parallelogramms mit Scherparameter $\boldsymbol{s}$. Bei spitzen Winkeln ($\boldsymbol{s} < \mathbf{0}$) ist die Cornerness deutlich höher als bei stumpfen Winkeln ($\boldsymbol{s} > \mathbf{0}$). Die größte Cornerness hat erwartungsgemäß der 90°-Winkel bei s=0.

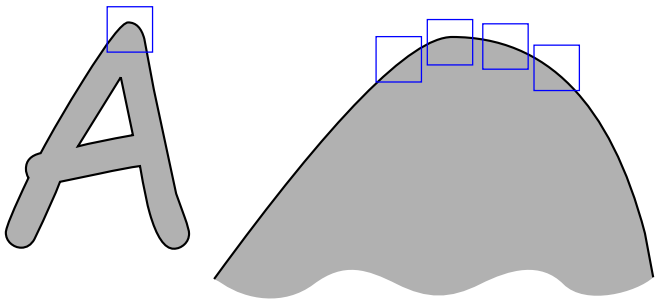


Abbildung 5.5: Durch die willkürlich festgelegte Größe des Suchfensters ist der Harris Corner Detector nicht skalierungsinvariant. Die abgerundete Spitze des Buchstaben „A“ wird in der linken Skalierung als Punkt erkannt. In der rechten Skalierung ist die Krümmung so gering, dass die Kurve nicht mehr als Ecke erkannt wird.

Invarianz unter Helligkeits- und Kontraständerung Wenn wird die Änderung als lineare Funktion $I' = \alpha I + \beta$ modellieren, ist leicht zu sehen dass $g'_u = \alpha g_u$ und damit $M'_z = \alpha^2 M_z$. Demnach ist $R' = \alpha^4 AB - \alpha^4 C^2 - \alpha^4 k(A+B)^2 = \alpha^4 R$. Da sich die neue Cornerness nur um einen konstanten Faktor von der ursprünglichen unterscheidet, treten die Maxima an denselben Stellen auf. In der Theorie ist der Harris Corner Detector also vollständig invariant unter linearer Helligkeits- und Kontraständerung. In der Praxis steigt mit sinkender Bildhelligkeit jedoch zumeist der Rauschanteil, was sich stark negativ auf die erkannten Ecken auswirken kann.

Invarianz unter Translation Da sich die lokale Strukturmatrix nur aus Punkten der nahen Umgebung errechnet, ist eine Invarianz unter Translation gegeben.

Invarianz unter Rotation Bei einer rotationssymmetrischen Fensterfunktion wirkt sich eine Rotation des Bildes nur auf die *Eigenvektoren*, nicht die *Eigenwerte* der lokalen Strukturmatrix aus, welche die Grundlage für die Cornerness bilden.

Invarianz unter Skalierung Der Harris Corner Detector ist nicht Skalierungsinvariant. Die gewählte Größe des Fensters bestimmt, welche Details noch als Merkmal erkannt werden können (s. Abb. 5.5).

Invarianz unter projektiver Transformation Projektive Transformationen sind nicht Winkel erhaltend. Spitze Winkel können zu stumpfen Winkeln werden. Abbildung 5.4 zeigt die Winkelabhängigkeit des Harris Corner Detectors. In der Praxis hat sich das Verhalten als brauchbar erwiesen, solange die Winkeländerungen nicht zu groß sind.

5.1.4 Weitere Methoden der Merkmalsextraktion

Ein weiteres Verfahren, das sich ebenfalls großer Beliebtheit erfreut, ist der dem Harris Corner Detector sehr ähnliche Kanade-Lucas-Tomasi-Algorithmus (KLT). Im Unterschied zu Ersterem werden die Eigenwerte der lokalen Strukturmatrix explizit berechnet. Die Merkmalspunkte sind dann die Punkte, an denen der jeweils *kleinere Eigenwert* ein lokales Maximum einnimmt. Auch hier wird ein Schwellwert eingesetzt. [ST94]

Lowe stellt mit *SIFT* (Scale Invariant Feature Transform, [Low03]) einen Algorithmus zur Extraktion skalierungsinvarianter Merkmale vor. Er erreicht dies, indem er die Merkmale nicht im 2-dimensionalen Bildraum, sondern im 3-dimensionalen Skalenraum des Bildes ansiedelt. Dessen dritte Komponente ist eine Größeninformation. Dazu ist es notwendig, das Eingabebild in einer so genannten Pyramide von verschiedenen Größen- und Glättungsstufen nach Extrema einer Funktion zu durchsuchen. Lowe verwendet als Funktion die Differenz zweier benachbarter Glättungsstufen ("Difference of Gaussian"). Der SIFT-Algorithmus verknüpft einen Merkmalspunkt („Keypoint") mit weiteren Kennwerten, um eine effiziente Korrespondenzensuche zu ermöglichen.

5.2 Matching der Merkmalspunkte zweier Bilder

Dieser Abschnitt beschäftigt sich mit der Frage, wie aus zwei unabhängig voneinander gesammelten Merkmalsmengen möglichst viele „zusammen passende" Merkmale ermittelt werden können, wobei dieser Begriff zuerst einmal definiert werden muss. Darauf folgend soll in 5.2.2 das Problem des „Merkmalspaare Findens" etwas formalisiert werden. Wie man nun tatsächlich ein „gutes" Matchingergebnis erzielt, behandeln die nachfolgenden Abschnitte.

5.2.1 Kriterien für korrespondierende Merkmale

Die Suche nach korrespondierenden Bildmerkmalen ist ein Vorgang, den das Wahrnehmungszentrum des Menschen zu fast jedem Zeitpunkt durchführt. Das beidäugige Sehen ermöglicht uns eine exakte Tiefenwahrnehmung. Dazu müssen die Nervensignale des linken und des rechten Auges durch das Gehirn zu einem dreidimensionalen Synthesebild verarbeitet werden. Dieser Vorgang setzt eine leistungsfähige Korrespondenzensuche voraus. Die folgenden Gesetze entspringen daher Beobachtungen der Wahrnehmungspsychologie[Ull79].

Das Prinzip der Exklusivität Wenn ein Merkmal des einen Bildes mit einem Merkmal des anderen Bildes korrespondiert, so ist diese Korrespondenz *exklusiv*. Kein anderes Merkmal ist mit einem dieser beiden Merkmale verknüpft.[2]

Das Prinzip der Ähnlichkeit Zusammengehörige Merkmale weisen in einer kleinen Umgebung eine hohe Ähnlichkeit auf. In dieser Form gilt das Gesetz allerdings nur bei Bildern, deren Aufnahmezentren nahe beieinander liegen und wenn keine Rotation in der Bildebene vorliegt. Im Fall des beidäugigen Sehens ist diese Bedingung erfüllt. Sollen Bilder aus stark unterschiedlichen Blickwinkeln verglichen werden, muss zuerst eine geometrische Transformation des Merkmals durchgeführt werden, um die Ähnlichkeit festzustellen.

Das Prinzip der Nachbarschaft Betrachtet man ein Merkmal nicht isoliert, sondern im Zusammenhang mit benachbarten Merkmalen, so bleibt deren *Anordnung* zueinander weitgehend identisch. Merkmale, die diesem Prinzip widersprechen, weisen auf Tiefenunterschiede und somit Objektgrenzen hin und werden in der menschlichen Wahrnehmung besonders hervorgehoben.

5.2.2 Die Paarungsmatrix

Es liegt folgende Situation vor: Gegeben zwei Bilder $\boldsymbol{I}$ und $\boldsymbol{I'}$. $\boldsymbol{I}$ besitzt eine endliche Menge von Merkmalen, $\boldsymbol{f_1 \dots f_m}$. Zu $\boldsymbol{I'}$ gehören die Merkmale $\boldsymbol{f'_1 \dots f'_n}$. Wenn es möglich ist, die „Zusammengehörigkeit“ zweier Merkmale $\boldsymbol{f_i}$ und $\boldsymbol{f'_j}$ als Zahl $\boldsymbol{c_{ij}}$ auszudrücken, können wir eine Matrix $\boldsymbol{C = (c)_{ij}}$ schreiben:

$$\begin{array}{c} \\ \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \vdots \\ \boldsymbol{f_m} \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{f'_1} & \boldsymbol{f'_2} & \dots & \boldsymbol{f'_n} \end{array} \\ \begin{pmatrix} \boldsymbol{c_{11}} & \boldsymbol{c_{12}} & \dots & \boldsymbol{c_{1n}} \\ \boldsymbol{c_{21}} & \boldsymbol{c_{22}} & \dots & \boldsymbol{c_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{c_{m1}} & \boldsymbol{c_{m2}} & \dots & \boldsymbol{c_{mn}} \end{pmatrix} \end{array}$$

Ein *Paar* sei in diesem Zusammenhang ein Tupel $\boldsymbol{(i, j)}$ mit $\boldsymbol{i \in \{1 \dots m\}}$ und $\boldsymbol{j \in \{1 \dots n\}}$ und lässt sich als Markierung des entsprechenden Eintrags in $\boldsymbol{C}$ darstellen. Um das Prinzip der Exklusivität zu wahren, darf es in jeder Zeile und jeder Spalte maximal einen markierten Eintrag geben.

[2] Das Gesetz der Exklusivität gilt eigentlich nur bei gleichzeitig aufgenommenen Bildern. In Bewegungssequenzen nimmt das Auge sehr wohl auch sich spaltende oder vereinende Merkmale wahr. Ullman modelliert dies, aber führt einen *Penalty*-Wert ein, um unnötige Mehrfachzuordnungen zu vermeiden.[Ull79]

Rückführung auf gewichtete Paarung im bipartiten Graph Es sei $(\boldsymbol{V}, \boldsymbol{E})$ ein ungerichteter, bipartiter Graph ohne Mehrfachkanten. Die Bipartition sei $\boldsymbol{V} = \boldsymbol{A} \dot{\cup} \boldsymbol{B}$ und jede Kante $\boldsymbol{e}$ habe die Form $(\boldsymbol{a}, \boldsymbol{b})$ mit $\boldsymbol{a} \in \boldsymbol{A}$ und $\boldsymbol{b} \in \boldsymbol{B}$. Eine Paarung (oder Matching) ist eine Teilmenge $\boldsymbol{M} \subset \boldsymbol{E}$, deren Kanten disjunkt sind: $(\boldsymbol{a}, \boldsymbol{b}) \neq (\boldsymbol{a'}, \boldsymbol{b'}) \Rightarrow \boldsymbol{a} \neq \boldsymbol{a'} \wedge \boldsymbol{b'} \neq \boldsymbol{b'}$. Die Aufgabe, eine „möglichst gute“ Paarung zu finden, nennt sich *Paarungsproblem.* [3] „Möglichst gut“ bezieht sich meistens auf eine Kantenbeschriftung $\boldsymbol{\lambda} : \boldsymbol{E} \rightarrow \mathbb{R}$, die jeder Kante einen Kostenwert (oder Nutzenwert) zuordnet.

Die Aufgabe, Korrespondenzen in zwei Bildern zu finden lässt sich somit reduzieren auf die Aufgabe, eine Paarung in einem bipartiten gewichteten Graph zu finden. Da jedes Merkmal des einen Bildes als potentieller „Partner“ jedes Merkmals des anderen Bildes in Frage kommt, ist der Graph *vollständig bipartit.* Das heißt, zwischen *allen* Knoten $\boldsymbol{a}$ aus $\boldsymbol{A}$ und $\boldsymbol{b}$ aus $\boldsymbol{B}$ existiert eine Kante $(\boldsymbol{a}, \boldsymbol{b})$.

Wenn wir die Knoten in den Partitionen durchnummerieren und mit $\boldsymbol{a_1 \ldots a_m}$ sowie $\boldsymbol{b_1 \ldots b_n}$ bezeichnen, können wir die Darstellung der Paarungsmatrix verwenden: $\boldsymbol{C} = (\boldsymbol{c})_{\boldsymbol{ij}} = \boldsymbol{\lambda}(\boldsymbol{a_i}, \boldsymbol{b_j})$

Matches mit lokaler Optimalität Lokale Optimalität ist eine einfache Definition von „möglichst gut.“ Sie verlangt, dass eine Kante $\boldsymbol{e} = (\boldsymbol{a_i}, \boldsymbol{b_j})$ nur dann in der Paarung ist, wenn:

- $\boldsymbol{\lambda}(\boldsymbol{e})$ minimal(maximal) unter allen $\boldsymbol{\lambda}(\boldsymbol{a_i}, \boldsymbol{b})$ mit $\boldsymbol{b} \in \boldsymbol{B}$ und
- $\boldsymbol{\lambda}(\boldsymbol{e})$ minimal(maximal) unter allen $\boldsymbol{\lambda}(\boldsymbol{a}, \boldsymbol{b_i})$ mit $\boldsymbol{a} \in \boldsymbol{A}$.

Auf die Paarungsmatrix bezogen bedeutet dies, dass der Eintrag $\boldsymbol{c_{ij}}$ minimal(maximal) sowohl in der Zeile $\boldsymbol{i}$, also auch in der Spalte $\boldsymbol{j}$ ist.

Optimales Matching, Kuhn-Munkres-Algorithmus Eine Paarung heißt *perfekt*, wenn ihre Kanten jeden Knoten aus $\boldsymbol{V}$ einschließen. Dies ist natürlich nur möglich, wenn beide Partitionen gleich viele Knoten enthalten. Für allgemeine Partitionen $\boldsymbol{A}$ und $\boldsymbol{B}$ ist die maximale Kardinalität einer Paarung $|\boldsymbol{M_{MAX}}| = \boldsymbol{min}(|\boldsymbol{A}|, |\boldsymbol{B}|)$.

Manchmal sind die Kantenbeschriftungen $\boldsymbol{\lambda}$ von einer Form, in welcher der Summe der Kantenbeschriftungen einer Paarung eine sinnvolle Bedeutung zukommt. Es könnte sich beispielsweise um die Abstände zwischen Merkmalen handeln. In diesem Fall kann es günstig sein, diese Summe zu optimieren und nicht die einzelnen Summanden. Eine Paarung maximaler Kardinalität, die die Summe der Kantengewichte optimiert, nennt sich *optimale Paarung.*

Der *Kuhn-Munkres-Algorithmus*, auch bekannt als *Ungarischer Algorithmus*, findet eine solche optimale Paarung in polynomieller Zeit ($\boldsymbol{O(N^3)}$, $\boldsymbol{N} = |\boldsymbol{A}| = |\boldsymbol{B}|$). Ullman gibt in [Ull79, 3.4 S.91ff] eine Lösung als Lineares Programm an, welches im Falle einer eindeutigen Lösung diskrete Werte annimmt und führt Argumente an, warum diese Modellierung dem Verfahren im menschlichen Gehirn entsprechen könnte.

In der Praxis der Merkmalspaarung hat sich die aufwändige Berechnung einer optimalen Lösung allerdings oft als unnötig herausgestellt, da sich durch die Wahl der richtigen Fehlerfunktion zwischen zwei Merkmalen erreichen lässt, dass die korrekten Matches genau diejenigen mit lokaler Optimalität sind.

[3] In vollkommener Anerkennung der Schlüpfrigkeit dieser Übersetzung des englischen Begriffs *Matching* gibt es auch noch die Bezeichnung *Heiratsproblem.* Speziell meint man hier das Problem, möglichst viele heiratswillige Paare in einer Gruppe von Männern und Frauen zu finden.

5.2.3 Ähnlichkeitsmaße

In Abschnitt 5.2.1 wurden Kriterien aufgestellt, denen korrespondierende Merkmale zweier Bilder genügen sollten. Durch die Rückführung des Matchingproblems auf ein Problem aus der Graphentheorie sind Werkzeuge hinzugekommen, die die Einhaltung des Exklusivitätsprinzips garantieren. Gleichzeitig kann die Einhaltung des Ähnlichkeitsprinzips gewährleistet werden, wenn die Ähnlichkeit zwischen zwei Merkmalen als Kostenfunktion $\boldsymbol{c_{ij} = c(f_i, f'_j)}$ ausgedrückt werden kann. Es sollen nun in Frage kommende Ähnlichkeitsmaße vorgestellt werden.

Summe quadrierter/absoluter Differenzen (SSD,SAD)

Es seien Punktmerkmale $\boldsymbol{f}$ in Bild $\boldsymbol{I}$ an der Position $\boldsymbol{p}$ (zweidimensional), $\boldsymbol{f'}$ in Bild $\boldsymbol{I'}$ mit Position $\boldsymbol{p'}$ gegeben. Eine sehr einfache Möglichkeit, die Merkmale miteinander zu vergleichen, ist das Differenzbild.

Das Differenzbild $\boldsymbol{\Delta}$ ist definiert über:

$$\boldsymbol{\Delta_x = I_{x+p} - I'_{x+p'}}$$

Die gesamte Bildfläche zu vergleichen ist jedoch wenig sinnvoll. Vielmehr empfiehlt es sich, einen kleinen Ausschnitt zu betrachten, der über eine Gewichtungsfunktion $\boldsymbol{W}$ eingeführt wird. Wie in Abschnitt 5.1.3 kann (und sollte) $\boldsymbol{W}$ auch die Form einer zweidimensionalen Gaußfunktion haben.

Die *Summe der absoluten Differenzen* (SAD) ist definiert als:

$$\boldsymbol{d_{SAD} = \sum_x W_x |\Delta_x| = \sum_x W_x |I_{x+p} - I'_{x+p'}|}$$

Der Vorteil der SSD-Funktion ist ihre Einfachheit und effiziente Implementierung. Nachteilig ist, dass kleine Differenzen, wie sie durch Bildrauschen entstehen können, genauso stark gewichtet werden wie große Fehler durch tatsächlich verschiedene Merkmale. Dieses Manko behebt die *Summe der quadrierten Differenzen* (SSD, von engl. Sum of Squared Differences):

$$\boldsymbol{d_{SSD} = \sum_x W_x \Delta_x^2 = \sum_x W_x (I_{x+p} - I'_{x+p'})^2}$$

Beiden Funktionen ist gemein, dass sie auf lineare Helligkeits- und Kontrastunterschiede vom Typ $\boldsymbol{I' = \alpha I + \beta}$ reagieren.

Kreuzkorrelation (NCC)

Aus der Wahrscheinlichkeitstheorie stammt der Begriff der Kovarianz:

$$\boldsymbol{\mathbf{Cov}[X, Y] = E[(X - \mu_X)(Y - \mu_Y)]}$$

wobei $\boldsymbol{X}$ und $\boldsymbol{Y}$ Zufallsvariablen und $\boldsymbol{\mu_X}$ bzw. $\boldsymbol{\mu_Y}$ ihre jeweiligen Erwartungswerte sind. $\boldsymbol{E[\,\cdot\,]}$ bezeichnet den Erwartungswert einer Funktion. Die Kovarianz lässt sich auf Werte zwischen -1 und 1 normieren, indem man sie durch die Standardabweichungen der beiden Zufallsvariablen teilt. Die enstehende Funktion nennt man Korrelation oder auch *Pearson-Koeffizient*:

$$\boldsymbol{\mathbf{Corr}[X, Y] = \frac{\mathbf{Cov}[X, Y]}{\sigma_X \sigma_Y}}$$

Die Korrelation ist invariant unter positiver Multiplikation und Addition:

$$\mathbf{Corr}[\boldsymbol{X}, \boldsymbol{Y}] = \mathbf{Corr}[\boldsymbol{\alpha X} + \boldsymbol{\beta}, \boldsymbol{\gamma Y} + \boldsymbol{\delta}]$$

für $\boldsymbol{\alpha} > \mathbf{0}, \boldsymbol{\gamma} > \mathbf{0}$. Besteht zwischen $\boldsymbol{X}$ und $\boldsymbol{Y}$ ein linearer Zusammenhang, $\boldsymbol{Y} = \boldsymbol{aX} + \boldsymbol{b}$ mit $\boldsymbol{a} > \mathbf{0}$, so ist $\mathbf{Corr}[\boldsymbol{X}, \boldsymbol{Y}] = \mathbf{1}$. Besteht ein *negativer* linearer Zusammenhang, $\boldsymbol{Y} = -\boldsymbol{aX} + \boldsymbol{b}$, dann ist $\mathbf{Corr}[\boldsymbol{X}, \boldsymbol{Y}] = -\mathbf{1}$. Die Korrelation kennt auch „Mischungen“ dieser beiden Extrema. Der Wert $\mathbf{0}$ besagt, dass kein *linearer* Zusammenhang besteht, die Zufallsvariablen sind *unkorreliert*. [4]

Die Korrelation lässt sich auf den Vergleich von Merkmalspunkten übertragen, indem wir die Bildintensitäten als (durch die Fensterfunktion gewichtete) Proben von Zufallsvariablen ansehen. Der entsprechende Operator nennt sich *normalisierte Kreuzkorrelation* (Normalized Cross Correlation, NCC).

$$\mathbf{NCC} = \frac{\mathbf{1}}{\boldsymbol{\sigma\sigma'}} \sum_{\boldsymbol{x}} \boldsymbol{W_x}(\boldsymbol{I_{x+p}} - \boldsymbol{\mu})(\boldsymbol{I'_{x+p'}} - \boldsymbol{\mu'})$$

Dabei ist

$$\begin{aligned} \boldsymbol{\mu} &= \sum_{\boldsymbol{x}} \boldsymbol{W_x I_{x+p}} \\ \boldsymbol{\sigma} &= \sqrt{\sum_{\boldsymbol{x}} \boldsymbol{W_x}(\boldsymbol{I_{x+p}} - \boldsymbol{\mu})^{\mathbf{2}}} \end{aligned}$$

und analog für $\boldsymbol{\mu'}$ und $\boldsymbol{\sigma'}$. Die Fensterfunktion $\boldsymbol{W}$ wird als normiert angenommen, d.h. $\sum_{\boldsymbol{x}} \boldsymbol{W_x} = \mathbf{1}$.

Einige Beispielwerte für die NCC-Funktion finden sich in Abbildung 5.6.

Rotationsinvariante Kreuzkorrelation (NCC-R)

Durch den Pixel-zu-Pixel-Vergleich ist die NCC-Funktion stark abhängig von einer exakten Deckungsgleichheit beider Bilder. Wenn man davon ausgeht, dass die Positionen des Merkmalszentrums übereinstimmen, bleibt als wesentlicher Störfaktor noch eine Rotationskomponente. Diese lässt sich verhältnismäßig leicht ausgleichen, indem den Merkmalen schon bei der Extraktion eine Orientierung zugewiesen wird. Es bietet sich hier der Gradient an. Um die Abhängigkeit von der exakten Lokalisierung des Merkmals etwas zu reduzieren, sollte dieser über das gesamte Suchfenster gemittelt werden. Ravena *et al.* [RDLW95] verwenden hierzu einen so genannten steuerbaren Filter. [FA91]

Der Gradient ist nicht die einzige Funktion, die verwendet werden kann, um einem Merkmal eine Orientierung zuzuweisen. In [SF95] schlagen Simoncelli und Farid weitere steuerbare Filter vor.

Es seien $\boldsymbol{\theta}$ und $\boldsymbol{\theta'}$ die ermittelten Orientierungen zweier zu vergleichender Merkmale. Nun müssen diese Merkmale zuvor um Winkel $\boldsymbol{\varphi}$ und $\boldsymbol{\varphi'}$ gedreht werden, sodass $\boldsymbol{\theta} + \boldsymbol{\varphi} = \boldsymbol{\theta'} + \boldsymbol{\varphi'}$ **mod** $\boldsymbol{2\pi}$. In der Praxis geschieht dies meist durch Wahl $\boldsymbol{\varphi} = \mathbf{0}$ und $\boldsymbol{\varphi'} = \boldsymbol{\theta} - \boldsymbol{\theta'}$, wodurch allerdings die Gleichbehandlung der Merkmale verloren geht, da nur in einem der beiden Merkmale durch die Rotation ein Interpolationsfehler entsteht. Dies stellt jedoch nur selten ein Problem dar, denn die Alternative, beide Merkmale zu rotieren, erzeugt Fehler in beiden Bildern.

[4] Dies bedeutet jedoch nicht, dass sie notwendigerweise unabhängig sind. So ist beispielsweise $\mathbf{Corr}[\boldsymbol{X}, \boldsymbol{X^2}] = \mathbf{0}$.

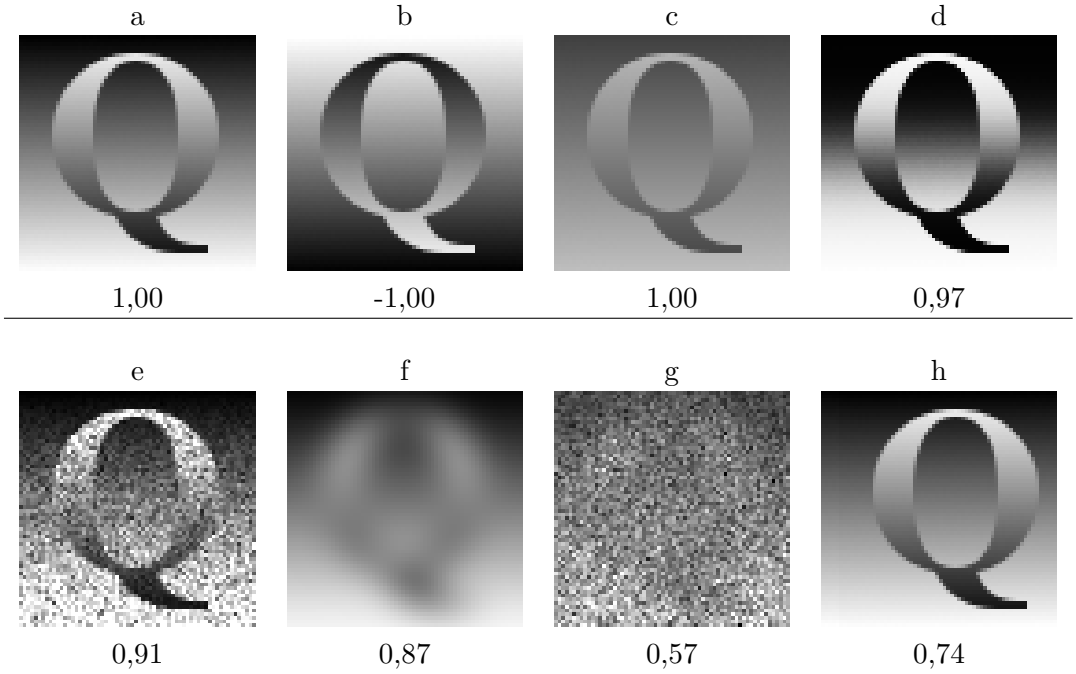


Abbildung 5.6: Beispiel für normalisierte Kreuzkorrelation. Das ursprüngliche Bild **a** eines Buchstaben „Q“ auf einem linearen Helligkeitsverlauf wurde transformiert und der NCC-Wert mit dem neuen Bild berechnet. Die Korrelation von **a** mit **a** ist 1. **b** ist das Inversbild von **a**. **c** entsteht durch Kontrastreduktion. **d** entsteht durch eine nichtlineare Transformation der Helligkeitswerte. **e** entsteht durch Addition von gleich verteiltem Rauschen der Amplitude 0,25. **f** entsteht durch Gaußsches Weichzeichnen mit Standardabweichung 5. **g** wurde weich gezeichnet, kontrastreduziert und verrauscht. **h** entsteht durch Verschiebung um 6 Pixel nach rechts und demonstriert mit einem schwachen NCC-Wert, wie stark die Korrelation von einer exakten Lokalisierung abhängt.

Hauptkomponentenanalyse

Einen anderen Weg geht die so genannte *Hauptkomponentenanalyse*. Sie ermöglicht es, einen Satz von Filtern zu generieren, die speziell auf die Erkennung eines bestimmten Merkmals ausgerichtet sind. Diese Filter werden *Eigenbilder* genannt.

Das Verfahren beruht auf der Beobachtung, dass sich Bilder als Linearkombination anderer Bilder approximieren lassen. Insbesondere lässt sich ein Merkmalspunkt daraufhin untersuchen, wie gut er sich durch eine Reihe von Ansichten eines bekannten Merkmals approximieren lässt. Je mehr solcher Ansichten existieren, desto besser wird sich das Merkmal approximieren lassen, vor allem wenn es sich um ein korrespondierendes Merkmal handelt. Seien also $\boldsymbol{I_1, \ldots, I_k}$ verschiedene Ansichten eines bekannten Merkmals. Üblicherweise handelt es sich um quadratische Bildausschnitte um das Merkmalszentrum. Wir schreiben die Intensitätswerte in einem solchen Fenster als Spaltenvektor $\boldsymbol{i}$, wir haben also $\boldsymbol{i_1, \ldots i_k}$. Ebenso wird das zu approximierende Merkmal $\boldsymbol{J}$ als Spaltenvektor $\boldsymbol{j}$ geschrieben. Es können dann Gewichte $\boldsymbol{w_1, \ldots w_k}$ (geschrieben als Vektor $\boldsymbol{w}$) ermittelt werden, sodass $\| \ [\boldsymbol{i_1}|\cdots|\boldsymbol{i_k}] \ \boldsymbol{w} \ - \boldsymbol{j} \ \|$ minimal ist. Diese Gewichte sind beispielsweise über eine Singulärwertzerlegung zu ermitteln.

Wir bezeichnen die Matrix $[\boldsymbol{i_1}|\cdots|\boldsymbol{i_k}]$ mit den Bildern als Spaltenvektoren als $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n \times k}}$. Mit steigender Anzahl der Merkmalsansichten steigt auch die Anzahl der Spalten von $\boldsymbol{A}$ und somit der Aufwand für das Ermitteln der Gewichte. Jedoch kann statt der Ansichten $\boldsymbol{i_1, \ldots, i_k}$ eine orthogonale Basis gefunden werden, die denselben Raum aufspannt, also $\boldsymbol{span(i_1, \ldots, i_k) = span(u_1, \ldots, u_k)}$. Eine solche Basis ist durch die Singulärwertzerlegung $\boldsymbol{A = U\Sigma V^T}$ gegeben, mit $[\boldsymbol{u_1}|\cdots|\boldsymbol{u_k}] = \boldsymbol{U}$. Die Singulärwertzerlegung ordnet die Singulärwerte in fallender Reihenfolge, somit ist $\boldsymbol{u_1}$ der „wichtigste“ Basisvektor.

Nun können wir neue Gewichte $\boldsymbol{\overline{w} = U^T j}$ bestimmen, sodass $\| \ \boldsymbol{U\overline{w} - j} \ \|$ minimal ist. Dies lässt sich beweisen durch die Normalgleichung: $\boldsymbol{U^T U\overline{w} = U^T UU^T j = U^T j}$. Da jedoch zu den Basisvektoren mit höherem Index kleinere Singulärwerte gehören, können diese unter geringem Genauigkeitsverlust weggelassen werden. Es kann mit einem relativ kleinen Satz dieser Eigenbilder eine gute Approximation eines Merkmals erreicht werden, wie Abbildung 5.7 zeigt. Als Ähnlichkeitsmaß dient dann die Summe der quadrierten Differenzen zwischen Merkmal im Bild und seiner Approximation durch die Eigenbilder.

Voraussetzung für eine gute Erkennung ist eine große Anzahl von Ansichten. Diese müssen jedoch nicht zur Gänze aus tatsächlichen Abbildungen stammen, sondern können durch geometrische Transformation einer kleinen Anzahl von Bildern erzeugt werden.[LPF04]

Weitere Ähnlichkeitsmaße

In gleicher Weise wie bei NCC-R eine dem Merkmal zugeordnete Orientierungsinformation verwendet wird, können auch andere Parameter zu verbesserten Ähnlichkeitsmaßen führen. Lowe ordnet Merkmalen in SIFT [Low03] nicht nur eine Orientierungs-, sondern auch eine Größeninformation zu. Shi und Tomasi gehen noch einen Schritt weiter und beschreiben in [ST94] eine Iteration, um die Korrelation unter beliebiger affiner Transformation (also einschließlich Scherung) zu maximieren.

5.2.4 Geometrischer Fehler

Wir wollen nun das letzte Kriterium für zusammengehörige Merkmale aus Abschnitt 5.2.1 erfüllen. Das Prinzip der Nachbarschaft besagt, dass die Anordnung benachbarter Merkmale zueinander in einem gewissen Rahmen erhalten bleibt. Die physikalische Erklärung dieses

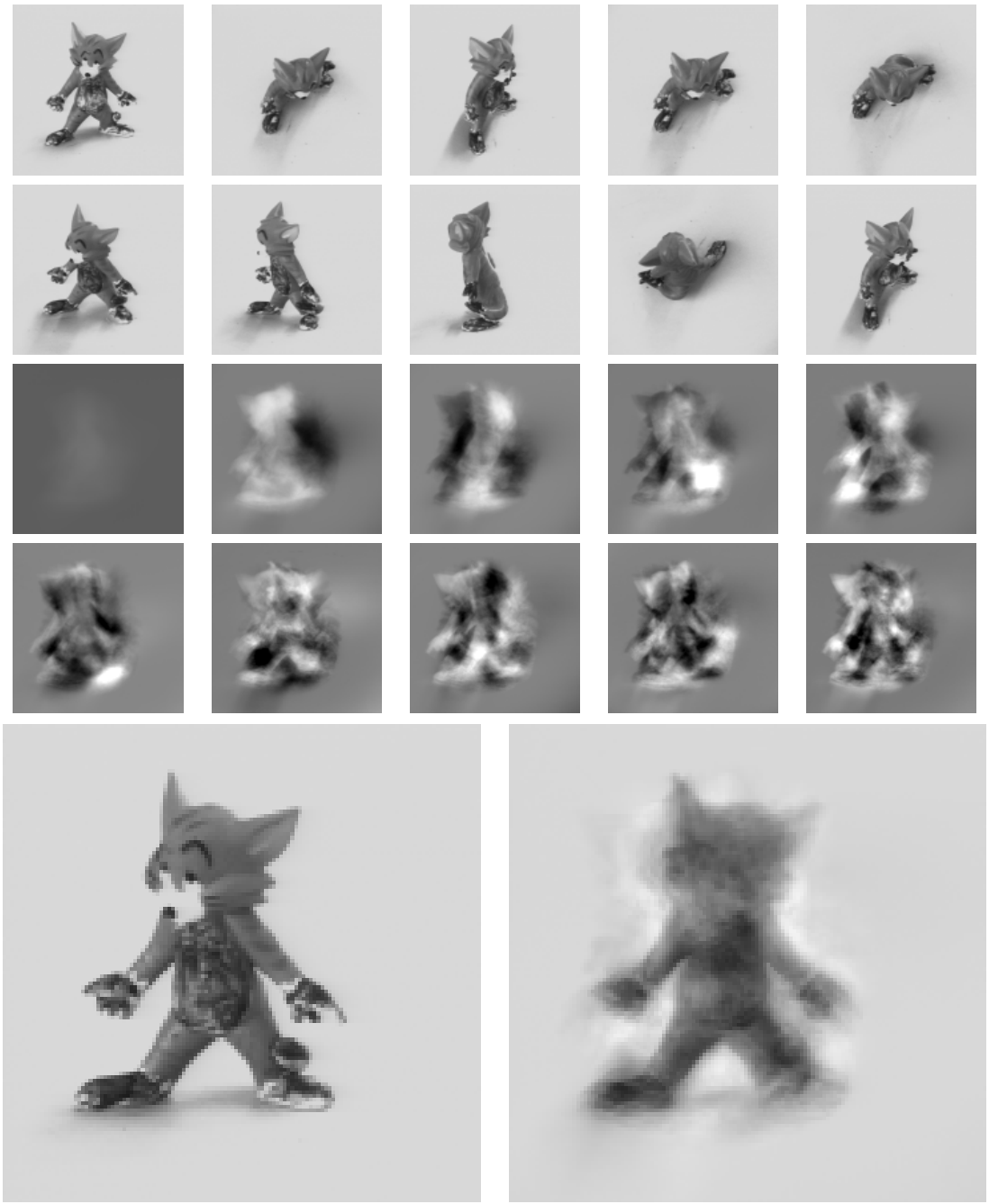


Abbildung 5.7: Beispiel für die Approximation eines Bilds durch Eigenbilder. Es wurden 300 zufällige Ansichten der Cartoonfigur ausgewählt (oben die ersten 10) und daraus 10 Eigenbilder extrahiert (Mitte). Diese approximieren eine weitere, zufällig ausgewählte Ansicht (unten). Originalbilder aus der Sammlung *Densely Sampled View Spheres*[PZM02] mit freundlicher Genehmigung von Gabriele Peters: `http://ls7-www.cs.uni-dortmund.de/~peters/pages/research/modeladaptsys/modeladaptsys_vba_rov.html`

Effekts liegt in der häufig auftretenden Situation, dass zwischen zwei Bildern eines Objektes der Winkel Kamera1-Objekt-Kamera2 klein im Verhältnis zur Rotation zwischen Kamera 1 und Kamera 2 ist, sodass die Projektionen des Objekts auf die jeweiligen Bildebenen sehr ähnlich sind, aber auf unterschiedliche Orte fallen.

Wir können uns diesen Effekt zunutze machen, indem wir jedem *möglichen Kandidatenpaar* $\boldsymbol{m_i \leftrightarrow m'_j}$ einen „Abstand“ zuordnen, der durch geometrische Betrachtungen definiert wird. Dieser Abstand nennt sich *geometrischer Fehler*.

Ebene-zu-Ebene-Transformation, Homographie

In Abschnitt 4.2.1 wurde gezeigt, dass zwei Bilder, die nur durch Rotation der Kamera zustande kommen, durch eine planare projektive Abbildung deckungsgleich gemacht werden können:

$$\tilde{\boldsymbol{m}}' = \boldsymbol{H}\tilde{\boldsymbol{m}}$$

Nehmen wir einmal an, $\boldsymbol{H}$ wäre bekannt. In diesem Fall wird das Korrespondenzproblem viel leichter, weil wir einfach nur jeden Punkt $\boldsymbol{m}$ auf das andere Bild abbilden müssen. Dort können wir dann den nächsten Nachbarpunkt $\boldsymbol{m'}$ finden und die Korrespondenz $\boldsymbol{m \leftrightarrow m'}$ festlegen.

Um die Suche symmetrisch zu machen (die Bilder sollen identische Rollen einnehmen) definieren wir den *Transferfehler* über eine Homographie $\boldsymbol{H}$ wie folgt:

$$\boldsymbol{d^2_H(f, f') = d^2(H\tilde{m}, \tilde{m}') + d^2(\tilde{m}, H^{-1}\tilde{m}')}$$

wobei $\tilde{\boldsymbol{m}}$ und $\tilde{\boldsymbol{m}}'$ die homogenen Koordinaten der Merkmalspunkte $\boldsymbol{f}$ und $\boldsymbol{f'}$ sind und $\boldsymbol{d^2([x, 1]^T, [x', 1]^T) = x^T x'}$. Das Arbeiten mit quadrierten Fehlern vermeidet das (rechnerisch aufwändige) Wurzelziehen und gewichtet größere Fehler stärker als kleinere.

Trägt man nun den Transferfehler anstelle des Ähnlichkeitsmaßes in die Paarungsmatrix $\boldsymbol{C}$ ein, $\boldsymbol{c_{ij} = d^2_H(f_i, f'_j)}$, so erhält man ein Matching, welches den quadratischen Transferfehler minimiert. Vorausgesetzt, dass $\boldsymbol{H}$ korrekt ermittelt wurde, ist dies ein sehr gutes Ergebnis.

Auch in Situationen, in denen eine Homographie den Unterschied zweier Bilder nicht vollständig beschreiben kann, approximiert sie diesen oft noch „gut genug“, um ein korrektes Matching zu erreichen.

Epipolargeometrie

Einen ähnlichen Abstand kann man auch ausrechnen, wenn die Fundamentalmatrix $\boldsymbol{F}$ zwischen den Bildern bekannt ist. Der geometrische Fehler wird dann als Abstand eines Punktes zur Epipolarlinie des korrespondierenden Punktes in der Bildebene definiert:

$$\boldsymbol{d^2_F(f, f') = d^2(\tilde{m}', F\tilde{m}) + d^2(\tilde{m}, F^T\tilde{m}')}$$

wobei

$$\boldsymbol{d^2(\tilde{x}, \tilde{l}) = d^2([x, 1]^T, [l, c]^T) = \frac{(x^T l)^2}{l^T l}}$$

den Abstand Punkt/Linie beschreibt.

3D-zu-2D-Projektion

Die Situation ist hier insofern anders als bei den zuvor genannten Fehlermaßen, als dass das Matching nicht mehr symmetrisch ist. Den Bildern kommen verschiedene Bedeutungen zu. Zu einem Merkmal $\boldsymbol{f}$ des einen Bildes sei nur seine zweidimensionale Position $\boldsymbol{m}$ bekannt, zu einem Merkmal $\boldsymbol{f'}$ des anderen Bildes sei ein Punkt $\boldsymbol{M}$ im Weltkoordinatensystem gegeben. Wie schon in 4.3.3 ist bei bekannter Projektionsmatrix $\boldsymbol{P}$ der geometrische Fehler als Abstand des projizierten zum gemessenen Punkt in der Bildebene definiert:

$$\boldsymbol{d_P^2(f, f') = d^2(\tilde{m}, P\tilde{M})}$$

mit $\boldsymbol{d^2}$ definiert als Abstand von 2D-Punkten in der Bildebene, $\boldsymbol{d^2([x, 1]^T, [x', 1]^T) = x^T x'}$.

Es stellt sich allerdings die Frage, woher die Projektionsmatrix kommen soll. Schließlich ist der Zweck der Korrespondenzenfindung unter anderem genau die Ermittlung von $\boldsymbol{P}$. Es liegt also ein klassisches „Henne-und-Ei"-Problem vor. Ohne Projektionsmatrix erhalten wir keine guten Korrespondenzen und ohne gute Korrespondenzen erhalten wir keine Projektionsmatrix.

Ein Ausweg aus dem Dilemma wird in 5.3 vorgestellt: Korrespondenzen und Projektionsmatrix sind *zusammen* zu ermitteln.

5.2.5 Kombination von Ähnlichkeit und geometrischem Fehler

Ähnlichkeit und geometrischer Fehler tragen zusammen zu einem guten Matching bei. Wie jedoch kombiniert man die beiden Maße? Welchem Maß schenkt man mehr Glauben, wenn sich die Aussagen widersprechen? Eine Möglichkeit besteht darin, ein Matching zu finden, das eines der Maße optimiert und dabei Randbedingungen aus dem zweiten Maß erfüllt:

- Maximiere *Ähnlichkeit* unter *geometrischer Fehler* $< \boldsymbol{\epsilon}$ oder
- Minimiere *geometrischen Fehler* unter *Ähnlichkeit* $> \boldsymbol{\epsilon}$.

Geschickter ist es jedoch, beide Maße zu einem gemeinsamen Fehlermaß zusammenzuführen. Angenommen, die Wahrscheinlichkeit[5], dass es sich bei einem Merkmalspaar $\boldsymbol{f \leftrightarrow f'}$ mit Ähnlichkeit $\boldsymbol{s}$ und geometrischem Fehler $\boldsymbol{d}$ um eine Korrespondenz handelt, ist $\boldsymbol{p(s, d)}$. Sei weiterhin (vereinfachend) angenommen, wir könnten $\boldsymbol{p}$ in unabhängige Wahrscheinlichkeiten $\boldsymbol{p(s, d) = q(s)r(d)}$ zerlegen. Dann können wir $\boldsymbol{p(s, d)}$ für jedes Merkmalspaar $\boldsymbol{f_i \leftrightarrow f'_j}$ berechnen und das Matching suchen, welches das Produkt der $\boldsymbol{p(s, d)}$ maximiert.

Durch Verwendung einer logarithmischen Skala kann das Problem in ein additives gewandelt werden. Statt $\boldsymbol{\prod p(s, d)}$ wird $\boldsymbol{\sum \log(p(s, d))}$ maximiert. Dies kann durch die in 5.2.2 vorgestellten Methoden geschehen.

Bleibt die Frage, wie die Wahrscheinlichkeiten $\boldsymbol{q}$ und $\boldsymbol{r}$ zu berechnen sind. Für den geometrischen Fehler können wir annehmen, dass er eine Gaußverteilung mit Erwartungswert $\mathbf{0}$ und Standardabweichung $\boldsymbol{\sigma}$ annimmt. Somit ist die Wahrscheinlichkeit $\boldsymbol{q}$ von der Form $\boldsymbol{q = e^{-d^2/2\sigma^2}}$.

Für die Verteilung der Kreuzkorrelation wählen wir vereinfachend ebenfalls die Gaußverteilung (mit Zentrum bei $\mathbf{1}$): $\boldsymbol{r = e^{(NCC-1)/2\tau^2}}$.

Es muss also $\boldsymbol{\sum \log(e^{-d^2/2\sigma^2} e^{(NCC-1)/2\tau^2}) = \sum -d^2/2\sigma^2 + (NCC-1)/2\tau^2}$ maximiert werden.

[5] genau genommen handelt es sich um ein *Likelihoodmaß*

5.3 Robuste Parameterbestimmung

Um ein exaktes Matching zu berechnen, ist die Kombination des Ähnlichkeitsmaßes mit dem geometrischen Fehler notwendig. Der geometrische Fehler wird jedoch über ein mathematisches Modell berechnet, dessen Parameter zum Zeitpunkt der Berechnung noch nicht zur Verfügung stehen. Dies ist ein Problem, schließlich lassen sich die Parameter nur aus Merkmalskorrespondenzen bestimmen, die noch nicht vorliegen.

Die Lösung besteht in der Hinzunahme eines „provisorischen“ Matchings, welches auch eine gewisse Menge falscher Zuordnungen beinhalten darf. Aus diesem *Initialmatching* kann ein Modell bestimmt werden, das die Ermittlung eines exakten Matchings ermöglicht. Das Initialmatching kann beispielsweise aus der alleinigen Maximierung des Ähnlichkeitsmaßes gewonnen werden.

Wie bestimmen wir ein exaktes Modell unter Verwendung von Initialmatches, die „Ausreißer“ enthalten? Die Modellschätzer aus 4.2.1, 4.2.2 und 4.3.1 können zwar eine Streuung in den Eingaben ausgleichen. Diese muss jedoch einer Normalverteilung unterliegen. Auf Ausreißer reagieren diese Algorithmen außerordentlich empfindlich, da sie die Summe der Fehlerquadrate minimieren und somit großen Fehlern eine überproportional große Gewichtung geben.

5.3.1 Der RANSAC-Algorithmus

Der RANSAC-Algorithmus ist ein stochastischer Algorithmus. Es wird eine zufällige Teilmenge der Initialmatches entnommen, in der Hoffnung, dass diese Teilmenge frei von Ausreißern ist. Daraus wird dann das Modell ermittelt. Zum Schluss wird geprüft, wie gut sich die gesamten Daten durch das Modell ausdrücken lassen. Dies wird solange wiederholt, bis ein „gutes“ Modell gefunden wurde.

Um die Wahrscheinlichkeit zu erhöhen, dass die Zufallsmenge ausreißerfrei ist, wird sie möglichst klein gewählt. Um beispielsweise mit dem DLT-Algorithmus eine Homographie zu erzeugen, werden 4 Punktkorrespondenzen benötigt.

Hier der RANSAC-Algorithmus in Pseudocode:

```
für i in 1..N:
  Ziehe eine minimale Zufallsmenge aus den Initialmatches
  Ermittle Modellparameter M mit der gefundenen Zufallsmenge
  Sammle die Merkmalspaare, die gut mit M übereinstimmen
  Wenn mehr als T Merkmalspaare gefunden:
    Ermittle verbesserte Modellparameter M' unter Einbeziehung
    der gefundenen Merkmalspaare
    ERFOLG! Gib Merkmalspaare und M' zurück

Wenn N erreicht:
  FEHLSCHLAG!
```

Wie viele Versuche sind notwendig, um mit hoher Wahrscheinlichkeit mindestens eine ausreißerfreihe Zufallsmenge zu erhalten? Angenommen, der Anteil an Ausreißern in der Initialmenge betrage ϵ. Soll mit Wahrscheinlichkeit p mindestens eine ausreißerfreihe Zufallsprobe der Größe s gefunden werden, so ist die Anzahl der Iterationen $N = \log(1-p)/\log(1-(1-\epsilon)^s)$. [HZ00, 3.7.1 S.104]

Zur Bestimmung der Homographie (4 Punktpaare, also $s = 4$) mit 99%iger Erfolgswahscheinlichkeit bei einer Fehlerquote von maximal 50% werden somit 72 Versuche benötigt.

Least Median Of Squares Ein enger Verwandter von RANSAC ist der LMedS-Algorithmus. Es wird angenommen, dass mindestens die Hälfte der Initialmatches korrekt sind. Demnach muss das Modell, welches den Median

$$\underset{k}{\text{med}}\, d_M(f_{i_k}, f'_{j_k})$$

minimiert, diese Hälfte gut beschreiben. Dabei sind die f_{i_k} und f'_{j_k} einfach die Initialmatches. Anstatt den gesamten Suchraum zu durchlaufen, wird das Modell in RANSAC-Manier durch Ziehen von Zufallsteilmengen bestimmt. [Zha95]

5.3.2 M-Schätzer

Die mangelnde Robustheit der Modellschätzer aus 4.2.1, 4.2.2 und 4.3.1 beruht auf der Minimierung von Fehlerquadraten, die großen Fehlern (also Ausreißern) ein hohes Gewicht beimisst. Es ist möglich, diesen Effekt zu reduzieren, indem eine subquadratische Funktion minimiert wird. Diesen Ansatz verfolgen so genannte *M-Schätzer*. Die Optimierungsaufgabe

$$\min_p \sum_i r_i^2(p)$$

wird ersetzt durch die Aufgabe

$$\min_p \sum_i \rho(r_i^2(p))$$

wobei p der zu bestimmende Parametervektor ist und r_i^2 die Komponenten der Fehlerfunktion. Eine globale Optimierung vorzunehmen ist schwer. Jedoch ist es möglich, das Problem durch ein Iterationsverfahren zu lösen, in dem in jedem Iterationsschritt ein gewichtetes Kleinste-Quadrate-Problem zu lösen ist. Unter der Voraussetzung, dass man das ursprüngliche Problem minimieren konnte, kann man auch, sofern man einen Initialwert besitzt, den M-Schätzer bestimmen. Insbesondere lassen sich M-Schätzer ohne großen Aufwand in Problemen einsetzen, die sowieso durch nichtlineare Optimierung (beispielsweise mit einem Levenberg-Marquardt-Verfahren) gelöst werden. Zhang gibt in [Zha95] eine gute Einführung in das Gebiet und stellt einige gebräuchliche M-Schätzer vor. Eine formalere Einführung findet in [PTVF92] statt.

Kapitel 6

Der STAGE-Algorithmus

In diesem Kapitel soll der Algorithmus dokumentiert werden, welcher in der Trackingsoftware *StageDesigner* verwirklicht wurde. Es handelt sich um die Software, die zur Vorbereitung dieser Diplomarbeit entwickelt wurde. STAGE ist das darin verwendete Tracking-Kernstück[1].

STAGE ist das Resultat Monate andauernder Entwicklung, jedoch vor allem eines Lernprozesses. Er erhebt nicht den Anspruch, der *beste* Algorithmus zu sein oder bahnbrechende Innovationen zu beinhalten. Jedoch verwendet er einige der in den Kapiteln 4 und 5 beschriebenen Methoden und kann daher als Anschauungsobjekt dienen.

Es sollte ein Tracking-System entwickelt werden, welches

- ohne die Verwendung von speziellen *Markern* auskommt, also nur „natürliche“ Merkmale des Bildes verwendet,
- in Echtzeit oder zumindest nahe Echtzeit arbeitet,
- mit handelsüblichen Digitalvideokameras funktioniert,
- zuvor erstellte und mit 3D-Informationen versehene Aufnahmen der Szene aus verschiedenen Winkeln verwendet, jedoch
- ohne die Erstellung eines Polygonmodells der Szene auskommt.

Diesen Ansprüchen genügt STAGE. Jedoch wurde das Ziel, den Algorithmus in beliebigen Umgebungen und verschiedenen Licht- und Umweltbedingungen einsetzbar zu machen, nicht erreicht. Unter kontrollierten Bedingungen liefert STAGE jedoch respektable Ergebnisse.

6.1 Beschreibung des STAGE-Algorithmus

Es folgt eine Einführung in die Funktionsweise von STAGE.

6.1.1 Der STAGE-Zyklus

STAGE ist ein ereignisgesteuerter Algorithmus, der auf nur ein Ereignis reagiert, nämlich das Eintreffen eines neuen Bildes. Daraufhin setzt sich der STAGE-Mechanismus in Gang. Nach einiger Zeit (einigen Millisekunden) wird auf das nächste Bild gewartet.

STAGE kennt zwei Zustände: *registriert* und *nicht registriert*. Der Anfangszustand ist *nicht registriert*. Das bedeutet, das keine Registrierung der Kamera vorliegt.

[1] Der Autor hat es sich nicht nehmen lassen, dem Algorithmus nach alter Sitte eine rekursive Abkürzung zu geben: STAGE steht für **S**TAGE **T**racking **A**lgorithm for **G**eneric **E**nvironments

Name	**STAGE**
Kategorie	Online-Tracker
Merkmalstyp	Punktmerkmale
Merkmalsextraktor	Harris Corner Detector
Merkmalsgröße	Einstellbar, Vorgabe: $\mathbf{11 \times 11}$
Ähnlichkeitsmaß	SSD für aufeinander folgende Bilder, ansonsten NCC oder wahlweise NCC-R
Ermittlung der Projektionsmatrix	Pose aus Homographie, anschließend Optimierung des geometrischen Fehlers durch Levenberg-Marquardt-Algorithmus
Robuste Modellbestimmung	RANSAC-Algorithmus
Relative Posebestimmung	aus Pose des vorangegangen Kamerabilds
Absolute Posebestimmung	aus Pose des aktuellen Keyframes
Auswahl des Keyframes	Round-Robin-Strategie

Tabelle 6.1: Einige Kenndaten von STAGE.

STAGE verwendet so genannte Keyframes. Das sind Bilder der Szene aus verschiedenen Ansichten, die zuvor gemacht wurden. Über den Harris Corner Detector wurden Merkmalspunkte in diesen Keyframes extrahiert. In einer Vorbereitungsphase wurden die Koordinaten dieser Merkmalspunkte in einem dreidimensionalen Weltkoordinatensystem berechnet. Diese dienen als absolute Information für die Kameraregistrierung.

Initiale Registrierung Im Zustand *nicht registriert* wird ein neues Bild mit einem dieser Keyframes verglichen. Das verwendete Keyframe wird Kandidat genannt. Können eine genügend große Anzahl Merkmalskorrespondenzen zum aktuellen Bild gefunden und eine Kamerapose berechnet werden, so war der Vergleich erfolgreich. Der Keyframe-Kandidat wird zum aktuellen Keyframe. Für spätere Vergleiche wird ein neuer Kandidat ausgesucht. Da die initiale Registrierung geglückt ist, wird in den Zustand *registriert* gewechselt.

Registrierung mit Keyframe Befindet sich der STAGE-Algorithmus bei Eintreffen eines neuen Bildes im Zustand *registriert*, so kann er davon ausgehen, dass das vorangegangene Bild ebenfalls mit dem aktuellen Keyframe registriert wurde. Die Pose des letzten Bildes der Kamera ist also bekannt. Es wird eine Pose aus dem Vergleich des aktuellen und vorangegangenen Kamerabilds berechnet. Eine weitere Pose wird aus dem Vergleich des aktuellen Kamerabilds mit dem Keyframe berechnet. Es wird die bessere Pose ausgewählt und durch nichtlineare Optimierung verfeinert.

Alternative Registrierung Die Registrierung mit dem Keyframe ist vielleicht nicht von Dauer, also müssen rechtzeitig Alternativen erkundet werden. Deshalb wird bei abnehmender Qualität der Registrierung mit dem Keyframe bereits geprüft, ob ein Keyframe-Kandidat eine bessere Registrierung ermöglicht. Dies geschieht vollkommen analog zur initialen Registrierung. Ist der Kandidat besser als das Keyframe, wird das bisherige Keyframe fallen gelassen und stattdessen der Kandidat übernommen.

Das Flussdiagramm 6.1 auf Seite 57 verdeutlicht diese Abläufe noch einmal. Nun folgt ein detaillierter Einblick in die einzelnen Arbeitsschritte.

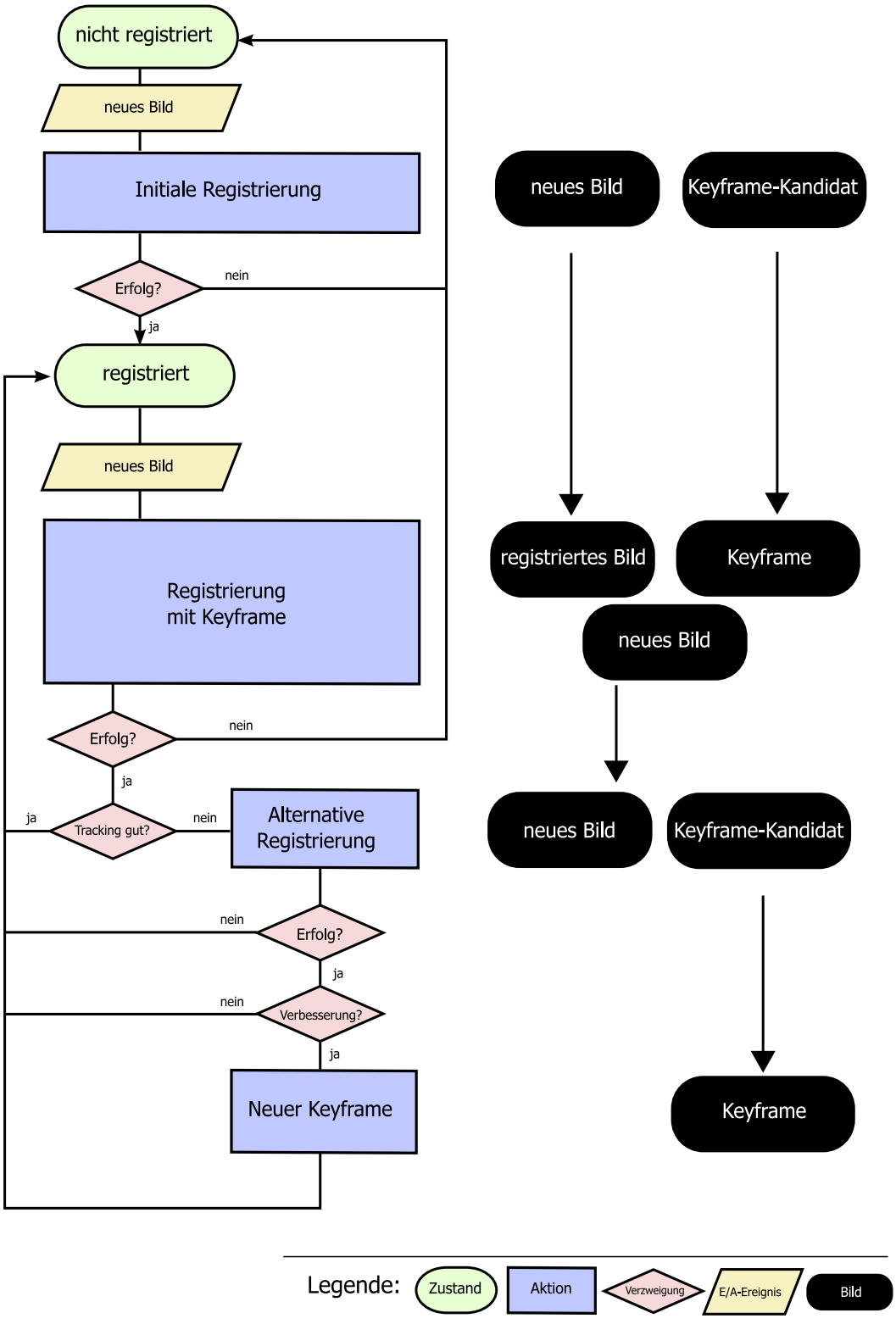


Abbildung 6.1: Der Ablauf des STAGE-Algorithmus als Flussdiagram. Die rechte Seite zeigt die an jedem Arbeitsschritt beteiligten Bilder.

6.1.2 Eintreffen eines neuen Bildes

Bei der Bearbeitung eines Bildes müssen zunächst einige technische Schwierigkeiten aus dem Weg geräumt werden. Handelsübliche Digitalvideokameras liefern zumeist Bilder, die in einem Zeilensprungverfahren dargestellt werden. Dabei werden, vereinfacht gesagt, zwei Bilder, die zu unterschiedlichen Zeitpunkten (bei PAL-Kameras genau im Abstand einer 50tel Sekunde) aufgenommen wurden, in einem Bild zeilenweise verflochten. Dies führt zu so genannten Kammeffekten, welche sich extrem negativ auf die zu verwendenden Bildbearbeitungsfunktionen auswirken. Deshalb wird eines der Halbbilder verworfen. Das übrig gebliebene Bild ist nur noch halb so hoch und wird durch Stauchung wieder in das ursprüngliche Bildseitenverhältnis (meist 4:3) gebracht.

Nun wird die Farbinformation verworfen und nur das Graustufenbild betrachtet, welches mittels der bekannten intrinsischen Kameraparameter so verzerrt wird, dass die Linsenverzeichnung ausgelöscht wird. Mittels des Harris Corner Detectors werden die 80^2 besten (also die höchste *Cornerness* aufweisenden) Merkmalspunkte extrahiert. Um später die Berechnung der normalisierten Kreuzkorrelation (NCC und NCC-R) zu beschleunigen, werden zu jedem Merkmal Mittelwert, Standardabweichung und Gradient im Merkmalsfenster vorberechnet.

6.1.3 Initiale Registrierung

Es soll versucht werden, das aktuelle Bild mit einem Keyframekandidaten zu matchen und dabei die aktuelle Projektionsmatrix zu ermitteln.

Zwischen je zwei Merkmalspunkten $\boldsymbol{f_i}$ des aktuellen Bildes und $\boldsymbol{f'_j}$ des Kandidaten wird die Kreuzkorrelation NCC berechnet. Es kann auch die aufwändiger zu berechnende rotationsinvariante NCC-R verwendet werden. Die einzelnen Ergebnisse werden in die Einträge einer Matching-Matrix $\boldsymbol{C} = (\boldsymbol{c})_{\boldsymbol{ij}}$ geschrieben. Durch Ermittlung der Matches lokaler Optimalität ist ein initiales Matching gegeben.

Ein RANSAC-Algorithmus versucht nun, mit diesen Initialmatches in maximal 30 Durchgängen eine Projektionsmatrix $\boldsymbol{P}$ zu finden, welche die 3D-Koordinaten $\boldsymbol{M_j}$ der Merkmalspunkte des Kandidaten „möglichst gut“ auf die 2D-Koordinaten $\boldsymbol{m_i}$ des aktuellen Bildes abbildet. „Möglichst gut“ heißt hierbei, dass die Anzahl der *lokal optimalen* Merkmalspaare (s. Abschnitt 5.2.2) mit

- Abstand $\boldsymbol{d(m_i, PM_j) < 6}$ Pixel
- NCC (bzw. NCC-R) zwischen $\boldsymbol{f_i}$ und $\boldsymbol{f'_j > 0.6}$

mindestens 30% der Anzahl an Merkmalspunkten des Keyframe-Kandidats betragen soll.

Die Projektionsmatrix wird durch die Suche nach einer Homographie ermittelt (vergl. S.34). Das bedeutet, dass 4 korrekte Merkmalspaare in komplanarer Lage gefunden werden müssen. Diese Komplanaritätsbedingung hat zur Konsequenz, dass der RANSAC-Algorithmus nicht einfach eine beliebige Zufallsmenge aus den Initalmatches zur Bestimmung der Homographie verwenden kann. Vielmehr muss gezielt nach komplanaren Punkten gesucht werden. Dies ist jedoch nicht sehr schwer, da die 3D-Koordinaten der Merkmalspunkte des Keyframe-Kandidaten ja bekannt sind. Es kann deshalb eine Zufallsprobe schnell auf Komplanarität geprüft werden. Die RANSAC-Implementierung in STAGE verwirft solange die gefundenen Zufallsproben, bis eine komplanare Probe gefunden wurde (bis zu 50 mal pro Zufallsprobe).

[2] Konstanten dieser Art sind relativ willkürlich gewählt und können bei Bedarf angepasst werden

Wurde eine Projektionsmatrix gefunden, die genug Merkmalspaare erzeugt (also 30% der Merkmalspunkte des Kandidaten matchen kann), so wird der Kandidat zum neuen Keyframe und die Projektionsmatrix wird übernommen.

6.1.4 Registrierung mit dem Keyframe

In ähnlicher Weise funktioniert auch die Registrierung, wenn ein Keyframe vorhanden ist. Es wird eine Projektionsmatrix ermittelt, die die Merkmalspunkte des Keyframes in selber Weise auf die Merkmale des aktuellen Bildes abbildet. Die so bestimmte Pose wird *absolute Pose* genannt, da sie sich nur aus den festen 3D-Positionen der Merkmale im Keyframe errechnet.

In den meisten Fällen ist die aufwändige Berechnung der absoluten Pose jedoch gar nicht notwendig, denn es steht noch die Pose des vorherigen Kamerabildes zur Verfügung. Da die Aufgabe, zwei „benachbarte“ und somit sehr ähnliche Bilder miteinander zu matchen sehr viel einfacher ist, als das Matchen von sich stark unterscheidenden Bildern, ist eine Posebestimmung über das vorherige Bild ein deutlich robusteres Verfahren.

Zur Bestimmung der *relativen Pose* wird wieder ein RANSAC-Verfahren angewandt. Aus den Merkmalspaaren, deren Summe der quadratischen Differenzen (SSD) minimal ist, werden Initialmatches extrahiert. Zur weiteren Effizienzsteigerung werden dabei nur Merkmalspaare betrachtet, deren Positionen sich nur wenig unterscheiden. Dazu wird um ein Merkmal im einen Bild ein kleines Suchfenster im anderen Bild gelegt, in welchem sich das korrespondierende Merkmal befinden darf. Die so gefundenen Initialmatches besitzen eine viel geringere Ausreißerquote als die Initialmatches aus den zuvor genannten Fällen. Deshalb benötigt der anschließende RANSAC-Algorithmus zur Posebestimmung aus einer Homographie meist nur wenige Iterationen, um eine gute Registrierung zwischen den beiden Bildern zu erreichen.

Dies geschieht über die Suche nach Merkmalspaaren, welche die folgenden Eigenschaften erfüllen:

- Suchfenster: Beide Merkmale dürfen nicht mehr als 2% der Bildbreite auseinanderliegen.
- Ähnlichkeit: Die Summe der quadrierten Differenzen zwischen beiden Merkmalen darf nicht mehr als 0,03 betragen (Intensitätswerte wurden auf das Intervall $[\mathbf{0, 1}]$ normiert): $\boldsymbol{SSD(f, f') < 0,03}$
- Der maximale geometrische Fehler zwischen Merkmalspunkten im aktuellen Bild und projizierten dreidimensionalen Punkten des vorherigen Bildes darf nicht mehr als 2 Pixel betragen: $\boldsymbol{d(\tilde{m}, P\tilde{M}) < 2}$

Dabei werden bis zu 20 RANSAC-Iterationen durchgeführt, bis eine Projektionsmatrix P gefunden wurde, die unter obigen Bedingungen mindestens 30% der Merkmalspunkte matchen kann.

Die Annahme, dass sich aufeinander folgende Bilder stark ähneln, ist in manchen Fällen ungültig, beispielsweise bei schnellen Kameraschwenks. In solchen Fällen schlägt die Merkmalssuche unter diesen relativ strikten Bedingungen fehl. Deshalb sieht STAGE eine mehrstufige Fallback-Strategie auf weniger strikte Suchparameter vor. So wird der „Normalfall“ effizient behandelt, und „schwierige Fälle“ führen nicht so schnell zu einem Abbruch des Trackings.

Ermittlung der besten Projektionsmatrix Die ermittelte relative Projektionsmatrix wird daraufhin untersucht, wie gut sie die Merkmale des Keyframes auf das aktuelle Bild abbildet.

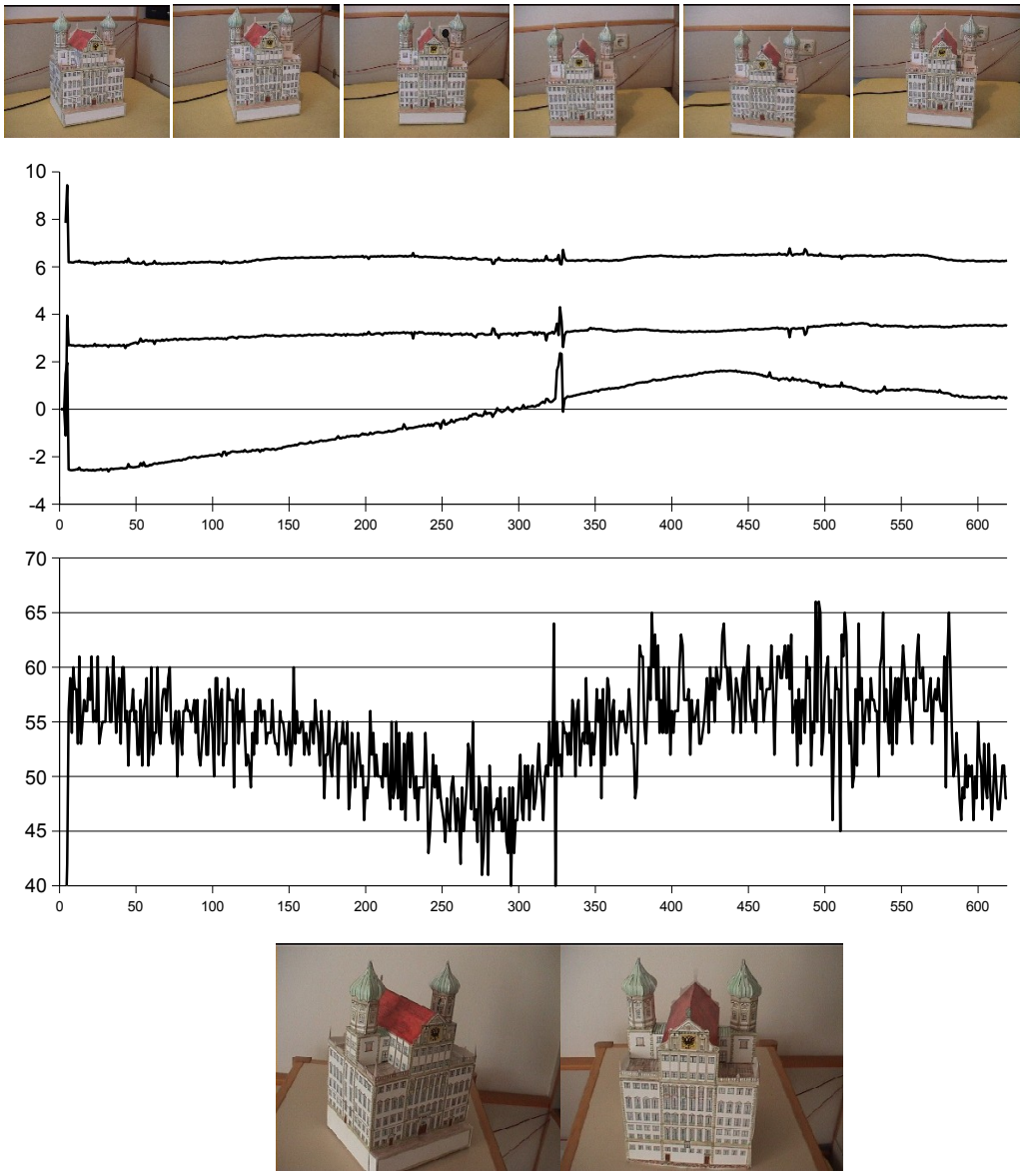


Abbildung 6.2: Der Verlauf einer zu trackenden Sequenz. Das obere Diagramm bildet den zeitlichen Verlauf der (von oben nach unten) Z,Y und X-Komponenten der Kameraposition ab. Der Sprung bei Bild 324 wurde durch einen „unsauberen“ Keyframe-Übergang ausgelöst. Die Anzahl der Verfolgten Keyframe-Merkmale (unteres Diagramm) nimmt daraufhin wieder zu. Die beiden verwendeten Keyframes sind unten abgebildet.

Ist dieser Wert höher als die Anzahl direkt ermittelter Matches aus der absoluten Posebestimmung, wird die relative Pose übernommen, ansonsten die absolute.

6.1.5 Alternative Registrierung

Um zu gewährleisten, dass möglichst zu jedem Zeitpunkt der beste Keyframe ausgesucht wurde, werden auch bei geglückter Registrierung mit dem Keyframe weitere Kandidaten getestet. Dabei werden die Kandidaten reihum aus der Liste der Keyframes gezogen. Die Registrierung findet dann genauso wie in der initialen Registrierung statt. Um Zeit zu sparen, werden jedoch nur 10 RANSAC-Iterationen ausgeführt. Ist die Anzahl der Matches um mindestens 10% größer als zum Keyframe, wird der Kandidat zum aktuellen Keyframe und die ermittelte Pose wird übernommen.

6.1.6 Nichtlineare Verfeinerung

Die Projektionsmatrizen, die über den linearen Algorithmus *Pose aus Homographie* ermittelt wurden, sind oft relativ ungenau. Dies liegt zum einen darin begründet, dass die Homographien, aus denen sie berechnet werden, nur einen algebraischen Fehler minimieren. Zum anderen fließen in die Berechnung der Homographie nur die Punkte auf einer Ebene im Raum ein. Alle anderen Punkte müssen aus der Berechnung ausgenommen werden. Auch ungenau ermittelte Kalibrationsparameter wirken sich direkt auf die Qualität der Projektionsmatrix aus.

Deshalb wird in einem letzten Schritt eine nichtlineare Optimierung der Projektion nach dem Algorithmus von 4.3.4 auf S.35 vorgenommen. Darin wird eine Rotation und Translation der Kamera gesucht, sodass der geometrische Fehler minimiert wird, welcher zwischen den Merkmalspunkten im aktuellen Bild und den projizierten Merkmalen, die im Keyframe vorkommen, auftritt. Es werden alle Merkmalspunkte miteinbezogen.

6.2 STAGE im Vergleich mit dem Tracker der EPFL

Das Tracking-System der EPFL diente in vielen Punkten als Vorbild für STAGE. Beide Systeme verwenden Keyframes mit vorher ermittelter Registrierung, um absolute Informationen einfließen zu lassen. Beide Systeme versuchen, die absoluten Informationen mit relativen Informationen zu kombinieren, um möglichst wenig Jitter aufzuweisen. Der Hauptunterschied besteht darin, dass STAGE auch ohne Polygonmodell funktionieren sollte. Dies führt zu einigen Einschränkungen.

Es soll nun ein Vergleich der beiden Systeme hinsichtlich der wichtigsten Verfahren gezogen werden.

6.2.1 Keyframe-Auswahl

Die Keyframe-Auswahl des EPFL-Systems geschieht deterministisch in Abhängigkeit von der Kamerapose[VLF04b, S.1387]. Dies ermöglicht den schnellen Zugriff auf das jeweilige Keyframe, sobald die Pose berechnet ist.

STAGE hingegen probiert sämtliche Keyframes der Reihe nach aus, bis es auf einen Kandidaten trifft, der sich gut mit dem Kamerabild matchen lässt. Pro eingetroffenem Kamerabild wird ein Keyframekandidat getestet. Die Suche wird unterbrochen, wenn das Matching mit

dem aktuellen Keyframe gut genug ist. Sobald die Anzahl der verfolgten Merkmale unter einen Grenzwert fällt, wird die Suche wieder aufgenommen.

Dieses Verhalten hat sich als recht erfolgreich herausgestellt, wenn die Anzahl der Keyframes nicht zu groß wird. Ein Schwachpunkt ist, dass die Qualität des Matchings von der Anzahl der Merkmalspunkte eines Keyframes abhängt und somit einige Keyframes bevorzugt ausgesucht werden.

6.2.2 Keyframe-Matching

Dem EPFL-System steht zum Matchen des aktuellen Kamerabilds mit dem Keyframe die Möglichkeit zur Verfügung, eine Ansicht des Keyframes aus der Perspektive des Vorgängerbilds zu berechnen. Dadurch wird der optische Vergleich mit dem aktuellen Kamerabild mittels normalisierter Kreuzkorrelation wesentlich aussagekräftiger, da störende Rotations-, Skalierungs- und Projektivtransformationen der Merkmalspunkte ausgeglichen werden.

Diese Möglichkeit steht STAGE nicht zur Verfügung. Damit sind die NCC-Werte zwischen korrelierenden Merkmalen kleiner und unterscheiden sich weniger von nicht korrelierenden. Das Keyframe-Tracking muss wesentlich geringer korrelierte Merkmalspaare untersuchen, was den Anteil an Falschzuordnungen erhöht. Dadurch leidet die Robustheit des Keyframe-Trackings.

6.2.3 Tracking

Das EPFL-System entnimmt aus der Korrelationsanalyse der Merkmalspunkte des jeweils vorherigen und des aktuellen Kamerabilds eine Anzahl von 2D-Merkmalspaaren, unter denen sich auch einige Falschzuordnungen befinden können. Genauso wird aus dem Vergleich der erzeugten Ansicht des Keyframes mit dem aktuellen Kamerabild eine Anzahl von 3D-Merkmalspaaren aufgestellt. In einer gemeinsamen Optimierung wird nun eine Kamerapose berechnet, die mit diesen Beobachtungen übereinstimmt. Um Robustheit gegen Falschzuordnungen zu gewährleisten, wird ein M-Schätzer minimiert. Einen Anfangswert liefert die Kamerapose des Vorgängerbilds.

Demgegenüber führt STAGE zwei getrennte Berechnungen durch. Anhand der Korrespondenzen zwischen Kamerabild und Keyframe wird eine Pose berechnet. Die Korrespondenzen zwischen aktuellem Kamerabild und dem Vorgängerbild liefern eine weitere Pose. In beiden Schritten wird ein RANSAC-Verfahren eingesetzt, um gleichzeitig korrekte Korrespondenzen und eine gute Posebestimmung und Registrierung zu erreichen. Aus den beiden errechneten Posen wird die bessere ausgewählt und anschließend nichtlinear verfeinert.

Aus zwei Gründen ist das Verfahren der EPFL als überlegen anzusehen:

- Die Berechnung berücksichtigt die Beziehung zwischen aktuellem und vorherigem Kamerabild, sowie aktuellem Kamerabild und Keyframe in einem Arbeitsschritt als gemeinsame Optimierung.
- Durch die Verwendung eines Polygonmodells der Szene ist es dem EPFL-System möglich, neue Merkmale zu integrieren, deren Weltkoordinaten unbekannt sind.

6.2.4 Ergebnisse und mögliche Erweiterungen

STAGE ist in der Lage, Objekte und Szenen zu tracken, in denen sich mindestens eine ebene Fläche mit ausreichend vielen Merkmalen findet. Die Geschwindikeit ist annähernd echtzeit-

geeignet. Auf einem Notebook mit einem *Pentium M 715* Prozessor (1,5GHz) beträgt die durchschnittliche Bearbeitungsdauer eines Bildes in halber PAL-Auflösung (352x288) etwa 38ms. Hinzu kommen jedoch noch ca. 15ms für die Dekodierung und Darstellung des Bildes, sodass die Wiederholfrequenz bei etwa 19Hz liegt. In schwierigen Szenen, die eine häufige Keyframesuche erfordern, fällt die Wiederholfrequenz auf bis zu 14Hz. Ein großer Anteil an der Rechenzeit mit etwa 20ms pro Bild entfällt auf die Merkmalsextraktion. Hier besteht noch größeres Optimierungspotential.

In relativ einfachen Szenen ist das System gut einsetzbar. Abbildung 6.2 zeigt eine Bildsequenz, in der die Kamera langsam an einem Modell des Augsburger Rathauses vorbei geführt wird. Nach einer kurzen Zeit, in der nach einem guten Keyframe gesucht wird, stabilisiert sichdas Tracking. Die meiste Zeit bleibt es auch stabil. Lediglich ein Wechsel des Keyframes erzeugt ein „Wackeln“. In anderen Durchläufen trat dieses Phänomen nicht auf.

STAGE hat Schwierigkeiten damit, ein Objekt zu finden, wenn der Hintergrund sehr viele Merkmale erzeugt. Dies ist ein allgemeines Problem von Systemen, welche die Anzahl ihrer Merkmale begrenzen. Abhilfe könnte schaffen, die Merkmalsdichte im Bildbereich des Objekts zu erhöhen.

Während des Trackings ist häufig ein leichter Zittereffekt (Jittering) sichtbar. Dies hat verschiedene Ursachen. Zum einen können falsch gepaarte oder schlecht lokalisierte Merkmale die Poseberechnung stören. Zum anderen kann die für jedes Bild vorgenommene Entscheidung zwischen relativ und absolut ermittelter Pose zu sichtbaren Sprüngen führen. Ausgleichend wirkt hier die anschließend durchgeführte nichtlineare Verfeinerung, die jedoch durch bereits ein falsch zugeordnetes Merkmal zu verzerrten Ergebnissen führen kann und dann verworfen werden muss. Hier müsste die Verwendung von M-Schätzern untersucht werden. Auch der Einsatz von Kalman-Filtern zur Glättung der Kamerapose über die Zeit könnte Abhilfe schaffen. Damit könnte auch eine Extrapolation der Kamerabewegung erfolgen, wenn das Tracking kurzzeitig abreißt.

Erschwerend für den Einsatz ist, dass die Lernphase viel „Handarbeit“ erfordert. Das Verknüpfen von Merkmalspunkten ist eine mühselige Arbeit. Zwar nehmen halbautomatische Verfahren dem Benutzer einiges an Arbeit ab. Dadurch können sich allerdings unbemerkt Fehler einschleichen. Auf längere Sicht könnten hier bessere Methoden eingesetzt werden, bis hin zur vollautomatischen Rekonstruktion der Szene aus einem Video.

Der Verzicht auf ein Polygonmodell macht das robuste Matching über weit entfernte Perspektiven schwer und begrenzt das Tracking auf die Menge der in der Lernphase gespeicherten Merkmalspunkte. Die manuelle Erstellung eines Modells erfordert allerdings zumindest Grundkenntnisse im Umgang mit CAD-Software und ist bestenfalls mühselig. Auch das Argument, Augmented-Reality-Anwendungen benötigten ein solches Modell in jedem Fall ist nicht richtig. Einen Ausweg könnte das automatische Suchen nach planar angeordneten Merkmalen liefern, was ja bei STAGE sogar in Echtzeit geschieht. Im Vorfeld ausgeführt würde so ein provisorisches Polygonmodell der Szene erzeugt, mit allen Vorteilen für das Tracking. Auch eine manuelle Zuordnung zu Ebenen dürfte einfacher sein als die Erstellung eines CAD-Modells.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Einblick in die Funktionsweise optischer Tracking- und bildverstehender Systeme vorgenommen. Dabei wurde eine mögliche Anwendung, die der *Augmented Reality*, immer im Blick behalten. Hierbei handelt es sich um die Technik, Elemente der Realität und der *Virtuellen Realität* zu einer *Erweiterten Realität* zu vereinen. Im bekanntesten Fall geschieht dies durch das Einfügen computergenerierter Objekte in das Bild einer reellen Kamera. Um dieses Erlebnis überzeugend zu machen, ist eine perspektivische Übereinstimmung der virtuellen Objekte mit dem reellen Bild notwendig. Diese Übereinstimmung versuchen Tracking-Systeme durch die exakte Erkennung und Vermessung bekannter Objekte im Bild zu erreichen. Eine solche Aufgabe vollautomatisch und in Echtzeit auszuführen ist eine große Herausforderung und wurde bisher nur unter vereinfachten Bedingungen gelöst.

Anhand dreier Beispiele wurden Systeme vorgestellt, welche die Bandbreite der Forschungsansätze demonstrieren sollen. Das System der Universität Oxford trifft vereinfachende Annahmen über die Szenengeometrie und ist in Umgebungen mit vielen ebenen Flächen einsetzbar. Dafür arbeitet es unkompliziert und ohne Vorwissen über seine Umwelt. Anders das System der Eidgenössischen Technischen Hochschule in Lausanne, das ein handgefertigtes Computermodell der Szene voraussetzt, dafür jedoch auch unter schwierigen Bedingungen eine stabile Bilderkennung erreicht. Die geringsten Voraussetzungen stellt das System ICARUS der Universität Manchester, welches die dreidimensionale Szene selbständig rekonstruiert. Dies bewältigt es jedoch nicht mehr in Echtzeit, sondern verwendet aufgezeichnete Videosequenzen.

Von essenzieller Bedeutung für die Analyse fotografischer Abbildungen ist ein Verständnis der geometrischen Vorgänge bei der Projektion von Lichtstrahlen auf eine Bildfläche. Es wurde deshalb auf die Geometrie des Kamerabilds besonders ausführlich eingegangen: Die Projektionsmatrix liefert zusammen mit einem Modell der Linsenverzeichnung ein gute Beschreibung der Beziehung zwischen Realität und Abbildung. Dieses Ergebnis wurde zur Analyse eines besonders wichtigen Spezialfalls eingesetzt: dem Vergleich zweier Abbildungen desselben Objekts aus verschiedenen Perspektiven. Deren Zusammenhang wird im allgemeinen Fall durch die Fundamentalmatrix beschrieben, die einen Punkt des einen Bildes einer Geraden des anderen zuordnet und umgekehrt. Ist die Szenenstruktur planar, lässt sich sogar eine Punkt-zu-Punkt-Beziehung finden. Es wurden Methoden gezeigt, wie sich die Parameter dieser Modelle aus bereits bekannten Punkt-zu-Punkt-Korrespondenzen schätzen lassen.

Vom Eintreffen des Bildes bis zum Tracking der Kamera ist es ein weiter Weg. Die dabei verwendeten Verfahren bauen schrittweise aufeinander auf. Der erste Schritt ist die Extraktion von Merkmalen im Bild, die geometrisch gedeutet werden können. Es wurde mit dem Harris Corner Detector ein Verfahren zur Suche nach leicht identifizierbaren Merkmalspunkten vorgestellt und auf seine Eigenschaften untersucht. Im nächsten Schritt werden gemeinsame Merkmalspunkte zweier Bilder identifiziert und verknüpft. Dabei kommen sowohl geometrische Modelle, als auch optische Vergleichsverfahren wie die Kreuzkorrelation zur Anwendung.

Schwierig ist dies, wenn das Modell erst noch aus Punkt-zu-Punkt-Korrespondenzen gewonnen werden muss, die jedoch ohne ein Modell nur unvollständig und mit vielen Falschzuordnungen bestimmbar sind. Der RANSAC-Algorithmus ist ein Verfahren zur robusten Parameterbestimmung und löst das Problem auf verblüffend einfache Weise.

Das Programm *StageDesigner* wurde als Teil dieser Diplomarbeit entwickelt und beinhaltet mit STAGE ein einfaches Trackingverfahren, das im letzten Kapitel vorgestellt wurde. Im Vergleich mit dem System der Eidgenössischen Technischen Hochschule in Lausanne, welches in vielen Punkten als Vorbild diente, wurden Vor- und Nachteile des einfacheren Ansatzes von STAGE erläutert.

Vielleicht kann man heute noch nicht davon sprechen, dem Computer das *Sehen* beigebracht zu haben, denn dazu gehört ein Verständnis des Gesehenen, das über das Vermessen von Bild- und Raumkoordinaten hinausgeht. Dem Ziel, Augmented Reality-Anwendungen praktikabel zu machen, ist man jedoch schon sehr nahe gekommen. Ein Blick auf aktuelle Trends in der Entwicklung der Unterhaltungselektronik verheißt eine rosige Zukunft. Mit millionenfach verkauften Kameras in Mobiltelefonen, billig verfügbaren „Webcams“ und leistungsfähigen Videokameras ist das digitale Auge allgegenwärtig. Vom Trend zu programmierbaren Grafikkarten und mehrfädigen Prozessorkernen profitieren die hervorragend parallelisierbaren Algorithmen der Bildverarbeitung ganz besonders. Es ist gut möglich und wahrscheinlich, dass eine Kamera bald ganz selbstverständlich als Eingabegerät wie jedes andere gehandhabt wird und diese Technik allgemeine Anwendung findet.

Bei so viel Euphorie sollte nicht vergessen werden zu erwähnen, dass die bisher erfolgreichsten Anwendungen militärischer Natur waren. Besonders die computergestützte Auswertung der Bilder von Spionagesatelliten und die Steuerung von Marschflugkörpern und anderen Waffensystemen sind seit langem Gegenstand intensiver Forschung, die natürlich unter Ausschluss der Öffentlichkeit stattfindet. So ist es schwer, eine Aussage über den tatsächlichen Stand der Forschung zu machen. Es bleibt zu hoffen, dass diesem hochinteressanten Gebiet durch weiteres Voranschreiten der zivilen Wissenschaft ein positives Bild erhalten bleibt.

Literaturverzeichnis

[Bar02] Bartoli, Adrien: The Geometry of Dynamic Scenes - On Coplanar and Convergent Linear Motions Embedded in 3D Static Scenes. In: *In Proceedings of the Thirteenth British Machine Vision Conference* Bd. I, 2002, S. 394–403. – Cardiff, UK.

[DD92] DeMenthon, Daniel ; Davis, Larry S.: Model-Based Object Pose in 25 Lines of Code. In: *European Conference on Computer Vision*, 1992, S. 335–343. – `citeseer.ist.psu.edu/dementhon95modelbased.html`

[FA91] Freeman, W. T. ; Adelson, E. H.: The design and use of steerable filters. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 13 (1991), Nr. 9, 891–906. `citeseer.ist.psu.edu/freeman91design.html`

[GCH+02] Gibson, Simon ; Cook, Jonathan ; Howard, Toby ; Hubbold, Roger J. ; Oram, Daniel: Accurate Camera Calibration for Off-line, Video-Based Augmented Reality. In: *ISMAR*, IEEE Computer Society, 2002. – ISBN 0–7695–1781–1, S. 37–46

[Har95] Hartley, R. I.: In defence of the 8-point algorithm. In: *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*. Washington, DC, USA : IEEE Computer Society, 1995. – ISBN 0–8186–7042–8, S. 1064. – `users.rsise.anu.edu.au/~hartley/Papers/fundamental/ICCV-final/fundamental.pdf`

[HS88] Harris, Chris ; Stevens, Mike: A Combined Corner and Edge Detector. In: *Proceedings of The Fourth Alvey Vision Conference*. Manchester, 1988, S. 147–151

[HS97] Heikkilä, Janne ; Silven, Olli: A Four-step Camera Calibration Procedure with Implicit Image Correction. In: *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA : IEEE Computer Society, 1997. – ISBN 0–8186–7822–4, S. 1106. – `http://www.ee.oulu.fi/~jth/doc/`

[HZ00] Hartley, R. I. ; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Erstausgabe. Cambridge University Press, ISBN: 0521623049, 2000

[Low03] Lowe, D.: *Distinctive image features from scale-invariant keypoints*. `citeseer.ist.psu.edu/lowe04distinctive.html`. Version: 2003

[LPF04] Lepetit, V. ; Pilet, J. ; Fua, P.: Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation. In: *Conference on Computer Vision and Pattern Recognition, Washington, DC*, 2004. – `cvlab.epfl.ch/publications/index.html`

[LVTF03] Lepetit, V. ; Vacchetti, L. ; Thalmann, D. ; Fua, P.: Fully Automated and Stable Registration for Augmented Reality Applications. In: *International Symposium on Mixed and Augmented Reality, Tokyo, Japan*, 2003. – `cvlab.epfl.ch/publications/index.html`

[PTVF92] Press, William H. ; Teukolsky, Saul A. ; Vetterling, William T. ; Flannery, Brian P.: *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA : Cambridge University Press, 1992. – ISBN 0521437148

[PZM02] Peters, Gabriele ; Zitova, Barbara ; Malsburg, Christoph von d.: How to Measure the Pose Robustness of Object Views. In: *Image and Vision Computing* 20 (2002), Nr. 4, 249-256. `http://ls7-www.cs.uni-dortmund.de/~peters/publication_sources/ivc02.pdf`

[QL99] Quan, Long ; Lan, Zhong-Dan: Linear N-Point Camera Pose Determination. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), Nr. 8, 774-780. `citeseer.ist.psu.edu/quan99linear.html`

[RDLW95] Ravela, S. ; Draper, B. ; Lim, J. ; Weiss, R.: Adaptive Tracking and Model Registration Across Distinct Aspects. In: *IROS95*, 1995. – `citeseer.ist.psu.edu/article/ravela95adaptive.html`

[Ros03] Rosenhahn, B.: *Pose Estimation Revisited*, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, Diss., 2003. `http://www.ks.informatik.uni-kiel.de/~vision/doc/Dissertationen/Bodo_Rosenhahn/tr0308.pdf`. – Online–Ressource

[SF95] Simoncelli, E. P. ; Farid, H.: Steerable wedge filters. In: *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*. Washington, DC, USA : IEEE Computer Society, 1995. – ISBN 0–8186–7042–8, S. 189. – `www.cs.dartmouth.edu/farid/publications/iccv95.html`

[SFZ00] Simon, G. ; Fitzgibbon, A. ; Zisserman, A.: Markerless Tracking using Planar Structures in the Scene. In: *Proc. International Symposium on Augmented Reality*, 120–128

[ST94] Shi, Jianbo ; Tomasi, Carlo: Good Features to Track. In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, S. 593 – 600

[Stu00] Sturm, Peter: Algorithms for Plane-Based Pose Estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, 2000, S. 1010–1017. – `perception.inrialpes.fr/Publications/2000/Stu00b`

[TMHF00] Triggs, Bill ; McLauchlan, Philip ; Hartley, Richard ; Fitzgibbon, Andrew: Bundle Adjustment – A Modern Synthesis. Version: 2000. `citeseer.ist.psu.edu/triggs00bundle.html`. In: Triggs, W. (Hrsg.) ; Zisserman, A. (Hrsg.) ; Szeliski, R. (Hrsg.): *Vision Algorithms: Theory and Practice*. Springer Verlag (LNCS), 298–375

[Ull79] Ullman, S.: *The Interpretation of Visual Motion*. Cambridge, MA : MIT Press, 1979. – ISBN 0–262–21007–X

[VLF03] Vacchetti, L. ; Lepetit, V. ; Fua, P.: Fusing Online and Offline Information for Stable 3D Tracking in Real-Time. In: *CVPR (2)*, IEEE Computer Society, 2003. – ISBN 0–7695–1900–8, S. 241–248

[VLF04a] Vacchetti, L. ; Lepetit, V. ; Fua, P.: Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. In: *International Symposium on Mixed and Augmented Reality, Arlington, VA,*, 2004. – `cvlab.epfl.ch/publications/index.html`

[VLF04b] Vacchetti, L. ; Lepetit, V. ; Fua, P.: Stable Real-Time 3D Tracking Using Online and Offline Information. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2004), Nr. 10, S. 1385–1391

[Wik05] Wikipedia: *Camera obscura — Wikipedia*. `de.wikipedia.org/wiki/Lochkamera`. Version: 2005

[XZ96] Xu, Gang ; Zhang, Zhengyou: *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Norwell, MA, USA : Kluwer Academic Publishers, 1996. – ISBN 0792341996

[Zha95] Zhang, Zhengyou: Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. Version: 1995. `citeseer.ist.psu.edu/zhang97parameter.html` (RR-2676). – Forschungsbericht. – Online–Ressource. – 44 S

[Zha96] Zhang, Zhengyou: Determining the Epipolar Geometry and its Uncertainty: A Review. Version: 1996. `citeseer.ist.psu.edu/zhang98determining.html`. Sophia-Antipolis Cedex, France, 1996 (2927). – Forschungsbericht. – Online–Ressource