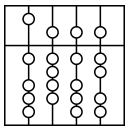


Institut für Informatik der
Technischen Universität München

Modellgetriebene Verfolgung formvariabler Objekte in Videobildfolgen

Dissertation

Christoph Hansen

Institut für Informatik
der Technischen Universität München
Lehrstuhl Univ.-Prof. Dr. B. Radig

Modellgetriebene Verfolgung formvariabler Objekte in Videobildfolgen

Christoph Hansen

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Alois Knoll

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Bernd Radig
2. Univ.-Prof. Dr. Helmut Mayer,
   Universität der Bundeswehr München

Die Dissertation wurde am 13.03.2002 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.06.2002 angenommen.

Danksagung

Ohne eine Vielzahl glücklicher Umstände wäre mir die Anfertigung dieser Arbeit nicht möglich gewesen. Mein Dank gebührt daher einer ganzen Reihe von Personen, die direkt oder indirekt zu deren Gelingen beigetragen haben.

Zunächst möchte ich mich bei meinem Doktorvater Prof. Dr. Bernd Radig für die Betreuung der Arbeit bedanken. Er gab mir die Möglichkeit, frei und eigenverantwortlich zu arbeiten und verstand es, stets für ein angenehmes Arbeitsklima und zwischenmenschliches Verhältnis zu sorgen. Prof. Dr. Helmut Mayer danke ich herzlich für die Übernahme des Zweitgutachtens. Trotz eines vollen Terminkalenders fand er die Zeit für ausführliche Diskussionen und hat in der Endphase des Zusammenschreibens durch viele detaillierte Ratschläge mitgeholfen, die Arbeit abzurunden.

Bedanken möchte ich mich weiterhin bei den Personen, mit denen ich während meiner täglichen Arbeit zu tun hatte und die mir gerade auch in der Anfangszeit behilflich waren. Nennen möchte ich insbesondere Dr. Olaf Munkelt, der mir 1998 den Quereinstieg in die Informatik ermöglichte, sowie Dr. Christof Ridder und David Hansel, die mich in die umfangreiche Software STABIL++ einführten und für Fragen jederzeit ein offenes Ohr hatten. Erwähnen möchte ich auch Ivan Laptev, durch den ich den Zugang zu flexiblen Konturmodellen gefunden habe. Den Jungs vom Lehrstuhl für Informatik IX danke ich für die zahlreichen fachlichen und nichtfachlichen Gespräche, Diskussionen und Unternehmungen. Insbesondere danke ich Dr. Michael Roth und Robert Hanek für die kritische Durchsicht der Arbeit sowie Michael Beetz, Ph.D., für ordnende Hinweise und Ratschläge.

Zu Dank verpflichtet bin ich auch einigen Personen, mit denen ich durch meine außeruniversitären, musikalischen Aktivitäten zu tun habe. Zusammen mit Nicole Marischka konnte ich immer wieder in eine ganz andere Welt eintauchen und bekam den Kopf so wieder frei für neue Ideen. Durch sie kam auch der Kontakt zu Dr. Hanno Boekhoff zustande, der mich in vielen Coaching-Gesprächen immer wieder angetrieben und ermutigt hat und mich so auf den richtigen Weg brachte.

Bedanken möchte ich mich auch bei meinen Eltern, die mir ein Studium ermöglicht und finanziert haben und damit den Grundstein für diese Arbeit legten.

Letztendlich danke ich natürlich auch noch mir selbst, ohne den diese Arbeit so wohl nicht entstanden wäre ... ;-)

Kurzfassung

Bei der automatisierten Analyse von menschlichen Bewegungen der Extremitäten und Gesichtszüge mittels videobasierter Techniken kommt der robusten Merkmalsextraktion eine Schlüsselrolle zu. Die zum Einsatz kommenden Methoden zur Bildsegmentierung müssen dabei möglichst unabhängig von den Umgebungsbedingungen sein, um so eine zuverlässige Interpretation der Bilddaten zu gewährleisten.

In dieser Arbeit wird ein System vorgestellt, mit dem durch die Verwendung unterschiedlicher Merkmale formvariable Objekte in Videobildfolgen automatisch verfolgt und deren Bewegungen dreidimensional vermessen werden können. Neben der Berücksichtigung von Farbinformationen wird insbesondere auf die kantenbasierte Merkmalsextraktion eingegangen.

Die Leistungsfähigkeit des Systems wird durch den Einsatz zur Detektion menschlicher Bewegungen in Videosequenzen demonstriert. Das dem System zugrunde liegende generische 3D-Objektmodell dient dabei zur Modellierung des menschlichen Körpers aus mehreren Objektmodellteilen. Zur genaueren Beschreibung werden die einzelnen Objektmodellteile durch Merkmale näher charakterisiert, wodurch gleichzeitig die bei der Bildsegmentierung anzuwendenden Verfahren festgelegt werden. Zur Modellierung deformierbarer Konturen werden bei der kantenbasierten Merkmalsextraktion statistische Punktverteilungsmodelle verwendet. Bei diesen ergibt sich die Beschränkung der Formvariation in natürlicher Weise aus einem Trainingsdatensatz.

Die entwickelten kantenbasierten Methoden werden anhand von Beispielen mit der Bildauswertung durch adaptive Farbklassifikation verglichen. Es wird gezeigt, wie die Detektion der für den Menschen charakteristischen Kopf-Schulter Partie dazu verwendet werden kann, um mit einem kalibrierten Stereokamerasystem Personen zu verfolgen und dreidimensionale Trajektorien für deren Bewegung zu erhalten. Zur genaueren Lokalisation der Gesichtszüge im Bild wird ein mehrteiliges Konturmodell für das Gesicht verwendet, das aus einer hierfür generierten Datenbank erstellt wurde. Das Modell wird für die Verfolgung der Gesichtszüge in Bildsequenzen eingesetzt, und es wird gezeigt, wie der zeitliche Verlauf der Modellparameter dazu verwendet werden kann, um verschiedene Mimiken zu charakterisieren. Die zusätzliche Berücksichtigung von Grauwertinformationen in einem heuristischen Modell führt hierbei zu einer verbesserten Lokalisation im Vergleich zum reinen Kantenmodell.

Das vorgestellte System erlaubt die robuste Verfolgung menschlicher Bewegungen in Bildfolgen. Es eröffnet die Möglichkeit einer Vielzahl an Anwendungen, bei denen die automatisierte Interpretation von Gestik und Mimik erforderlich ist, etwa im Bereich der Mensch-Maschine Kommunikation oder auch in medizinischen Anwendungen.

Inhaltsverzeichnis

„Okay, let´s do it!“

- Rainer Bielfeldt -

The Big Show, 1998

1 Einleitung

Bei der Entwicklung von intelligenten Systemen zum Erfassen und Verstehen der Umwelt spielt die Auswertung visueller Informationen eine entscheidende Rolle. Die Einsatzmöglichkeiten der verwendeten videobasierten Techniken sind vielfältig und reichen von einfachen industriellen Vermessungsaufgaben bis hin zu Problemstellungen, die die Auswertung hoch komplexer Szenen erfordern, wie dies beispielsweise bei der Entwicklung autonomer mobiler Roboter oder Systemen zur multimodalen Mensch-Maschine Interaktion der Fall ist.

Während sich der Einsatz von Videotechnik bis vor einigen Jahren noch auf die reine Aufzeichnung, Speicherung und Wiedergabe von Einzelbildern und Bildsequenzen beschränkte, werden durch die zunehmende Rechenleistung heutiger Computer auch Anwendungen möglich, bei denen die Auswertung des Bildmaterials nicht durch einen menschlichen Beobachter, sondern automatisiert durch einen Rechner erfolgt. Durch die rasanten Entwicklungen beispielsweise im Bereich der Bildeinzugskarten[1] kommen die eingesetzten Systeme dabei zunehmend mit aus Standardkomponenten aufgebauter Hardware aus. Ziel der digitalen Bildverarbeitung ist es, relevante Informationen aus den Bildern zu extrahieren, um so eine Interpretation der aufgenommenen Szene zu ermöglichen.

In dieser Arbeit wird ein modellbasiertes System zur videobasierten Detektion formvariabler Objekte in Videobildfolgen vorgestellt, das durch die Verwendung verschiedener Merkmale eine robuste Verfolgung der Objekte ermöglicht. Das System wird eingesetzt, um menschliche Bewegungen in Videobildfolgen zu vermessen. Neben der Extraktion von Farbinformationen aus den Bilddaten liegt der Schwerpunkt der Arbeit insbesondere auf der von der Beleuchtung weitgehend unabhängigen Bildsegmentierung mit kantenbasierten Methoden.

[1] engl. *Framegrabber*

1.1 Thematischer Kontext

Die vorliegende Arbeit ist im Kontext des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Graduiertenkollegs *Sprache, Mimik und Gestik im Kontext technischer Informationssysteme* entstanden. An diesem sind verschiedene Institute der *Technischen Universität* (TU) sowie der *Ludwig-Maximilians-Universität* (LMU) München beteiligt. Ziel des Kollegs ist die Untersuchung menschlicher Interaktion mit technischen Informationssystemen durch verbale und nonverbale Ausdrucksformen, um auf Basis der gewonnenen Erkenntnisse innovative Anwendungskonzepte und Schnittstellen zu realisieren. Die Integration visueller Informationen erfordert hierbei eine schritthaltende, automatische Verfolgung und dynamische Analyse von Körpergestik und Gesichtsmimik.

Als Testumgebung für einen Teil der in dieser Arbeit vorgestellten Modelle und Algorithmen diente das am *Bayerischen Forschungszentrum für Wissensbasierte Systeme* (FORWISS) entstandene System STABIL++[2]. Es wurde mit dem Ziel entwickelt, eine automatisierte Detektion und Verfolgung von Objekten in Videobildfolgen zu ermöglichen. Die generische Systemarchitektur erlaubt die Verfolgung beliebig aufgebauter artikularer Objekte mit einem kalibrierten Stereoaufbau[3]. Die Modellierung des Objektmodells und die restriktionsgesteuerte Korrespondenzsuche während des Interpretationsprozesses erfolgt durchgehend in 3D.

Mit der am Lehrstuhl für Informatik IX der *Technischen Universität München* entwickelten Software HALCON [Hal02] stand darüber hinaus eine umfassende Bibliothek an elementaren Bildverarbeitungsoperatoren zur Verfügung, deren Datenstrukturen auch innerhalb von STABIL++ verwendet werden.

1.1.1 Beobachtung menschlicher Bewegungen

Die *automatische Vermessung und Interpretation menschlicher Bewegungen und Verhaltensweisen* bildet die Grundlage für eine Vielzahl von Anwendungen. Der Einsatz von Kameras als Sensoren hat gegenüber anderen Verfahren den Vorteil, dass die Messung berührungslos und weitestgehend nicht-invasiv erfolgt, so dass die beobachtete Person frei agieren kann und in ihrer Bewegungsfreiheit nicht eingeschränkt wird. Die Anwendung videobasierter Systeme zur Beobachtung von Bewegungen des menschlichen Körpers reicht von der Bestimmung der Position und der Verfolgung einzelner oder mehrerer Personen bis hin zur hochgenauen Erfassung von Bewegungsabläufen einzelner Gliedmaßen oder der Gesichtszüge. Neben der reinen Vermessung gewinnt

[2] **S**ystem for **T**racking **A**rticulated O**b**jects Using **I**mage-Based 3D **L**ocalization Implemented in C++

[3] Die Anzahl der verwendeten Kameras ist beliebig und prinzipiell nicht auf zwei beschränkt.

dabei auch die automatisierte Interpretation der gewonnen Daten im Hinblick auf Gestik und Mimik zunehmend an Bedeutung.

Bei Systemen, die dem Benutzer Bedienkonzepte für einen interaktiven Zugang zu Maschinen bieten, kommt der Interpretation von Gestik und Mimik der aufgenommenen Person eine Schlüsselrolle zu. Bei der Mensch-Maschine-Interaktion (MMI) zur Steuerung eines Gerätes z.B. durch Handzeichen kann so eine natürliche und auch für den Laien intuitive Kommunikation erreicht werden. Aber auch beispielsweise beim Einsatz von virtuellen Charakteren als elektronische Verkaufsassistenten kann durch die automatisierte Mimikerkennung des Kunden und die Synchronisation der Avatar-Animation mit entsprechenden Sprachausgaben eine realitätsgetreue Situation nachempfunden werden.

Im Bereich der Sicherheitstechnik kommen zur Zeit häufig Videomanagementsysteme mit einer Vielzahl an Kameras zum Einsatz, mit denen ganze Gebäudekomplexe überwacht werden können. Die Auswertung des Bildmaterials erfolgt entweder online durch ständige Beobachtung durch das Sicherheitspersonal oder aber durch die nachträgliche visuelle Analyse aufgezeichneter Sequenzen im Schadensfall. Die Verfolgung verdächtiger Personen mit Hilfe schwenk-/neigbarer Dome-Kameras geschieht manuell von einem Bedienplatz aus oder automatisch durch festgelegte Kamerabewegungen. Durch die automatische Detektion von Personen im Bild kann die Steuerung der Kamera automatisiert werden und so zu einer Entlastung des Wachpersonals beitragen. Die Entwicklung intelligenter Softwaremodule zur automatisierten Szenenanalyse eröffnet darüber hinaus die Möglichkeit, durch das Erkennen verdächtiger Situationen rechtzeitig Präventivmaßnahmen bei der Verbrechensbekämpfung ergreifen zu können, um so Straftaten zu vermeiden.

1.1.2 Modellgetriebene Bildinterpretation

Dem Menschen fällt es leicht, Gegenstände zu erkennen, Personen voneinander zu unterscheiden oder beispielsweise Mimik und Gestik bei der Kommunikation zu deuten. Er greift bei der Analyse auf vielfältiges Wissen zurück, das er sich im Lauf seines Lebens angeeignet hat und das ihm eine sinnvolle Interpretation seiner Wahrnehmung ermöglicht. Die Semantik einer Beobachtung ist dabei stets kontextbezogen. Um eine automatische Objektidentifikation und die Interpretation einer aufgenommenen Szene in einem Gesamtzusammenhang auch für einen Rechner zu ermöglichen, müssen daher zusätzliche Informationen über die im Bild zu suchenden Objekte, deren Umgebung und Interaktionsmöglichkeiten verwendet werden.

Diesen Überlegungen trägt der modellbasierte Ansatz zur Bildanalyse Rechnung, bei dem während der Bildsegmentierung und der anschließenden Interpretation der extrahierten Merkmale auf eine Reihe von Modellvorstellungen zurückgegriffen wird (siehe Bild 1.1).

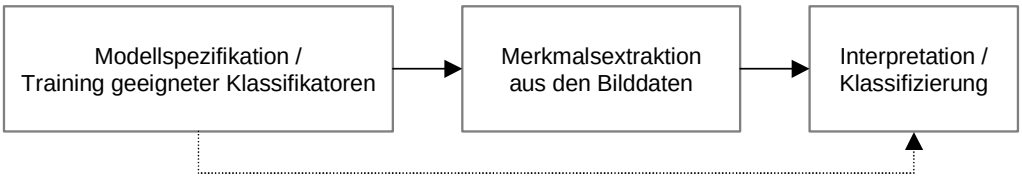


Bild 1.1: ***Prinzipieller Ablauf der modellbasierten Bildinterpretation****.*

Die Wissensbasis enthält alle a priori Informationen, die für den Interpretationsprozess relevant sind. Den zentralen Bestandteil eines modellbasierten Systems bildet die Beschreibung des im Bild zu suchenden Objektes in einem Objektmodell. Das Objekt wird durch die Definition geeigneter Merkmale charakterisiert, durch die implizit auch die Algorithmen für die Bildsegmentierung festgelegt werden.

Je nach Komplexität der Anwendung wird die Modellierung um zusätzliche Modelle erweitert, um so zu einer möglichst realistischen Beschreibung der Welt zu gelangen. Neben dem eigentlich im Bild zu suchenden Objekt werden oft auch Elemente der Umgebung, wie z.B. Inventargegenstände berücksichtigt. Für exakte Vermessungsaufgaben müssen darüber hinaus beispielsweise die Abbildungseigenschaften der verwendeten Optik berücksichtigt und der Abbildungsprozess durch ein geeignetes Kameramodell beschrieben werden. Neben den erwähnten Modellen zur Beschreibung des zu suchenden Objektes und dessen Einbettung in seine Umgebung wird der Interpretationsprozess weiterhin bestimmt durch das Wissen um die Semantik der extrahierten Informationen. Die Interpretation der Messung erfolgt meist mit einem zuvor trainierten Klassifikator.

1.1.3 Formvariable Objekte

Die Form *starrer* Objekte wie Tische oder Stühle ist fest vorgegeben und kann durch den Aufbau geeigneter CAD[4]-Modelle hinreichend genau beschrieben werden. Das Aussehen eines solchen Objektes im Bild hängt daher in erster Linie von seiner relativen Lage in Bezug auf die Kamera ab. Verschiedene Ansichten können beispielsweise durch Triangulation der Gaußschen Sphäre vor der eigentlichen Bildinterpretation berechnet werden, so dass während des Interpretationsprozesses darauf zurückgegriffen werden kann. Ein solcher ansichtszentrierte Ansatz der Bildinterpretation ist z.B. in der Arbeit von Munkelt [Mun94] beschrieben.

Im Gegensatz dazu müssen bei der Vermessung *nicht-starrer* Objekte eine Vielzahl von Formvariationen berücksichtigt werden. Diese legen die Verwendung eines komplexeren Modells und die Auswertung der Bilder durch einen objektzentrierten Ansatz nahe.

[4] Computer Aided Design

Die Bewegungen und Formvariationen nicht-starrer Objekte lassen sich allgemein einteilen in:

- *artikulare* Bewegungen / Formvariationen
- *flexible*[5] Bewegungen / Formvariationen

Artikulare Objekte sind aus mehreren in sich starren Teilen aufgebaut, die über Gelenke miteinander verbunden sind. Zusätzlich zu einer Translation und Rotation des gesamten Objektes können die einzelnen Objektmodellteile rigide Bewegungen ausführen und sich relativ zueinander bewegen. Die sich hieraus ergebenden möglichen Konfigurationen im Raum der Modellparameter können durch die Anwendung geeigneter Restriktionen beispielsweise für die zulässigen Gelenkwinkel eingeschränkt werden. Während der Korrespondenzsuche, bei der versucht wird, eine gültige Zuordnung der segmentierten Bildmerkmale zum Objektmodell zu finden, kann so eine vollständige Traversierung des Interpretationsbaumes vermieden werden (vgl. Abschnitt 3.4.2). Die Modellierung durch ein artikulares Objektmodell wird in dieser Arbeit für die Beschreibung des menschlichen Skeletts verwendet.

Bei der Verfolgung von Personensilhouetten oder bei detaillierterer Vermessung z.B. der Hände oder der Gesichtszüge müssen zusätzlich Konturvariationen berücksichtigt werden, die sich aus einer Formveränderung der Objektmodellteile selbst ergeben. Das Aussehen der Instanzen[6] einer bestimmten Objektklasse ist für verschiedene Individuen zwar ähnlich, die genaue Form variiert aber zwischen unterschiedlichen Personen und darüber hinaus auch zwischen verschiedenen Aufnahmen derselben Person. Die Bildsegmentierung mit flexiblen Konturmodellen wird in dieser Arbeit zur genauen Bestimmung der Gesichtszüge verwendet, und um Personen anhand ihrer Silhouette im Bild zu verfolgen.

1.2 Zielsetzung

Um zu einer umfassenden und robusten Interpretation einer Szene zu gelangen, muss ein System zur automatisierten Erfassung und Deutung menschlicher Bewegungen und Verhaltensweisen die Realität möglichst exakt abbilden. Die Detailtreue der Modellierung, die Komplexität der Bildsegmentierung und die Methoden zur Interpretation der Daten richten sich dabei nach den konkreten Anforderungen. Neben der Modellspezifikation und der Definition geeigneter Merkmale zu dessen Charakterisierung werden durch die Definition verschiedenster Restriktionen meist Annahmen über die Umge-

[5] Der Begriff *flexibel* wird in dieser Arbeit synonym zu dem in der Literatur ebenfalls häufig verwendeten Ausdruck *deformierbar* verwendet (engl.: *deformable models*).

[6] Der Begriff *Instanz* meint hier das in einem Bild gefundene Objekt.

bung (Beleuchtung, Hintergrundvariation, Inventar, etc.), die Anzahl der in der Szene befindlichen Objekte sowie deren mögliche Haltungen und Bewegungen gemacht.

Im Labor sind die Umgebungsbedingungen meist kontrollierbar, und die Vermessung beispielsweise der Bewegung der Extremitäten kann vor definiertem Hintergrund oder in eng anliegender Spezialkleidung erfolgen. Bei einfachen Überwachungsaufgaben, bei denen lediglich die Anwesenheit von Personen detektiert werden soll, reicht es häufig aus, eine Segmentierung des Bildes in Vordergrund und Hintergrund durchzuführen und die Detektion anhand von Größe und Form der Vordergrundregion vorzunehmen. Bei größeren Variationsmöglichkeiten der Umgebungsbedingungen und zunehmender Komplexität der Messaufgabe muss jedoch auf detailliertere Modelle und aufwändigere Verfahren zur Segmentierung zurückgegriffen werden. Darüber hinaus müssen entsprechende Klassifikatoren zur Interpretation der Daten trainiert werden.

Insbesondere bei der Vermessung menschlicher Bewegungen stellt die zuverlässige Bildinterpretation immer noch eine große Herausforderung dar. Beispielsweise versagt die vielfach angewendete Detektion hautfarbener Bereiche im Bild bei starken Schwankungen der Umgebungsbeleuchtung oder wenn die Person sich umdreht. Das Aussehen im Bild variiert darüber hinaus zwischen verschiedenen Personen und wird bestimmt durch die vielen Freiheitsgrade bei den Bewegungsmöglichkeiten im Dreidimensionalen.

Ziel dieser Arbeit ist es daher, ein möglichst robustes System zur automatisierten Beobachtung formvariabler Objekte zu entwickeln, das eine zuverlässige dreidimensionale Verfolgung in Bildsequenzen ermöglicht. Um von Umgebungsbedingungen wie z.B. der Beleuchtung unabhängig zu werden, ist dazu die Auswertung *mehrerer* sich ergänzender Objekteigenschaften wie Farbe, Umriss, Textur etc. notwendig. Im Bereich der Sicherheitstechnik werden hierdurch beispielsweise anspruchsvolle Anwendungen im Außenbereich möglich, bei denen die Umgebungsbedingungen oft starken Schwankungen unterworfen sind. Durch die Wahl geeigneter Merkmale und Beschreibungen zur Charakterisierung von Personen können bei automatisierter Detektion und Verfolgung auch statistische Daten zur Beschreibung des Aussehens und der Verhaltensweise ohne Benutzer-Interaktion gewonnen werden. Aus diesen kann beispielsweise eine Datenbank generiert werden, die auch nach einem längeren Zeitraum durch Vergleich mit den hinterlegten Daten eine Wiedererkennung möglich macht.

Die Objektmodellierung sowie die verwendeten Algorithmen zur Bildsegmentierung sollten generisch gehalten werden, damit das System flexibel an neue Problemstellungen angepasst und in verschiedenen Bereichen verwendet werden kann. Um das System beispielsweise in interaktiven MMI-Anwendungen sinnvoll einsetzen zu können, sollte die Bildinterpretation darüber hinaus schritthaltend in Echtzeit erfolgen. Weiterhin sollte auf spezielle Hardware verzichtet und das System aus preisgünstigen Standardkomponenten aufgebaut werden.

1.3 Inhalt der Arbeit

Gegenstand dieser Arbeit ist die merkmalsbasierte Detektion und Verfolgung formvariabler Objekte in Videobildfolgen. Es wird ein System vorgestellt, das durch die Verwendung von kanten- und farbbasierten Segmentiermethoden eine robuste Verfolgung der Objekte ermöglicht. Die Leistungsfähigkeit des Systems wird durch den Einsatz zur Beobachtung menschlicher Bewegungen demonstriert.

Der Begriff *Detektion* meint in diesem Zusammenhang das Auffinden von Objektmodellinstanzen im Bild. Er wird in dieser Arbeit unterschieden von dem in der Literatur häufig synonym verwendeten Ausdruck der *Identifikation* von Personen und der *Interpretation* ihrer Mimik und Gestik.

Kapitelübersicht

Die Erfassung und Interpretation menschlicher Bewegungen und Verhaltensweisen ist ein komplexes Themengebiet mit vielen Teilbereichen. ***Kapitel 2*** gibt zunächst einen Literaturüberblick über verwandte Arbeiten. In der vorliegenden Arbeit wird zur Modellierung des Menschen ein hierarchisches Objektmodell verwendet, dessen einzelne Teile zur genaueren Beschreibung mit verschiedenen Merkmalen versehen werden können. Durch diese sind auch die Methoden bestimmt, mit denen die Bildsegmentierung erfolgt. Die Übersicht wurde dementsprechend in die Teilgebiete *Menschmodelle*, *Modellierung flexibler Konturen, Bildsegmentierung und Objektverfolgung* und *Anwendungen* aufgeteilt.

Das System STABIL++ wurde zum Testen der entwickelten Modelle und Algorithmen um geeignete Klassen und Methoden erweitert. In ***Kapitel 3*** wird das Gesamtsystem vorgestellt, und es werden die für das Verständnis dieser Arbeit notwendigen Systembestandteile näher erläutert. Nach einem kurzen Überblick über mögliche Anwendungen und Einsatzbereiche wird insbesondere auf die Modellierung des Objektmodells, seine Beschreibung durch Merkmale (*features*) und den restriktionsgesteuerten Interpretationsprozess bei der Korrenspondenzsuche eingegangen. Der abschließende Abschnitt zeigt die Grenzen der bislang vorwiegend verwendeten Segmentierung durch Farbklassifikation auf. Dies führt auf die Notwendigkeit zusätzlicher Merkmale, um eine robustere, von den Beleuchtungsbedingungen weitgehend unabhängige Interpretation zu erreichen.

Die zu diesem Zweck entwickelten kantenbasierten Merkmale werden in ***Kapitel 4*** vorgestellt, das zunächst einen kurzen Überblick über die Charakterisierung von Kanten und Kantendetektoren gibt und die Anforderungen an ein Modell zur Beschreibung flexibler Konturen darstellt. Zur Erstellung der in dieser Arbeit verwendeten Punktverteilungsmodelle (*Point Distribution Models*, PDMs) wurde eine hinreichend große Anzahl an Trainingsbildern ausgewertet, aus denen die Trainingskonturen mit einem halbautomatischen Verfahren (*intelligent scissors*) extrahiert wurden. Für die Detektion von Per-

sonen zur Überwachung sicherheitsrelevanter Bereiche bietet sich die Verfolgung der charakteristischen Kopf-Schulter Partie an. Zur genaueren Lokalisation des Gesichtes im Bild beispielsweise bei der Positionierung schwenk-/neigbarer Kameras oder für die Vermessung von Gesichtsbewegungen müssen die einzelnen Gesichtszüge genauer detektiert werden. Unterschiede in der Gesichtsform zwischen verschiedenen Individuen müssen hierbei ebenso berücksichtigt werden wie Formvariationen, die sich aus der relativen Position der einzelnen Gesichtsteile z.B. bei verschiedenen Blickrichtungen oder Mimiken ergeben. Zur Erstellung eines entsprechenden repräsentativen Trainingsdatensatzes wurde eine Datenbank mit Aufnahmen verschiedener Personen erstellt. Abschließend werden die im Rahmen dieser Arbeit erstellten Modelle für die menschliche Silhouette und ein zehnteiliges Modell für das menschliche Gesicht zusammen mit entsprechenden Suchalgorithmen zur Detektion von Modellinstanzen im Bild vorgestellt.

Kapitel 5 demonstriert die Anwendung der in Kapitel 4 entwickelten Modelle und Algorithmen für die Bildinterpretation. Das Silhouettenmodell für den menschlichen Oberkörper wird innerhalb des Systems STABIL++ als kantenbasiertes Merkmal dazu verwendet, um Personen in Mono- und Stereobildfolgen zu detektieren und dreidimensional zu verfolgen. Die Resultate werden den Ergebnissen der bislang meist verwendeten Farbklassifikation gegenübergestellt und mit diesen verglichen. Weiterhin wird gezeigt, wie mit dem Gesichtsmodell Gesichter in Einzelbildern detektiert und in Bildsequenzen verfolgt werden können. Durch die Erweiterung des PDM-Modells um Grauwertinformationen wird eine genauere Lokalisation der Modellteile im Bild möglich als bei ausschließlicher Berücksichtigung von Kanteninformationen. Bei der Auswertung von Bildfolgen, in denen eine bestimmte Mimik zu sehen ist, ergibt sich im Raum der Modellparameter ein charakteristischer zeitlicher Verlauf, der die betreffende Mimik beschreibt. Die automatisierte Merkmalsextraktion ermöglicht so den Aufbau einer Wissensbasis für einen Klassifikator zur automatisierten Mimik-Interpretation. Abschließend werden Laufzeitmessungen für den Zeitbedarf bei der Modellsuche vorgestellt und die Grenzen des Verfahrens aufgezeigt.

In ***Kapitel 6*** werden die Resultate dieser Arbeit noch einmal kurz zusammengefasst, und es wird ein Ausblick auf zukünftige Weiterentwicklungen und Anwendungsgebiete gegeben.

2 Wissenschaftlicher Kontext

Die vorliegende Arbeit fällt in das Themengebiet der videobasierten Erfassung von Bewegungen formvariabler Objekte in Videobildfolgen, insbesondere der Vermessung menschlicher Bewegungen. Zu diesem komplexen Themengebiet mit vielen Teilaspekten existieren zahlreiche Arbeiten, die sich hinsichtlich der verwendeten Modelle, der Methoden zur Bildanalyse und der Interpretation der Daten unterscheiden.

In der Literatur finden sich mehrere Übersichtsartikel, die sich bezüglich der gewählten Taxonomie zur Einteilung der untersuchten Publikationen unterscheiden. Moeslund und Granum geben in [MG01] einen Überblick über die Arbeiten der letzten 20 Jahre zum Thema videobasierte Erfassung menschlicher Bewegungen. Eine Einteilung der Publikationen wird hier entsprechend dem zeitlichen Ablauf des Interpretationsprozesses (Bild 1.1) vorgenommen in: Initialisierung einschließlich Modellspezifikation, Segmentierung und Schätzung der Modellparameter, Modellrekonstruktion (*pose estimation*) sowie Erkennung einer bestimmten Haltung oder Bewegung, d.h. Klassifizierung von statischen und dynamischen Messdaten.

Eine Übersicht über mehr als 100 Publikationen bis etwa 1999 findet sich in der Arbeit von Moeslund [Moe99]. Zusätzlich zu einer kurzen Zusammenfassung jedes Papers gibt der Autor für die einzelnen Arbeiten relevante Stichwörter an und erleichtert durch zusätzliche Kommentare eine Beurteilung der vorgestellten Methode.

Einen Überblick über Arbeiten zur Beobachtung des menschlichen Körpers und Bewegungen der Hand gibt auch Gavrila [Gav99]. Die Einteilung der Arbeiten erfolgt hier in 2D-Ansätze mit und ohne explizitem Menschmodell sowie 3D-Ansätze. Gavrila gibt darüber hinaus einen kurzen Überblick über verschiedene Techniken zur Klassifikation zeitabhängiger Muster (Dynamic Time Warping, Hidden Markov Modelle, Neuronale Netze).

Zur Einordnung der vorliegenden Arbeit in den wissenschaftlichen Kontext werden in den folgenden Abschnitten thematisch verwandte Publikationen vorgestellt, wobei eine Einteilung in *Menschmodelle*, *Modellierung flexibler Konturen*, *Bildsegmentierung und Objektverfolgung* und *Anwendungen* gewählt wurde. Hierbei werden beispielhaft einige repräsentative Arbeiten vorgestellt. Für weitere Literaturangaben sei auf die oben genannten Übersichtsartikel verwiesen.

2.1 Menschmodelle

Generell kann bei der Beobachtung menschlicher Bewegungen zwischen Ansätzen mit und ohne explizitem Menschmodell unterschieden werden.

Bei Ansätzen, die nicht auf ein explizites Menschmodell zurückgreifen, führen geometrische oder texturelle Merkmale einer *region of interest* (ROI) durch statistische Auswertung oder Heuristiken direkt zu einer Interpretation der Bilder, ohne erst explizit die Haltung einer Person zu rekonstruieren. Das von Wren et al. in [WAD96] beschriebene System *Pfinder* beispielsweise modelliert das Aussehen einer Person im Bild durch 7 benachbarte Regionen[7] (*blobs*) mit einer charakteristischen räumlichen Verteilung und Farbverteilung (siehe Bild 2.1). Bei der Bildinterpretation wird zunächst durch Beobachtung der leeren Szene ein statistisches Modell für die Umgebung erstellt. Zur Personendetektion werden anschließend die einzelnen Pixel mit diesem Szenenmodell verglichen. Die ROI wird durch diejenigen Pixel gebildet, deren Mahalanobis-Distanz im Farbraum einen bestimmten Schwellwert überschreitet. Wenn die Größe der Region, in der sich das Bild verändert hat, eine gewisse Größe überschreitet, wird online ein *blob*-Modell für die Person erstellt.

Auch bei der *Analyse* menschlicher Bewegungen kann häufig auf ein explizites Menschmodell verzichtet werden. Psychologische Untersuchungen der menschlichen Wahrnehmung haben gezeigt, dass dem menschlichen Betrachter zur Identifikation eines bestimmten Bewegungsmusters häufig die Beobachtung einer Menge von Punkten genügt, ohne dass eine Zuordnung der Punkte zu bestimmten Körperteilen notwendig ist. Im Gegensatz zur strukturierten Analyse, bei der vor der Interpretation der Daten ein Objektmodell explizit rekonstruiert wird, spricht man hier auch von globaler Interpretation (engl.: *global/structured interpretation of motion*). Boyd und Little [BL97] extrahieren z.B. aus dem optischen Fluss von Bildern solcher Punktmengen (*moving light displays*, MLDs) eine Vielzahl an skalaren geometrischen Merkmalen. Durch Analyse der Phasenbeziehungen der sich ergebenden Zeitreihen ist es so möglich, Personen anhand ihres Gangs zu unterscheiden, ohne die Skelettstruktur zu rekonstruieren.

In den meisten Fällen wird jedoch auf ein explizites Menschmodell zurückgegriffen. Aufgrund der skelettartigen Struktur des menschlichen Körpers bietet sich ein artikularer Aufbau des Modells aus einzelnen starren Objektmodellteilen an, die die Körperteile repräsentieren und die durch Gelenke miteinander verbunden sind. Zur Modellierung der die Knochen umgebenden Muskeln, des Fettes und der Haut wird häufig eine volumetrische Beschreibung durch einfache dreidimensionale geometrische Primitive (Zylinder, Ellipsoide, Kegelstümpfe etc.) gewählt oder die genauere Approximation der Oberfläche durch geeignete Polygone. Hierbei werden teilweise auch anatomische Daten berücksichtigt. Zusätzlich zur geometrischen Beschreibung können darüber hinaus

[7] zusammenhängende Menge von Pixeln

auch kinematische Informationen zur Modellierung von Bewegungen berücksichtigt werden.

In der Arbeit von Rohr [Roh93] wird ein Volumenmodell zur Detektion von Fußgängern in monokularen Sequenzen verwendet. Das dreidimensionale artikulare Modell besteht aus 14 Zylindern mit elliptischem Querschnitt und berücksichtigt zur Modellierung der Gehbewegung in einem kinematischen Ansatz die von Murray et al. [MDK64] erhaltenen Bewegungsdaten. Unter der Voraussetzung, dass eine zyklische Gehbewegung in einer Ebene parallel zur Kamera beobachtet wird, kann die Körperhaltung der Person, d.h. die relative Lage der einzelnen Objektmodellteile zueinander, durch einen einzigen Parameter beschrieben werden.

Für eine realistische und naturgetreue Modellierung des Menschen in der Computergrafik oder auch für Animationen von virtuellen Schauspielern (Avatare) ist eine komplexere Beschreibung und die Berücksichtigung zusätzlicher anatomischer Daten notwendig. Plänkers et al. [PFA99][PF01] verwenden ein mehrschichtiges Modell, bei dem die Detailtreue der Modellierung in jeder Ebene verfeinert wird (Bild 2.1). Die unterste Ebene wird auch hier durch ein hierarchisches Skelettmodell gebildet. Das Aussehen von Knochen, Muskeln und Fettgewebe wird durch eine Vielzahl elliptischer Metakugeln modelliert, die an den einzelnen Segmenten des Skeletts haften und anatomisch sinnvoll angeordnet werden. Jede Metakugel definiert die Quelle eines Potentialfeldes. Eine realistische Ansicht ergibt sich schließlich durch Rendering der Äquipotentialfläche $S = \{(x, y, z) \in R^3 \mid F(x, y, z) = T\}$, die sich aus der globalen Feldfunktion F ergibt.

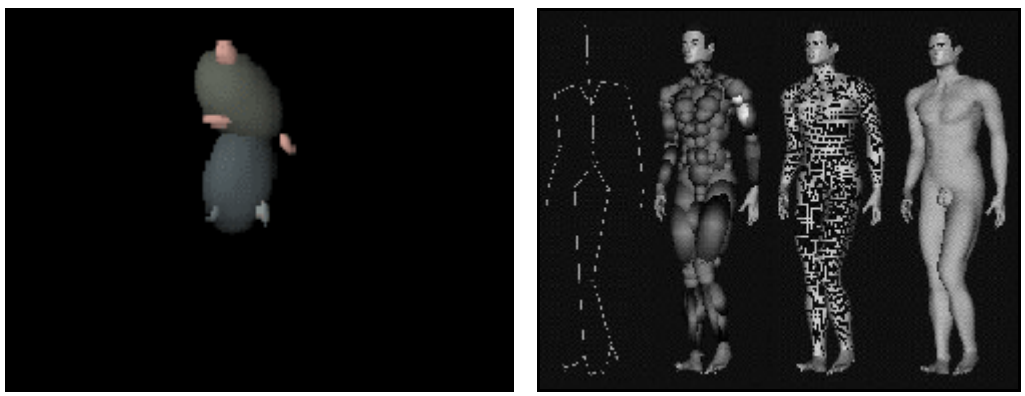


Bild 2.1: ***Menschmodelle****. Links: einfache Modellierung durch eine zusammenhängende Menge von Blobs [WAD96]. Rechts: detaillierte Modellierung durch ein vielschichtiges Modell [PF01].*

2.2 Modellierung flexibler Konturen

Insbesondere bei der Modellierung des Menschen als Ganzes durch die Beschreibung seiner Silhouette oder bei der genaueren Vermessung einzelner Körperteile, wie z.B. Hände und Gesicht, muss das verwendete Modell eine große Vielfalt an Formen berücksichtigen. Ein einfaches, starres Modell oder ein aus mehreren starren Teilen aufgebautes artikulares Objektmodell ist bei der Segmentierung dieser feineren Strukturen häufig nicht ausreichend. Bei der Bildanalyse werden statt dessen flexible Konturmodelle zur Modellbeschreibung verwendet. Konturmodelle sind Kurven oder Flächen, die sich unter dem Einfluss verschiedener Kräfte verformen können und so eine Anpassung an die Bildstrukturen erreichen.

Im Folgenden wird näher auf die Modellierung flexibler Konturen mit Snakes, Splines und die in dieser Arbeit verwendeten PDMs eingegangen. Eine gute Übersicht über die Bildsegmentierung mit flexiblen Konturmodellen geben Xu et al. in [XPP00].

2.2.1 Snakes

Die Segmentierung mit flexiblen Konturmodellen wurde erstmals durch die Einführung so genannter S*nakes* durch Kass, Witkin und Terzopoulos [KWT87] beschrieben. Der (zweidimensionale) Konturverlauf im Bild wird dabei beschrieben durch eine parametrische Kurve $X(s)=(x(s),y(s))$ mit Laufparameter $s\in[0,1]$. Im statischen Fall besteht ein Gleichgewichtszustand zwischen den inneren Kräften, die sich aus der Definition der Kontur selbst ergeben, und den äußeren Kräften, die die Wechselwirkung der Kontur mit den Bilddaten beschreiben und diese zu den Bildkanten hinziehen.

Die Gesamtenergie

$$E(X(s)) = E_{\text{int}}(X(s)) + E_{ext}(X(s)) \tag{2.1}$$

der Kontur setzt sich aus der inneren Energie $E_{int}(X(s))$ und der äußeren Energie $E_{ext}(X(s))$ zusammen. Die innere Energie ist darin durch den Ausdruck

$$E_{\text{int}}(X(s)) = \frac{1}{2}\int_0^1 \alpha(s)\left|\frac{\partial X(s)}{\partial s}\right|^2 + \beta(s)\left|\frac{\partial^2 X(s)}{\partial s^2}\right|^2 ds \tag{2.2}$$

definiert. Der erste Teil beschreibt die Elastizität der Kurve, der zweite Term modelliert ihre Steifheit. Über die zwei Terme, deren Gewichtung durch die Parameter $\alpha(s)$ und $\beta(s)$ kontrolliert wird, werden Spannung und Glattheit der Kurve erreicht.

Die externe Energie

$$E_{ext}(X(s)) = \int_0^1 P(X(s))ds \tag{2.3}$$

berücksichtigt die Wechselwirkung der Kontur mit dem Bild. *P(X(s))* ist eine aus den Bilddaten abgeleitete Potentialfunktion, die durch Gradientenbildung auf die Potentialkräfte führt. Die Funktion wird meist so gewählt, dass sie an Bildkanten Minima aufweist.

Gesucht ist eine Kurve, welche die Summe aus innerer und äußerer Energie minimiert. Die Minimierung von Gleichung (2.1) führt mit Hilfe der Variationsrechnung auf die bekannte Euler-Lagrange-Gleichung [BS96]

$$\frac{\partial}{\partial s}\left(\alpha(s)\frac{\partial X(s)}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta(s)\frac{\partial^2 X(s)}{\partial s^2}\right) = \nabla P(X(s)) \tag{2.4}$$

Gleichung (2.4) drückt das Gleichgewicht von inneren und externen Kräften im statischen Fall aus. Um zu einer dynamischen Formulierung des Problems zu kommen, wird meist ein künstlicher, von der Zeit *t* abhängender Zusatzterm eingeführt, der bei Anpassung der Kontur an die Bildstrukturen im Gleichgewicht verschwindet und so wieder auf (2.4) führt.

Alternativ bietet sich eine Ansatz an, der die verschiedenen auf die Kontur wirkenden Kräfte in der aus der Physik bekannten Newtonschen Bewegungsgleichung ($F = m\ddot{x}$) zusammenfasst und so unmittelbar auf eine zeitabhängige Formulierung führt. Die Kraft *F* enthält unter anderem Trägheits- und Dämpfungsglieder, die im Gleichgewicht wiederum verschwinden.

Mit Snakes werden insbesondere dann gute Ergebnisse erzielt, wenn die Segmentierung semi-automatisch erfolgt und durch Interaktion von Seiten des Benutzers beispielsweise die initiale Kontur definiert wird oder Regionen im Bild markiert werden, an die eine Anpassung erfolgen soll. Anwendungen finden sich vorwiegend in der Medizin beispielsweise zur Auswertung von Ultraschallaufnahmen oder von Bildern aus der Computertomographie oder Kernspinresonanz [XP98][IT96] (siehe Bild 2.2), aber auch z.B. zur Extraktion von Straßen aus Satellitenbildern [NFI97][LML00].

Unzureichende Segmentierergebnisse ergeben sich jedoch häufig, wenn die initiale Kontur den tatsächlichen Kantenverlauf nur schlecht approximiert. Bei der iterativen Suche wird dann nicht der optimale Kantenverlauf gefunden, sondern die Suche fängt

sich in einem lokalen Minimum. Insbesondere starke Verformungen und Vergrößerungen der Kontur bei konkaven Kantenverläufen lassen sich häufig nicht erreichen. Eine Verbesserung ist z.B. durch die so genannten *Ziplock Snakes* möglich [NFI97]. Bei diesen erfolgt die Anpassung an die Bildstrukturen, ausgehend von zwei Startpunkten am Rand der Kontur, von aussen nach innen. Die aktiven Konturpunkte an den beiden äußeren Enden unterliegen den vom Bild herrührenden Potentialkräften, während für die dazwischen liegenden passiven Punkte die kräftefreie Euler-Lagrange Gleichung gelöst wird. Der Term auf der rechten Seite von Gleichung (2.4) verschwindet in diesem Fall.

Ein weiterer Nachteil von Snakes ist die fehlende Möglichkeit, zusätzliches a priori Wissen explizit in das Modell mit aufzunehmen. Oft werden Snakes daher um künstlich eingeführte Kräfte erweitert. Zum Einsatz kommen Feder- oder Ballonkräfte, die die Kontur je nach Kraftrichtung zusammenschrumpfen lassen oder ausdehnen, oder auch so genannte Vulkankräfte, mit denen Punkte im Bild definiert werden können, die ein repulsives Potential haben und an die infolgedessen keine Anpassung stattfindet [KWT87][XPP00].

2.2.2 Splines

Eine Spline-Funktion *n*-ter Ordnung ist eine stückweise polynomiale Funktion, die sich aus Segmenten von Polynomen *n*-ter Ordnung zusammensetzt, so dass die Funktion an den Knotenpunkten C^{n-1} ist[8]. Splines werden meist durch eine Linearkombination so genannter B-Splines mit kompaktem Träger repräsentiert:

$$s^n(t) = \sum_{k \in Z} q_k B^n(t-k) \tag{2.5}$$

Aus dem Rechteckpuls

$$B^0(t) = \begin{cases} 1 & , \; -\tfrac{1}{2} < t < \tfrac{1}{2} \\ \tfrac{1}{2} & , \; |t| = \tfrac{1}{2} \\ 0 & , \; sonst \end{cases} \tag{2.6}$$

ergibt sich der B-Spline n-ter Ordnung

[8] d.h. die Funktion ist *(n-1)* mal stetig differenzierbar

$$B^n(t) = \underbrace{B^0 * B^0 * \cdots * B^0}_{n+1\ mal}(t) \tag{2.7}$$

durch *n*-fache Faltung [Uns99]. Im Zweidimensionalen führen Kontrollpunkte (q_{kx},q_{ky}) zur Definition einer parametrischen Kurve $(x(t),y(t))$, bei der *x* und *y* jeweils durch eine Spline-Funktion gemäß Gleichung (2.5) beschrieben werden [BI98] (siehe Bild 2.2).

Splines werden häufig auch in Kombination mit den ebenfalls in dieser Arbeit verwendeten Punktverteilungsmodellen (vgl. Abschnitt 2.2.3) eingesetzt [BH94] oder als so genannte B-Spline Snakes bzw. B-Snakes in Kombination mit Snakes [MSM90][GL97] [BHU00].

2.2.3 Point Distribution Models

Die bei den Snakes in Abschnitt 2.2.1 erwähnten Zusatzkräfte werden künstlich in das Modell aufgenommen und meist heuristisch definiert. Wünschenswert wäre jedoch eine Formulierung, in der sich die zulässigen Modellformen und die Variationsbeschränkung des Konturverlaufs explizit beschreiben lassen und beispielsweise aus einem Trainingsdatensatz gelernt werden können. Ein vielversprechender Ansatz, zusätzliches a priori Wissen in das Modell mit aufzunehmen, ist die Beschreibung der Kontur durch statistische Punktverteilungsmodelle (*point distribution models*, PDMs). Bei diesen wird der Konturverlauf durch eine Reihe von Punkten approximiert, und die Formvariation ergibt sich in einfacher Weise aus einem Trainingsdatensatz [CTC92][CT93] (siehe Bild 2.2).

Punktverteilungsmodelle können immer dann sinnvoll eingesetzt werden, wenn die Form des zu detektierenden Objektes in etwa bekannt ist und ein entsprechendes Modell trainiert werden kann. Die Einsatzgebiete reichen von Anwendungen in der Medizin, beispielsweise zur Segmentierung von Echokardiogrammbildern oder Magnet-Resonanz Aufnahmen [XP98][CHT94][CET99][DJD01], über die Verfolgung von Personen [BH94] bis hin zu industriellen Applikationen zur Qualitätskontrolle [CTC95].

Punktverteilungsmodelle werden in dieser Arbeit verwendet, um die Silhouette des menschlichen Oberkörpers und das Gesicht zu modellieren. Eine detaillierte Beschreibung der Modelle und der in dieser Arbeit verwendeten Algorithmen findet sich in Kapitel 4.

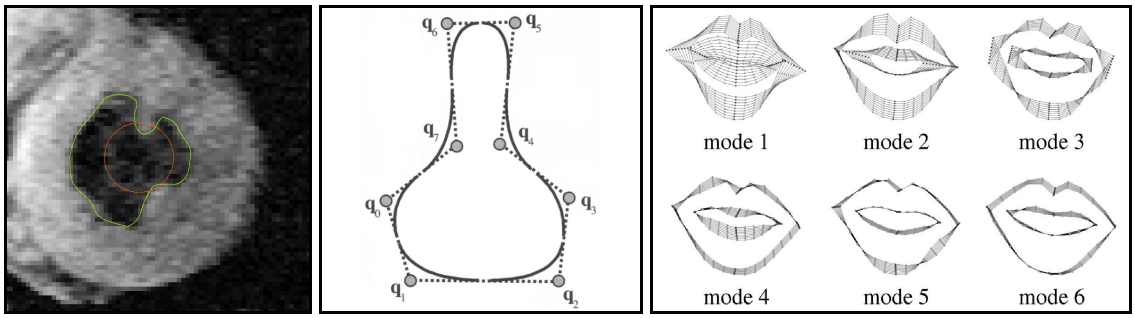


Bild 2.2: ***Modellierung flexibler Konturen****. Links: Snakes zur Segmentierung einer MR-Aufnahme [XP98]. Mitte: Kontrollpunkte zur Definition einer parametrischen Kurve mit Splines [BI98]. Rechts: Variationen der ersten 6 Moden eines PDM für die Lippen [MCB02].*

2.3 Bildsegmentierung und Objektverfolgung

Zur Vermessung von Körperhaltungen oder Bewegungen beispielsweise der Hand werden beim Einsatz nicht-optischer Verfahren häufig aufwändige elektromagnetische Sensoren eingesetzt, bei denen an den zu verfolgenden Punkten des Objektes Receiver befestigt werden, deren relative Positionen zu einem Emitter bestimmt werden können [BHG93][MHB97]. Diese auch schon kommerziell eingesetzten Systeme [Asc02] schränken jedoch wegen der für die Stromversorgung notwendigen Verkabelung die Bewegungsfreiheit der agierenden Person häufig stark ein, und die für die Messung emittierten Felder werden durch Metallgegenstände oder andere elektromagnetische Streufelder leicht gestört, wodurch es häufig zu verfälschten Messergebnissen kommt.

Visuelle Messverfahren, bei denen Kameras als Sensoren eingesetzt werden, haben demgegenüber den Vorteil, dass die Bewegungserfassung berührungslos und weitestgehend nicht-invasiv erfolgt. Eine Verkabelung entfällt und die Bewegungsfreiheit der Person bleibt erhalten. Die zur Bildsegmentierung verwendeten Methoden hängen von den konkreten Systemanforderungen und den für die Charakterisierung des Objektes benutzten Eigenschaften ab.

- *Schwellwertbildung und Hintergrundsubtraktion*

Häufig werden Verfahren eingesetzt, die sich auf die Auswertung von Lichtpunkten stützen. Bei dieser unter dem Namen *moving light displays* (MLDs) bekannten Technik werden Teile des Objektes mit reflektierenden Markern oder Lichtpunkten markiert. In dem mit einer Infrarotkamera aufgenommenen Bild ist die Person als solche nicht zu erkennen, sondern nur die einzelnen Punkte der Markierungen. Zu deren Extraktion genügt die Erzeugung eines Binärbildes durch eine einfache Schwellwert-Operation. Herda et al. [HFP00][HFP01] verwenden diese Technik zur Auswertung von Bildsequenzen, die mit 8 verschiedenen Infrarot-Kameras von einer sich bewegenden Person

aufgenommen wurden, die mit 33 Punkten markiert ist. Die Bildsegmentierung erfolgt hier nicht getrennt von der Modellrekonstruktion, sondern nutzt bei der Stereo-Triangulation Restriktionen, die sich aus der Modellbeschreibung ergeben. Durch Berücksichtigung von Verdeckungen und durch Prädiktion der Punktpositionen wird eine zuverlässige Verfolgung möglich. Ein Nachteil des Verfahrens ist neben des großen experimentellen Aufwandes die notwendige manuelle Initialisierung, d.h. die Zuordnung der Marker im ersten Bild zu den einzelnen Objektmodellteilen durch einen Experten. Marker, die während der Verfolgung verloren werden, können nur durch Benutzer-Interaktion wieder zugeordnet werden.

Von MLDs spricht man auch, wenn die beobachteten Objekte nicht mit Markern versehen sind, sondern die Binärbilder künstlich erzeugt werden. Boyd und Little untersuchen in [BL97] die Gehbewegung einer Person, die sich parallel zur Kamera bewegt und vergleichen die Grauwert-basierte Merkmalsextraktion aus Bildern des optischen Flusses mit manuell erzeugten MLD-Sequenzen. Die Erzeugung von Binärbildern erfolgt hier manuell durch Markierung der Gelenkwinkel mit der Maus in den einzelnen Bildern der Sequenz.

Ein generelles Problem von MLDs ist die fehlende Unterscheidbarkeit der Marken im Bild, da die Detektion bei nur einer Wellenlänge im Infraroten erfolgt. Hierdurch kommt es im Interpretationsprozess leicht zu Mehrdeutigkeiten bei der Zuordnung von den im Bild extrahierten Punkten zu den Modellpunkten, was häufig ein Eingreifen von Seiten des Benutzers notwendig macht.

Aber auch bei der Verwendung von CCD-Kameras im sichtbaren Spektralbereich kann bei kontrollierbaren Umgebungsbedingungen, wie sie beispielsweise in Innenräumen mit definierter Beleuchtung oder im Labor vorliegen, nach einer geeigneten Bildvorverarbeitung ein einfaches Schwellwert-Verfahren angewendet werden, um eine geeignete *region of interest* (ROI) zu bestimmen. Häufig wird hierbei ein statischer Hintergrund angenommen, der von dem aktuellen Kamerabild subtrahiert wird [MN95][WAD96] [MG00][CH00]. Zur Unterstützung der Segmentierung wir meist eine Vorhersage des Systemzustandes gemacht. Gebräuchlich ist z.B. der Kalman-Filter, der auch ein Maß für die Unsicherheit der Schätzung liefert [AM79][RMK95].

- *Farbklassifikation*

Bei der Verwendung von Farbkameras bietet sich für die Bildsegmentierung die Nutzung von Farbinformationen zur Klassifizierung von Objekten an [Sch94][Sch95] [Haf99]. Je nachdem, welche Primärvalenzen[9] zur Modellierung der Farbe verwendet werden, können technik- und wahrnehmungsorientierte Farbräume voneinander unterschieden werden. Während die technikorientierten Farbräume durch einfache lineare

[9] Basisvektoren des verwendeten Farbraums

Transformationen z.B. aus dem RGB-Farbraum hervorgehen, orientieren sich wahrnehmungsorientierte Farbräume an der Psychologie des Farbensehens beim Menschen. Gleiche Abstände im Farbraum werden hier als gleich starke Variation im Farbton wahrgenommen. Durch geeignete Transformationen kann die eigentliche Farbinformation von der Helligkeitsinformation separiert werden, so dass zur Darstellung einer bestimmten Farbe die Angabe von zwei Werten ausreicht.

Beim Vermessen menschlicher Bewegungen spielt insbesondere die Detektion von hautfarbenen Regionen zum Auffinden der Hände oder des Gesichtes eine Rolle. Die Segmentierung hautfarbener Regionen wird meistens als Vorstufe für eine weitere Merkmalsextraktion verwendet. Je nach geforderter Genauigkeit unterscheiden sich die Ansätze hinsichtlich des gewählten Farbraums. Toyama [Toy98] verwendet zur Verfolgung des Gesichtes einen mehrschichtigen Systemaufbau, bei dem die unterste Schicht eine Detektion hautfarbener Pixel durchführt. Ein Pixel wird immer dann als hautfarben klassifiziert, wenn die RGB-Werte innerhalb eines keilförmigen Bereiches mit $k_{RG}^{-} < R/G < k_{RG}^{+}$ und $k_{RB}^{-} < R/B < k_{RB}^{+}$ liegen. Häufig wird jedoch auf einen Klassifikator zurückgegriffen, bei dem die Verteilung der Hautfarbe im Farbraum zuvor anhand von Beispielen trainiert wurde. Starner und Pentland beschreiben in [SP95] ein System zur automatisierten Interpretation von Gebärdensprache. Die Testperson trägt hier entweder einen einfarbigen Handschuh, der leicht im Bild segmentiert werden kann. Alternativ wird die Hand mit einem empirisch ermittelten a priori Modell für die Hautfarbe verfolgt. Für die gefundene Region werden anschließend weitere geometrische Merkmale ermittelt. Auch in der Arbeit von Darrell et al. [DGH98] dient die Segmentierung hautfarbener Bereiche mit einer empirisch ermittelten Verteilung als Vorstufe zu einer sich anschließenden verfeinerten Analyse der erhaltenen ROI. Zur Detektion von Hautfarbe im wahrnehmungsorientierten *CIE*[10] *Lab* Farbraum siehe z.B. die Arbeit von Cai und Goshtasby [CG99].

- *Extraktion von Bildkanten*

Die allein auf Farbklassifikation basierende Bildsegmentierung führt nur bei kontrollierbaren Umgebungsbedingungen zu einer zufriedenstellenden Szeneninterpretation. Für eine robuste Personenverfolgung beispielsweise im Bereich der Sicherheitstechnik müssen darüber hinaus auch Forminformationen, z.B. auf Grundlage von Kanten, berücksichtigt werden.

Rohr [Roh93] ermittelt durch *line matching* des Modells an die Kantenstrukturen im Bild die Position und Haltung einer Person. Die Beobachtung erfolgt hier monokular, und es wird vereinfachend angenommen, dass sich die Person parallel zu einer statischen Kamera bewegt. Auch Wachter [Wac97] untersucht die Gehbewegung von Fuß-

[10] Commission Internationale de l´Eclairage

gängern unter Berücksichtigung von Bildkanten. Das dreidimensionale Personenmodell ist aus elliptischen Kegelstümpfen aufgebaut, die polygonal approximiert werden. Die Bestimmung des Parametervektors, der die Freiheitsgrade des Modells beschreibt, wird als Schätzaufgabe verstanden, bei der die Projektion des Personenmodells in Übereinstimmung mit den Bildstrukturen gebracht wird. Durch Verdeckungsrechnung können verdeckte Kanten- und Flächenabschnitte bei dem ins Bild projizierten Modell eliminiert werden. Das System wird neben der Untersuchung des Gehvorgangs auch an einer Sequenz mit einer gymnastischen Bewegung des Oberkörpers demonstriert, allerdings erfolgt die Beobachtung der Bewegung auch hier nur monokular.

Einen Ansatz, der Punktverteilungsmodelle mit Splines kombiniert, stellen Baumberg und Hogg in [BH94] vor. Die Erstellung eines zweidimensionalen Punktverteilungsmodells für die menschliche Silhouette als Ganzes ist automatisiert und erfolgt durch die Auswertung von Videosequenzen, in denen sich einzelne Personen durch das Bild bewegen. Die Silhouette wird durch B-Spline Funktionen approximiert, deren Stützpunkte zur Konstruktion eines PDM verwendet werden. Durch die Erweiterung des Modells um einen Geschwindigkeitsvektor, der die (zweidimensionale) Bewegungsrichtung der Person im Bild beschreibt, wird ein erweitertes Punktverteilungsmodell erzeugt, das es erlaubt, die aus dem Bild extrahierte Kontur für eine Vorhersage im nächsten Bild zu verwenden. Die Einsatzmöglichkeiten des Modells zur Verfolgung von Personen sind jedoch begrenzt, da die Person stets ganz im Bild sichtbar sein muss. Gerade im Bereich der Sicherheitstechnik bei der Überwachung von Innenräumen ist dies jedoch häufig nicht der Fall. Insbesondere die Beine sind hier häufig durch Tische oder Stühle ganz oder teilweise verdeckt (siehe Bild 3.3).

- *Kombination verschiedener Merkmale*

Ein System, das nicht nur im Labor, sondern auch unter realen Bedingungen eine zuverlässige Szeneninterpretation erlaubt, muss durch die Integration mehrerer Module die spezifischen Schwächen der einzelnen Segmentiermethoden kompensieren, um so zu einem robusten Gesamtsystem zu gelangen.

Das von Wren et al. [WAD96] entwickelte System *Pfinder* modelliert eine Person als zusammenhängende Menge von Regionen mit einer charakteristischen räumlichen und Farbverteilung (vgl. auch Seite 11). Zusätzlich wird durch statistische Analyse der Form der Vordergrundregion ein einfaches Silhouettenmodell zur Charakterisierung des Kopfes, der Hände und der Füße aufgebaut. Aufgrund des einfachen Personenmodells und der monokularen Bildauswertung kann jedoch für die dreidimensionale Position einer Person nur eine grobe Schätzung vorgenommen werden. Hierzu wird weiterhin angenommen, dass sich die Person auf einem ebenen Untergrund befindet.

Eine Kombination verschiedener Merkmale verwenden auch Darrell et al. [DGH98]. Durch die Auswertung von durch einen Stereoaufbau erzeugten Disparitätskarten kön-

nen hier Personen vom Hintergrund getrennt und durch die zusätzliche Verwendung von Farbinformationen statistische Modelle für deren Aussehen erzeugt werden. Ein einfaches Modul zur Gesichtserkennung gestattet weiterhin die Identifizierung von Personen. Auch bei diesem System ist kein explizites Menschmodell hinterlegt und die auf der Identifizierung mit neuronalen Netzen basierende Gesichtserkennung geht davon aus, dass die Person frontal in die Kamera blickt.

Matsumoto und Zelinsky [MZ99] verwenden einen Stereoansatz zur Detektion des Gesichtes. Die Initialisierung des Systems erfolgt entweder durch manuelle Markierung charakteristischer Gesichtspunkte oder automatisiert durch Segmentierung von Hautfarbe. Die Umgebung der charakteristischen Punkte wird als Template für einen Stereomatch verwendet. Das System verwendet ein sehr einfaches Gesichtsmodell, das durch lediglich 6 Punkte definiert wird. Zur Animation eines starren 3D-Modells ist dies ausreichend, eine detaillierte Vermessung der Mimik ist jedoch nicht möglich.

Zhong et al. [ZJD00] verwenden in ihrem monokularen Ansatz zur Verfolgung bewegter Objekte Kanten-, Farb- und Texturinformationen und gehen von einer Bewegung des Objektes aus, so dass bei der Objektverfolgung zusätzlich Informationen aus dem Differenzbild zweier aufeinander folgender Frames verwendet werden können. Der Ansatz wird demonstriert zur Verfolgung der Hand und des Kopfes als Ganzem, erfordert jedoch die Initialisierung durch manuelle Definition einer prototypischen Silhouette im ersten Bild.

2.4 Anwendungen

Die Anwendungsmöglichkeiten videobasierter Techniken zur Beobachtung menschlicher Bewegungen sind vielfältig und reichen von Überwachungsaufgaben im Bereich der Sicherheitstechnik über den Einsatz bei der Mensch-Maschine Kommunikation bis hin zu Applikationen in Medizin und Sport.

Im Bereich der Videoüberwachung kann die automatisierte Szenenanalyse ein wichtiges Hilfsmittel beim Ergreifen von Präventivmaßnahmen zur Vermeidung von Verbrechen oder bei der Verkehrsüberwachung sein. Neben der zuverlässigen Detektion muss hierbei zusätzlich eine Interpretation der beobachteten Verhaltensweisen erfolgen. Haritaoglu et al. [HHD00] verwenden in ihrem System W^4 kein explizites Menschmodell, sondern benutzen zur Charakterisierung von Personen und deren Verhalten geometrische Merkmale, mit denen statistische Modelle für das Aussehen und die Bewegung aufgebaut werden. Neben der Verfolgung von Einzelpersonen können auch Personen in Gruppen segmentiert werden und beispielsweise Objekte identifiziert werden, die von den Personen getragen werden. Die Beobachtung erfolgt monokular mit einer Schwarzweiß- oder Infrarotkamera. Einen Ansatz, bei dem auch eine Interpretation der Aktivitäten der aufgenommenen Personen erfolgt, stellen Brand und Kettnaker in

[BK00] vor. Die Szenenanalyse mit einem Hidden Markov Modell (HMM) wird zur Überwachung eines Büros und für den Einsatz zur Verkehrsszenenanalyse demonstriert. Das System erlaubt auch die automatisierte Detektion anomaler Situationen, in denen die beobachtete Person beispielsweise einschläft oder Verkehrssituationen, in denen ein Autofahrer an einer Kreuzung falsch abbiegt.

Auch bei multimodalen Systemen, in denen eine Interaktion zwischen Mensch und Maschine stattfindet, können videobasierte Techniken zu einer intuitiven Bedienbarkeit beitragen. Zur Charakterisierung von dynamischen Gesten oder Mimiken kommen dabei zunehmend HMM-basierte Ansätze zum Einsatz, die bislang vor allem in der Spracherkennung Verwendung fanden. Lee und Kim [LK99] verwenden ein Hidden Markov Modell zur Interpretation von Bewegungen der Hand und demonstrieren den Einsatz zur Steuerung einer Powerpoint-Präsentation mittels einfacher, zuvor definierter Gesten. Auch Starner und Pentland [SP95] verfolgen Handbewegungen und können durch Trainieren eines HMM amerikanische Gebärdensprache erkennen. Zur visuellen Interpretation von Handgesten bei der Mensch-Maschine Interaktion vergleiche auch Pavlovic et al. [PSH97].

Neben der Vermessung der Extremitäten zur Interpretation der Gestik spielt die Gesichtsdetektion zur Personenidentifikation, bei der Mimikerkennung oder auch in medizinischen Anwendungen eine große Rolle [KP88][LTC97][MZ99][GLC01][HWR01]. Ein vielfach verwendetes Schema zur Einteilung der Ausdrucksweise durch Variation der Gesichtszüge ist das von Ekman und Friesen [EF78] entwickelte *Facial Action Coding System* (FACS), bei dem die von einem menschlichen Beobachter visuell noch wahrnehmbaren Veränderungen der Gesichtszüge in 46 so genannte AUs (Action Units) eingeteilt werden. Ein Nachteil von FACS ist jedoch, dass in diesem System keine zeitlichen Informationen enthalten sind und die Analyse durch einen menschlichen Experten subjektiv und fehleranfällig ist. Dies führt auf die Notwendigkeit von Systemen zur automatisierten Vermessung des Gesichtes und der Interpretation verschiedener Mimiken. Einen Überblick über die Arbeiten zur Analyse der Ausdrucksweise des menschlichen Gesichtes bis etwa 1999 geben Pantic und Rothkrantz in [PR00].

Weitere Einsatzmöglichkeiten finden sich auch im Bereich der Ergonomie zur Gewinnung anthropometrischer Daten. Die ergonomische Gestaltung von Arbeitsplätzen gehört hierzu ebenso wie die Optimierung des Fahrzeuginnenraumes in der Automobilindustrie (vergleiche z.B. [Sei94]).

3 Modellierung und Interpretationsprozess

In diesem Kapitel wird das System STABIL++[11] vorgestellt, das in den vergangenen Jahren am *Bayerischen Forschungszentrum für wissensbasierte Systeme* (FORWISS) mit dem Ziel entwickelt worden ist, die Detektion und Verfolgung von Personen in Videobildfolgen in Echtzeit zu ermöglichen. Das System stand als Umgebung für die im Rahmen dieser Arbeit entstandenen Modelle und Methoden zur Verfügung und wurde unter anderem um Klassen und Algorithmen erweitert, die eine Verfolgung von Personen mit flexiblen Konturmodellen erlauben.

Im Folgenden wird auf die wesentlichen Bestandteile des Systems eingegangen. Abschnitt 3.1 gibt zunächst eine Übersicht über das Gesamtsystem, das sich grob durch das hinterlegte Modellwissen und den für die Szenenanalyse verwendeten Interpretationsprozess charakterisieren lässt. Aufgrund seines modularen Aufbaus ist das System in vielen Bereichen der videobasierten Bildanalyse einsetzbar. Abschnitt 3.2 stellt die wichtigsten Anwendungsmöglichkeiten vor, bevor in den Abschnitten 3.3 und 3.4 eine detailliertere Beschreibung des Modellwissens und des Interpretationsprozesses folgt. Der abschließende Abschnitt 3.5 zeigt die Grenzen der bislang verwendeten Bildinterpretation durch Farbsegmentierung auf und führt so auf die Notwendigkeit weiterer Merkmale, um so eine robustere Objekterkennung zu erreichen.

3.1 Übersicht über das System STABIL++

Das System STABIL++ ermöglicht die modellbasierte Interpretation von Videobildfolgen, in denen beliebig aufgebaute artikulare Objekte automatisch detektiert und dreidimensional verfolgt werden können. Die Modellierung des hinterlegten hierarchischen Objektmodells und des Interpretationsprozesses erfolgt durchgehend in 3D. Der durch die objektorientierte Implementierung in der Programmiersprache C++ erzielte modulare Aufbau ermöglicht eine flexible Erweiterung der Systemarchitektur und die einfache Anpassung des Systems an die jeweilige Anwendung.

Bild 3.1 gibt eine Übersicht über das Gesamtsystem. Die Szeneninterpretation erfolgt durch Beobachtung des observierten Bereiches mit einem kalibrierten Stereokamerasystem. Die Anzahl der verwendeten Kameras ist beliebig und prinzipiell nicht auf zwei beschränkt.

---

[11] **S**ystem for **T**racking **A**rticulated O**b**jects Using **I**mage-Based 3D **L**ocalization Implemented in C++

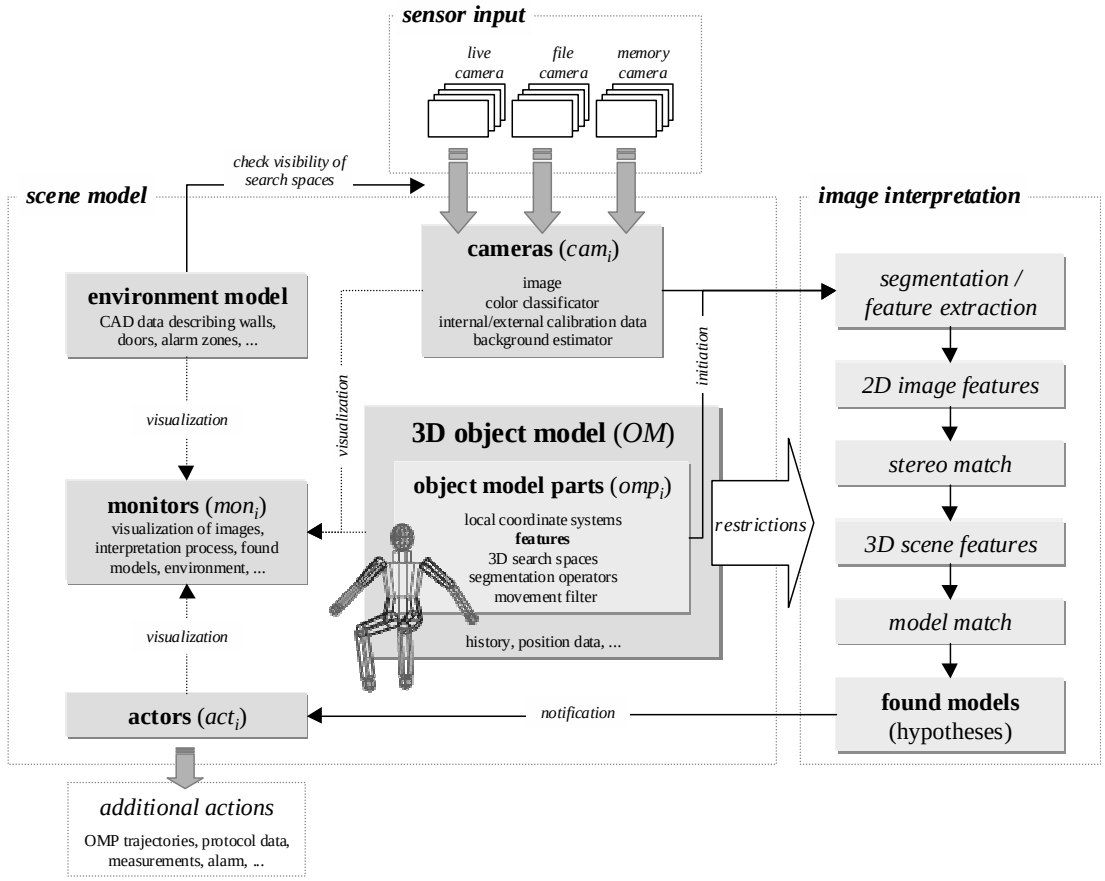


Bild 3.1: ***Übersicht über das System STABIL++*** *zur dreidimensionalen Detektion und Verfolgung von Objekten in Videobildfolgen.*

Zentraler Bestandteil des Systems ist das hierarchisch aufgebaute dreidimensionale Objektmodell (*OM*), welches das Wissen darüber beinhaltet, wonach im Bild gesucht werden soll und wie die Segmentierung zu erfolgen hat. Das Objektmodell erlaubt die Beschreibung beliebig aufgebauter artikularer Objekte, so dass das System leicht an die konkrete Anwendung angepasst werden kann und die Anwendung nicht auf Personen beschränkt ist. Das Objektmodell setzt sich aus mehreren Objektmodellteilen (OMPs, omp_i) zusammen, die zu ihrer Charakterisierung mit Merkmalen (*features*) ausgestattet sind. Die Merkmale kapseln das Wissen darüber, wie im Interpretationsprozess die Segmentierung im Bild durchgeführt werden soll und initiieren die Extraktion der Objektmodellteile in den verschiedenen Ansichten der Szene.

Im restriktionsgesteuerten Interpretationszyklus führen die Ergebnisse der Bildsegmentierung zunächst auf 2D-Bildmerkmale, d.h. geometrische Primitive wie z.B. Ellipsen oder Kanten, die durch verschiedene zusätzliche Attribute wie Farbe oder geometrische Kennzahlen charakterisiert sind. Mit den internen und externen Kalibrierdaten der verwendeten Kameras werden die Bildmerkmale in einem Stereomatch in so genannte 3D-Szenenmerkmale überführt, deren Zuordnung zu den im Objektmodell hinterlegten Modellmerkmalen schließlich zu keiner, einer oder mehreren Hypothesen über die gefun-

denen Objektmodellinstanzen führt. Um die Komplexität der Suche einzuschränken, werden bei der Bildinterpretation Restriktionen sowohl auf Merkmals- als auch auf Modellebene verwendet. Eine detaillierte Beschreibung des restriktionsgesteuerten Interpretationsprozesses und der Generierung von Hypothesen sowie deren qualitative Bewertung findet sich in der Arbeit von Ridder [Rid99].

Da ein detektiertes Objekt nicht losgelöst von der es umgebenden Welt betrachtet werden kann, wird die Umgebung zusätzlich zu der Spezifizierung von Objektmodell und Kameras durch ein CAD-Umgebungsmodell (*environment model*) beschrieben. Dieses umfasst neben der Definition von Wänden und Inventargegenständen auch sensitive Alarmbereiche in einem Gebäudegrundriss und ermöglicht, den Bildeinzug und damit auch den Interpretationsprozess auf die Kameras zu beschränken, in deren Sichtbereich das Objektmodell erwartet wird.

Die Ergebnisse eines Interpretationszyklusses können entweder in den Monitoren (mon_i) visualisiert werden, oder das System erzeugt Ausgaben über die so genannten Akteure (act_i). Diese flexiblen Schnittstellen können bei Bedarf aktiviert oder neu programmiert werden, um beispielsweise Protokolldaten zu erzeugen, Messergebnisse abzuspeichern oder auch um einen Alarm auszulösen, wenn das System mit einem externen Alarmgeber gekoppelt ist.

3.2 Anwendungen des Systems

Das dem System STABIL++ zugrundeliegende hierarchische Objektmodell erlaubt eine Vielzahl von Anwendungen. Das Modell kann flexibel an die konkreten Anforderungen angepasst werden und erlaubt die Detektion beliebig aufgebauter artikularer Objekte und deren Verfolgung in Videobildfolgen. Durch die Vermessung der dreidimensionalen Positionen der einzelnen Objektmodellteile können Daten über deren 3D-Bewegung im Raum sowie Lageinformationen der Teile zueinander gewonnen werden.

Insbesondere die Vermessung des menschlichen Körpers und die Analyse menschlicher Bewegungen spielt in vielen Beispielen aus der Praxis eine wichtige Rolle. Die Anwendungen reichen von ergonomischen Studien über den Einsatz in der Mensch-Maschine Kommunikation bis hin zur Animation virtueller Charaktere oder auch Anwendungen in der Sicherheitstechnik. Beispielhaft werden im Weiteren der Einsatz von STABIL++ im Bereich der Ergonomie, der videobasierten Sicherheitstechnik und der Virtual Reality näher erläutert.

3.2.1 Ergonomie

Menschmodelle werden in den verschiedensten Bereichen zur Untersuchung ergonomischer Fragestellungen eingesetzt. Neben dem menschlichen Skelett werden häufig auch die Weichteile, deren geometrische und physikalische Kopplung mit dem Skelett und die Kinematik der Bewegungen modelliert und animiert (vgl. Abschnitt 2.1). Einige Modelle beschreiben darüber hinaus das menschliche Wohlbefinden bei bestimmten Haltungen. Die Einsatzmöglichkeiten solcher Menschmodelle sind vielfältig. An die ergonomische Gestaltung von Arbeitsplätzen beispielsweise bei der Bedienung von Maschinen ist hier ebenso zu denken wie an die Bewegungs- und Haltungsanalyse in der Sportmedizin beim Leistungssport oder zur Rehabilitation.

Die Erstellung eines realistischen Modells erfordert die Berücksichtigung einer Vielzahl anthropometrischer und kinematischer Daten. Die Datengewinnung erfolgt dabei durch Vermessung der Haltung und des Bewegungsablaufes real agierender Personen. Bei diesem komplexen Prozess kommen meist sensible Messapparaturen zum Einsatz, die einerseits teuer sind und die darüber hinaus die untersuchte Person in ihrer Bewegungsfreiheit einschränken und so zu verfälschten Resultaten führen.

Das System STABIL++ erlaubt die automatisierte Vermessung von Bewegungsvorgängen, bei der das hinterlegte Objektmodell flexibel an die konkrete Anwendung angepasst werden kann. Bild 3.2 zeigt den Einsatz zur Vermessung des Einstiegsvorganges in ein Auto. Die Vermessung und die Analyse der 3D-Trajektorien der einzelnen Objektmodellteile und deren Gelenkwinkel wird in der Automobilindustrie zur ergonomischen Fahrzeug-Innenraumgestaltung eingesetzt, um so eine möglichst optimale Positionierung der Bedienelemente und Form der Karosserie zu garantieren. Die Person trägt einen Anzug, welcher die Bewegungsfreiheit nicht beeinträchtigt und dessen Gelenkwinkel mit farbigen Markierungen versehen sind. Die Bildsegmentierung erfolgt in diesem Fall mit einem zuvor trainierten Farbklassifikator (vgl. Abschnitt 3.5). Durch die Verwendung verschiedener Farben (vgl. Abschnitt 3.3.2) können im Interpretationsprozess Mehrdeutigkeiten, wie sie z.B. bei der Auswertung von MLD-Aufnahmen vorkommen (vgl. Abschnitt 2.3), vermieden werden.

Die Bewegung wird mit einem kalibrierten Stereokamerasystem mit mindestens zwei Kameras aufgenommen. Die Verwendung einer größeren Anzahl von Kameras erleichtert die Detektion, da so zeitweilige Verdeckungen in einem Bild durch die übrigen Ansichten ausgeglichen werden können. Aus den Kalibrierdaten der Kameras wird für jede Farbmarke eine 3D-Position ermittelt und aus dem Abgleich der Daten mit dem hinterlegten Objektmodell die Haltung der Person für jedes Bild exakt vermessen. Aus den Positionen der einzelnen Skelettpunkte lassen sich insbesondere auch die Gelenkwinkel bestimmen. Die so gewonnenen kinematischen und anthropometrischen Daten können für die weitergehende Analyse verwendet und in ein komplexeres Menschmodell integriert werden.

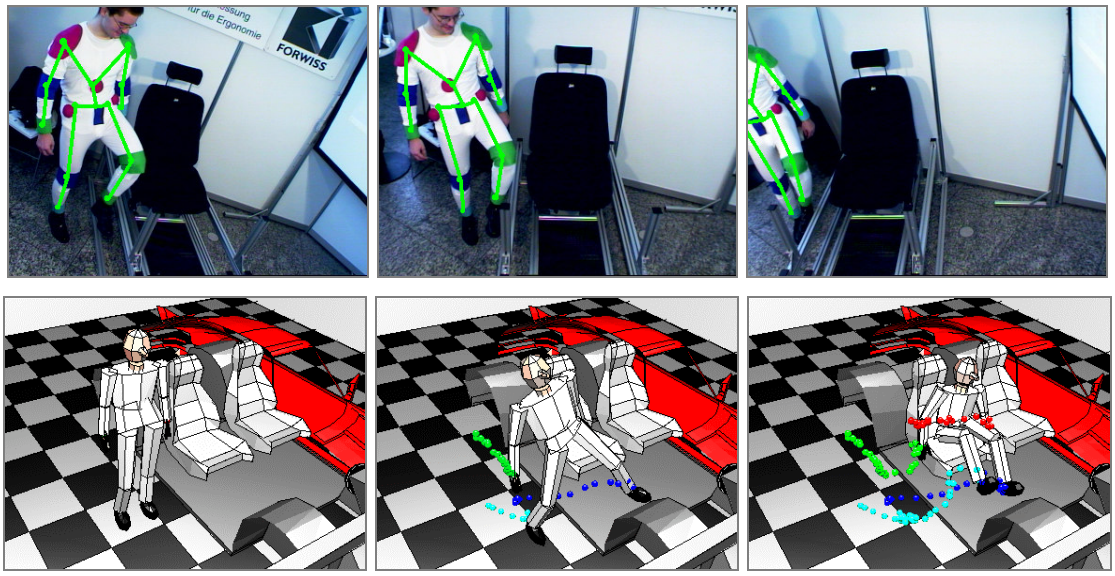


Bild 3.2: ***Anwendung von STABIL++ in der Ergonomie*** *zur Vermessung des Einstiegsvorgangs in ein Auto. Oben: Simultane Aufnahme der Szene mit drei Kameras mit eingezeichneter Modellhypothese (grün). Unten: dreidimensionale Visualisierung des Trajektorienverlaufes für Füße und Hände.*

3.2.2 Sicherheitstechnik

Für die videobasierte Überwachung sensibler Bereiche beispielsweise in Banken oder auch die Überwachung von Kaufhäusern werden Systeme eingesetzt, die aus einer Vielzahl von Kameras und mehreren Bedienplätzen für das Wachpersonal bestehen (vgl. Bild 3.3). Neben dem Einsatz von Fixkameras, deren Position und Ausrichtung im Raum fest vorgegeben und unveränderbar sind, können auch schwenk-/neigbare Domekameras verwendet werden, mit denen eine Person aktiv verfolgt werden kann. Die Auswertung der Videobilder erfolgt durch Wachpersonal und ist daher personal- und kostenintensiv. Die Positionierung der Kameras geschieht manuell und ist damit relativ langsam und ungenau.

Mit STABIL++ können im Bereich der videobasierten Überwachung Personen automatisch detektiert und vom System dreidimensional verfolgt werden. Informationen über die Position im Gebäudegrundriss oder das Aussehen können daraufhin beispielsweise zur Beweissicherung gespeichert oder an den Etagendetektiv weitergeleitet werden. Das integrierte Umgebungsmodell (siehe auch Abschnitt 3.3.5) erlaubt darüber hinaus die Definition von Alarmzonen, bei deren Betreten beispielsweise ein externen Alarmgeber aktiviert werden kann.

Eine weitere denkbare Anwendung ist die automatische Überwachung von Eingangsbereichen. In Kombination mit schwenk-/neigbaren Kuppelkameras werden durch Zooming formatfüllende Portraitaufnahmen eintretender Personen möglich.

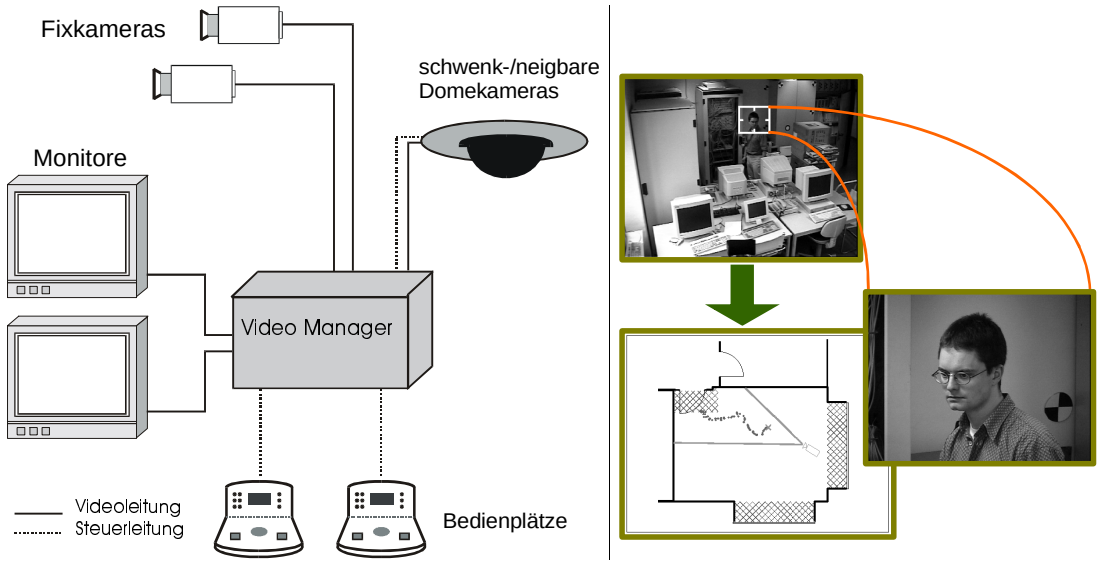


Bild 3.3: ***Anwendung des Systems in der Sicherheitstechnik*** *zur Überwachung sensibler Bereiche. Links: gängiger Aufbau eines videobasierten Überwachungssystems. Rechts: die eindringende Person wird automatisch detektiert, deren Position dreidimensional vermessen und in einem Grundriss dargestellt. Durch Positionierung schwenk-/neigbarer Domekameras können formatfüllende Aufnahmen erstellt werden.*

3.2.3 Virtual Reality

Bei der Produktion von Filmen und Werbespots werden häufig Animationen virtueller Charaktere eingesetzt. Dabei werden die Bewegungen der Extremitäten oder der Gesichtsausdruck eines realen Schauspielers vermessen, um mit diesen Daten eine Trickfigur zu animieren. Die mit STABIL++ gewonnenen kinematischen Daten bei der dreidimensionalen Vermessung menschlicher Bewegungen können auch für diesen Zweck verwendet werden. Das in dieser Arbeit erstellte Gesichtsmodell ermöglicht neben Anwendungen in der Sicherheitstechnik auch die Verfolgung von Gesichtsausdrücken in Videobildfolgen und damit die Generierung der grundlegenden Daten für die Animation geeigneter Avatare mit verschiedenen Mimiken.

3.3 Modellwissen

3.3.1 Szenenmodell

Das Szenenmodell in STABIL++ beinhaltet sämtliche Informationen, die für den Interpretationsprozess relevant sind und strukturiert das Wissen über das zu detektierende Objekt und dessen Umgebung. Durch die Definition verschiedener Szenen kann die

Wissensbasis je nach Anwendung so modifiziert werden, dass eine reale Szene möglichst gut modelliert wird.

Neben der Beschreibung des *Objektmodells* und dessen Charakterisierung durch Merkmale enthält eine Szene Informationen über die verwendeten *Kameras* und die *Umgebung*, in der die Detektion stattfindet. Die Systemarchitektur erlaubt die Definition eines *initialen Suchraumes*, an dem nach dem *initialen Objektmodell* gesucht wird. Der initiale Suchraum kann sich aus mehreren dreidimensionalen Suchraumkugeln zusammensetzen und beschreibt diejenigen Raumbereiche, in denen Objekte erwartet werden. Nach der erfolgreichen Detektion verwaltet das Szenenmodell die gefundenen *Objektmodellinstanzen* und ermöglicht so deren Re-Detektion in einem folgenden Interpretationszyklus. Am Ende eines Interpretationszyklus können die Ergebnisse in *Monitoren* visualisiert werden, oder das System initiiert über die *Akteure* erweiterte Ausgaben oder Aktionen.

3.3.2 3D-Objektmodell

Einen wesentlichen Bestandteil eines modellbasierten Systems zur Interpretation von Videobildfolgen bildet das Wissen zur Beschreibung der Objekte, die in den Bildern detektiert werden sollen. Das im Szenenmodell hinterlegte Objektmodell (*object model*, OM) beschreibt, wonach im Bild gesucht werden soll und wie die Ergebnisse der Extraktion aus den Bildern und die sich daraus ergebenden Messdaten zu interpretieren sind. Das in dieser Arbeit verwendete Objektmodell setzt sich aus mehreren Objektmodellteilen (*object model parts*, OMPs) zusammen und erlaubt die Detektion artikularer Objekte, bei denen die OMPs zwar in der Größe fixiert sind, aber beliebige Positionen zueinander einnehmen können. Aufgrund seiner hierarchischen Struktur ist das Modell flexibel an die jeweilige Anwendung anpassbar. In dieser Arbeit wird insbesondere die Anwendung auf die Detektion des menschlichen Körpers verwendet.

Jedem OMP ist ein Merkmal (*feature*) zugeordnet, das es beschreibt und durch das die Bildverarbeitungsoperatoren definiert werden, die bei der Extraktion in den Bildern anzuwenden sind. Im Interpretationsprozess initiieren die einzelnen OMPs die Extraktion ihrer Merkmale in den verschiedenen Kamerabildern, die Merkmale selber kapseln wiederum das Wissen darüber, wie die Extraktion in den Bildern vorzunehmen ist. Auf die in dieser Arbeit verwendeten Merkmale *Farbe* und *Kanten* wird in Abschnitt 3.5 und Kapitel 4 näher eingegangen.

Das Beschreibung des Objektmodells lässt sich in eine innere, eine geometrische und eine äußere Modellstruktur einteilen, die im Folgenden kurz vorgestellt werden. Weitere Details werden in [Rid99] und [RMR99] beschrieben.

Innere Modellstruktur

Die innere Modellstruktur beschreibt den Zusammenhang zwischen den einzelnen Objektmodellteilen. Aufgrund des hierarchischen Aufbaus des Objektmodells kann die innere Struktur als Baum

$$B = (V, E) \tag{3.8}$$

mit

$$\begin{aligned} V &= \{V_1, ..., V_n\} \\ E &= \{E_1, ..., E_{n-1}\} \end{aligned} \tag{3.9}$$

dargestellt werden. Die Menge der Knoten V (*vertices*) beschreibt die einzelnen OMPs, die Menge der Kanten E (*edges*) die Verbindungen der OMPs zueinander. Bis auf ein ausgezeichnetes Objektmodellteil, das die Wurzel des Baumes bildet und keinen Vorgänger hat, besitzt jedes OMP genau einen Vorgänger sowie keinen, einen oder mehrere Nachfolger. Bild 3.4 und Tabelle 3.1 zeigen die Modellierung des menschlichen Körpers durch 16 Knoten und deren Zuordnung zu den jeweiligen Objektmodellteilen. Die Knoten entsprechen charakteristischen Punkten und Gelenken des menschlichen Skelettes. Als Wurzelelement (*root object model part*) dient die Hüfte.

Bei der Wahl der farbigen Markierungen für die Gelenke sollte darauf geachtet werden, dass die entsprechenden Farbklassen im Farbraum möglichst weit separiert sind (vgl. Bild 3.11). Geeignet sind z.B. die Farben rot, grün, blau, magenta, cyan und gelb. Durch die Verwendung verschiedener Farben wird die Anzahl der möglichen Hypothesen im Interpretationsprozess (siehe Abschnitt 3.4.2) reduziert. Weiterhin sollte darauf geachtet werden, dass Gelenke, die sich während des Bewegungsablaufes sehr nahe kommen –wie z.B. rechte Hand und rechter Oberschenkel– mit verschiedenfarbigen Markierungen versehen werden.

Die in Bild 3.4 und Tabelle 3.1 gewählte Farbzuordnung entspricht der in Bild 3.2 gezeigten Markierung der Gelenke. Auf die Verwendung der Farben rot und gelb wurde in diesem Beispiel verzichtet, da rot bei variierender Beleuchtung häufig als magenta klassifiziert wird (vgl. Bild 3.12) und gelb bei hellem Licht durch Reflexionen oft weiß erscheint.

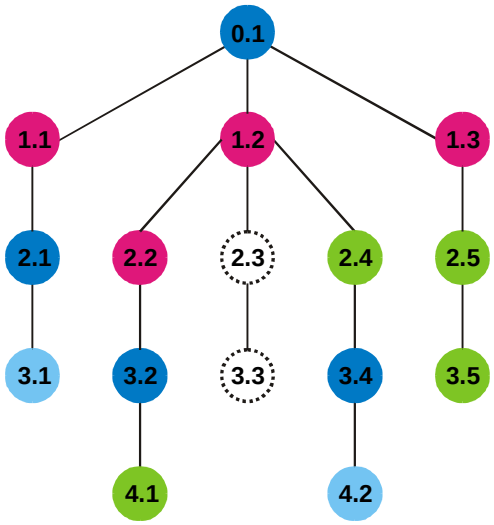


Bild 3.4: ***Innere Modellstruktur****. Gezeigt ist der Baum für die Modellierung des menschlichen Skeletts mit 16 Objektmodellteilen (OMPs). Jedem OMP kann ein Merkmal zugeordnet werden. Im Beispiel erhält jedes OMP eine farbige Markierung als Merkmal. Die Farbe der Knoten entspricht den Farben der Markierungen. Den beiden schwarz gepunkteten Knoten, die Hals und Kopf repräsentieren, sind in diesem Beispiel kein Merkmal zugeordnet.*

Tabelle 3.1: ***Zuordnung der Knoten in Bild 3.4 zu den Objektmodellteilen****.*

Knoten	OMP	Farbe	Knoten	OMP	Farbe
0.1	Hüfte	*blau*	**2.5**	linker Unterschenkel	*grün*
1.1	rechter Oberschenkel	*magenta*	**3.1**	rechter Fuß	*cyan*
1.2	Rumpf	*magenta*	**3.2**	rechter Unterarm	*blau*
1.3	linker Oberschenkel	*magenta*	**3.3**	Kopf	-
2.1	rechter Unterschenkel	*blau*	**3.4**	linker Unterarm	*blau*
2.2	rechter Oberarm	*magenta*	**3.5**	linker Fuß	*grün*
2.3	Hals	-	**4.1**	rechte Hand	*grün*
2.4	linker Oberarm	*grün*	**4.2**	linke Hand	*cyan*

Geometrische Modellstruktur

Zur Erweiterung der inneren Modellstruktur wird jedem Knoten aus V ein 3D-Punkt und jeder Kante aus E eine Richtung zugeordnet. Jedes Objektmodellteil erhält so ein lokales Koordinatensystem, das seine Lage im Raum beschreibt und dessen Ursprung als Ursprung des OMP angesehen werden kann. Die Ausrichtung der z-Ache ist bei dem in Bild 3.5 gezeigten Menschmodell so gewählt, dass sie nach Möglichkeit mit der Skelettstruktur zusammenfällt. Die Koordinatensystem-Ursprünge liegen in den Gelenken.

Die Lage des lokalen Koordinatensystems für ein Objektmodellteil omp_j wird im System des Vorgängers omp_{j-1} angegeben. Damit wird durch die zugehörige homogene Transformation

$$^{j-1}T_j \in \mathbb{R}^{4x4} \tag{3.10}$$

welche die Transformation der (homogenen) Koordinaten beschreibt (siehe Anhang A), die Beschreibung der Lage der einzelnen OMPs zueinander möglich. Aufgrund der hierarchischen Modellstruktur kann durch wiederholte Multiplikation der Transformationsmatrizen von den lokalen Koordinaten des Objektmodellteils omp_j auf Weltkoordinaten zurückgerechnet werden. Bei der Transformation

$$^{w}T_j = {}^{w}T_0\,{}^{0}T_1 \cdots {}^{j-2}T_{j-1}\,{}^{j-1}T_j \tag{3.11}$$

mit der (durch Anwendung auf den Nullvektor) auch der Ursprung von omp_j im Weltkoordinatensystem (WKS) berechnet werden kann, wird die Transformationsreihenfolge durch die innere Modellstruktur vorgegeben: Der Baum wird, ausgehend von dem zu omp_j gehörenden Knoten, rekursiv bis zum Wurzelelement durchlaufen. $^{w}T_0$ ist die Transformationsmatrix für den Übergang vom Koordinatensystem des *root object model part* ins Weltkoordinatensystem.

Äußere Modellstruktur

Durch die äußere Modellstruktur wird die Erscheinungsform des Objektmodells beschrieben. Die einzelnen OMPs werden durch rotationssymmetrische Volumenkörper approximiert, deren Rotationsachsen mit den *z*-Achsen der lokalen Koordinatensysteme zusammenfallen. Die Art des verwendeten Volumenkörpers hängt von dem zu modellierenden OMP ab. Verwendet werden Kugeln, Ellipsoide, Zylinder und Kegelstümpfe. Bei dem in Bild 3.6 gezeigten Menschmodell beispielsweise werden Kopf, Hände und Füße durch Kugeln und Ellipsoide, der Rumpf sowie die Extremitäten durch Kegelstümpfe modelliert. Die Berücksichtigung der Ausdehnung der einzelnen OMPs kann im Interpretationsprozess dazu verwendet werden, um geometrischen Restriktionen zu definieren, mit denen überprüft wird, ob sich die einzelnen Teile gegenseitig durchdringen.

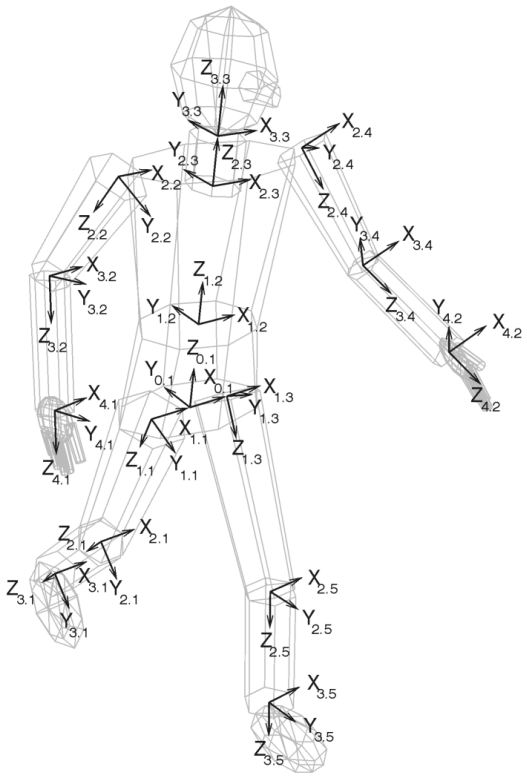


Bild 3.5: ***Geometrische Modellstruktur****. Jedem Objektmodellteil ist ein lokales Koordinatensystem zugeordnet, das seine Lage im Raum bestimmt. Die z-Achse wird so gelegt, dass sie mit der Richtung des zugehörigen Knochens zusammenfällt.*

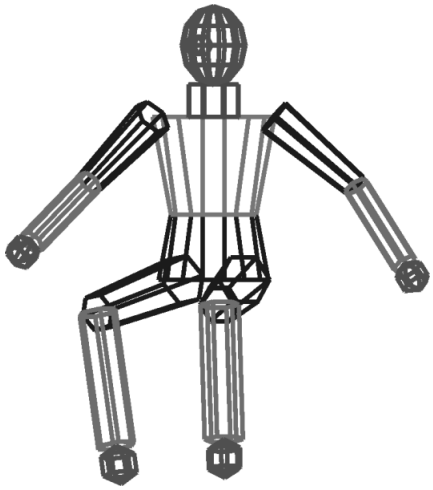


Bild 3.6: ***Äußere Modellstruktur****.*

Modellmerkmale

Neben der Darstellung durch Volumenkörper können die OMPs zusätzlich durch Merkmale (*features*) beschrieben werden, durch die auch die Methoden für die Bildsegmentierung festgelegt werden (vgl. Abschnitt 3.3.6). Die den OMPs zugeordneten Merkmale werden als Modellmerkmale f_i bezeichnet. Im Interpretationsprozess initiieren die einzelnen OMPs die Extraktion ihrer Merkmale in den Bildern der verschiedenen Kameras. Die bei der Segmentierung erhaltenen 2D-Bildmerkmale werden anschließend in so genannte 3D-Szenenmerkmale überführt. Die Szenenmerkmale sind Punkte, Kanten oder Ellipsen, für die eine 3D-Position und Lage bestimmen wird.

Da die Beschreibung des Objektmodells durch Merkmale ein wesentlicher Bestandteil des in dieser Arbeit verwendeten Systems darstellt, wird auf die verwendeten Merkmale *Farbe* und *Kanten* in Abschnitt 3.5 sowie in Kapitel 4 detailliert eingegangen.

3.3.3 Kameramodell

Ziel der quantitativen Analyse von Bildern zur videobasierten Interpretation einer aufgenommenen Szene ist es, Aussagen über die Eigenschaften von Objekten in der realen Welt zu treffen. Objekteigenschaften wie Größe oder Lage im Raum werden dabei in Bezug auf ein externes WKS ermittelt und angegeben. Um solche Messgrößen aus den von der Kamera aufgenommenen zweidimensionalen Bildern berechnen zu können, ist ein detailliertes Verständnis des zugrunde liegenden Abbildungsvorganges nötig. Dieser beschreibt, wie die Raumkoordinaten eines im WKS angegebenen Punktes in die internen Rechner- bzw. Pixelkoordinaten transformiert werden.

Lochkameramodell

Zur Modellierung wird das in Bild 3.7 dargestellte Lochkameramodell verwendet, das auch radiale Verzerrungen berücksichtigt (vgl. [Len87]). Die Abbildung eines Weltpunktes $\vec{p}_w = (x_w, y_w, z_w)^T$ in Pixelkoordinaten $\vec{p}_r = (x_r, y_r)^T$ wird durch einen vierstufigen Prozess beschrieben:

1. Transformation ins Kamerakoordinatensystem (KKS)

Im ersten Schritt wird der Weltpunkt $\vec{p}_w = (x_w, y_w, z_w)^T$ durch Angabe einer Rotation R und eines Translationsvektors $\vec{T}$ gemäß

$$\vec{p}_c = (x_c, y_c, z_c)^T = {}^cR_w^{-1}(\vec{p}_w - \vec{T}) \tag{3.12}$$

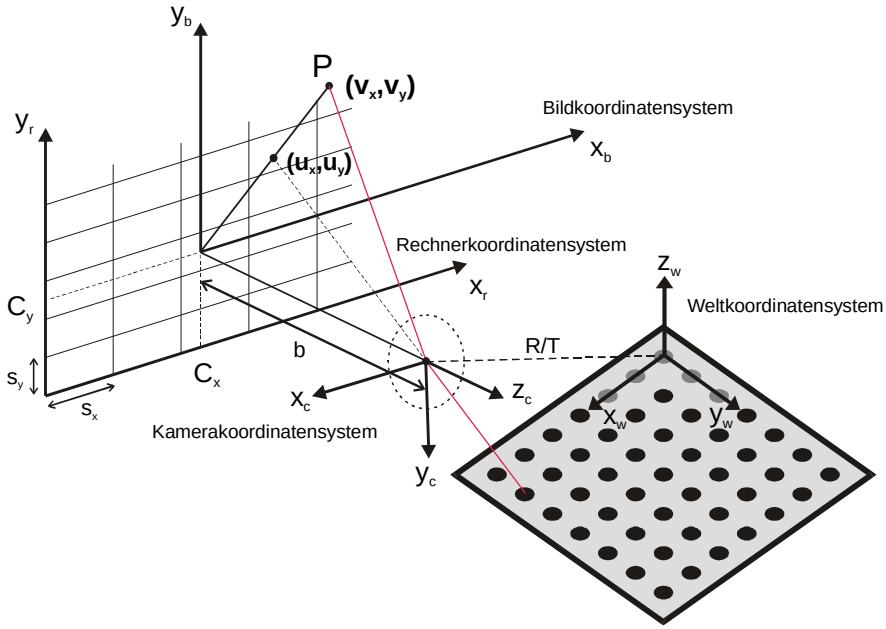


Bild 3.7: ***Lochkameramodell mit radialer Verzerrung.***

ins Kamerakoordinatensystem transformiert (vgl. auch Anhang A). Die Rotationsmatrix ${}^{c}R_{w} \in I\!R^{3x3}$ sowie der Translationsvektor $\vec{T}$ beschreiben die Lage der Kamera in Bezug auf das WKS und geben an, wie sich die Basisvektoren $(\vec{e}_i)_w$ des WKS in die Basisvektoren $(\vec{e}_i)_c$ des KKS transformieren. ${}^{c}R_{w} \cdot (\vec{e}_i)_w$ enthält also die Koordinaten der Basisvektoren des KKS, ausgedrückt zur Basis des WKS. $\vec{T}$ gibt die Translation des KKS im Vergleich zum WKS an, enthält also die Koordinaten des KKS-Ursprungs, ausgedrückt durch die Basisvektoren des WKS. Die Darstellung der Transformation mittels homogener Koordinaten ist in Anhang A gezeigt.

2. Perspektivische Projektion in die Bildebene

Die Projektion des Punktes $\vec{p}_c$ in das zweidimensionale Bildkoordinatensystem (BKS) geschieht durch perspektivische Projektion:

$$\vec{u} = (u_x, u_y)^T = (b\frac{x_c}{z_c}, b\frac{y_c}{z_c})^T \tag{3.13}$$

Die Abbildungseigenschaften werden dabei durch die Kamerakonstante *b* beschrieben. Man erhält so Koordinaten u_x und u_y, die sich bei einer fehlerfreien, d.h. unverzerrten Abbildung ergeben würden.

3. Korrektur der radialen Verzerrung

Die in der Praxis auftretenden Linsenfehler führen dazu, dass die unverzerrten Koordinaten (u_x,u_y) korrigiert werden müssen. Die zu beobachtenden Verzerrungen werden in erster Ordnung durch

$$\vec{v} = (v_x, v_y)^T = \left(\frac{2u_x}{1+\sqrt{1-4\kappa(u_x{}^2 + u_y{}^2)}}, \frac{2u_y}{1+\sqrt{1-4\kappa(u_x{}^2 + u_y{}^2)}}\right)^T \qquad (3.14)$$

berücksichtigt. Für $\kappa > 0$ spricht man von kissenförmigen Verzeichnungen, durch $\kappa < 0$ werden tonnenförmige Verzeichnungen modelliert.

4. Umrechnung in Pixelkoordinaten

Im letzten Schritt müssen die sich für (v_x,v_y) ergebenden kontinuierlichen Werte schließlich noch diskretisiert und gemäß

$$\vec{p}_r = (x_r, y_r)^T = \left(\frac{v_x}{s_x} + C_x, \frac{v_y}{s_y} + C_y\right)^T \qquad (3.15)$$

auf Pixelkoordinaten umgerechnet werden. Die Skalierungsfaktoren s_x und s_y geben den horizontalen bzw. vertikalen Abstand zweier Pixel auf dem CCD-Chip an. Die Lage des Hauptpunktes (Lot des Abbildungszentrums auf die Bildebene) im Pixelkoordinatensystem ist durch (C_x,C_y) gegeben.

3.3.4 Kamerakalibrierung

Zur quantitative Auswertung von Videobildern muss der verwendete experimentelle Aufbau kalibriert werden. Hierbei werden die inneren Parameter des in Abschnitt 3.3.3 beschriebenen Lochkameramodells bestimmt. Wenn Messungen gemacht werden sollen, die die absolute Position eines Objektes in der Umgebung bestimmen, muss weiterhin die Lage der Kameras in Bezug auf ein externes (Welt-) Koordinatensystem bestimmt werden. Die Kalibrierung des in dieser Arbeit verwendeten Stereoaufbaus mit zwei oder mehr Kameras vollzieht sich daher in mehreren Schritten: Zunächst werden für jede Kamera die internen Parameter durch das im folgenden Abschnitt beschriebene Verfahren ermittelt. Im zweiten Schritt wird die Lage einer Referenzkamera in Bezug

auf ein frei wählbares Weltkoordinatensystem bestimmt, und schließlich werden alle weiteren Kameras in Bezug auf die Referenzkamera kalibriert.

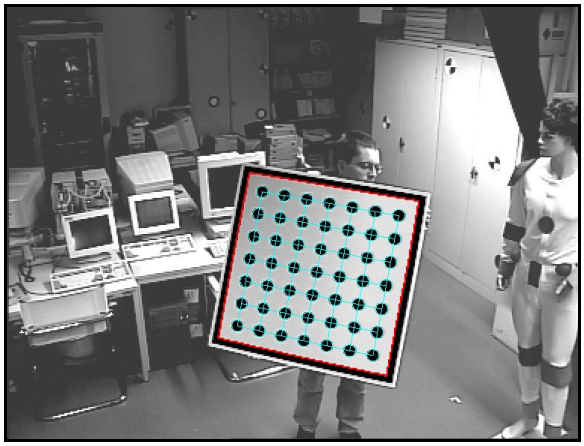


Bild 3.8: ***Kalibrierung der internen und externen Kameraparameter*** *mit Hilfe einer exakt vermessenen zweidimensionalen Kalibrierplatte und mehreren Kalibriermarken, die im Hintergrund an den Schrankwänden zu sehen sind.*

3.3.4.1 Innere

Die Bestimmung der 6 inneren Parameter des Lochkameramodells (b, κ, s_x, s_y, C_x, C_y) wird für jede Kamera separat durchgeführt. Da die genauen Abbildungseigenschaften auch von der verwendeten Hardware zur Digitalisierung des Videosignals abhängen, wird dabei stets die Kombination aus Objektiv, Kamera und Framegrabber kalibriert. Zur Kamerakalibrierung können sowohl 2D-Eichkörper als auch 3D-Eichkörper verwendet werden. Dreidimensionale Eichkörper liefern bei einer entsprechend großen Ausdehnung exaktere Ergebnisse, haben jedoch den Nachteil, dass sie schwieriger herzustellen und zu transportieren und darüber hinaus auch teurer sind.

Für diese Arbeit wurde eine exakt vermessene Kalibrierplatte mit insgesamt 49 kreisförmigen schwarzen Marken verwendet (siehe Bild 3.8). Die Marken haben einen Radius von 2,5 cm und sind im Abstand von 10 cm angeordnet. Prinzipiell kann die Kalibrierung auf Grundlage eines einzelnen Bildes durchgeführt werden, in dem die Kalibrierplatte vollständig zu sehen ist. Eine wesentliche höhere Genauigkeit lässt sich allerdings durch Multibildkalibrierung erreichen [LZB95]. Hierzu werden etwa 10-20 Bilder aufgenommen, auf denen die Kalibrierplatte in unterschiedlichen Positionen und Abständen von der Kamera zu sehen ist. In den Einzelbildern wird zunächst im Bereich der Kalibrierplatte (heller Bereich mit $N = 49$ dunklen Löchern) eine subpixelgenaue Kontursuche durchgeführt. Nach einer Entzerrung der Konturen gemäß Gleichung (3.14) werden anschließend Position und Größe der N Marken im Bild ermittelt. Aus den Halbmessern und Brennpunkten der detektierten Ellipsen können für jedes Bild j die Lageparameter ${}^{c}R_{w}{}^{j}$ und $\vec{T}^{j}$ der Kalibrierplatte in Bezug auf das Kamerakoordinatensystem geschätzt und als Startparameter für die Multibildkalibrierung verwendet

werden. Die Position der Kalibrierplatte in Bezug auf die Kamera muss also nicht explizit vermessen werden, sondern es genügt, die Lageparameter der Platte aus den gemessenen Ellipsen abzuschätzen.

Bei der anschließenden Multibildkalibrierung wird durch simultane Auswertung der *M* Kalibrierbilder eine Minimierung des Abstandes der projizierten Punkte der Kalibrierplatte und der im Bild gefundenen Ellipsenmittelpunkte durchgeführt. Seien M_i die Koordinaten der *i*-ten Kalibriermarke im Koordinatensystem des Eichkörpers, $m_i^{\,j}(M_i, \vec{x}^{\,j})$ die Projektion von M_i in das *j*-te Bild gemäß den Kameraparametern

$$\vec{x}^{\,j} = ({}^{c}R_{w}^{\,j}, \vec{T}^{\,j}, b, \kappa, s_x, s_y, C_x, C_y)^T \tag{3.16}$$

und $\tilde{m}_i^{\,j}$ die zu M_i korrespondierenden 2D-Koordinaten des extrahierten Ellipsenmittelpunkts. Die nichtlineare Optimierung von

$$\sum_{j=1}^{M} \sum_{i=1}^{N} \| \tilde{m}_i^{\,j} - m_i^{\,j}(M_i, \vec{x}^{\,j}) \|^2 \rightarrow \min! \tag{3.17}$$

wird durch mehrere hintereinandergeschaltete Ausgleichsrechnungen durchgeführt, bei denen die Anzahl der Unbekannten sukzessive erhöht wird und die als Ergebnis die internen Parameter der Kamera liefert. Als Startwerte werden die Angaben des Herstellers verwendet. Weitere Details finden sich in [Lan97] und [LZB95].

3.3.4.2 Externe

Bei der inneren Kalibrierung wurde im Koordinatensystem der Kalibrierplatte gerechnet, und es war möglich, die Lage der Kamera in diesem System aus den im Bild detektierten Ellipsen abzuschätzen. Um auch Messungen durchführen zu können, die sich auf die Umgebung beziehen, muss allerdings auch die Lage der Kameras in Bezug auf ein (frei wählbares) externes Weltkoordinatensystem bekannt sein. Die 6 äußeren Kameraparameter (R_i, T_i), *i=1 ... 6*, die Translation und Rotation der Kamera in Bezug auf das WKS angeben, werden bei der externen Kalibrierung mit Hilfe von mehreren an verschiedenen Stellen in der Umgebung angebrachten Kalibriermarken, auch Passpunkte genannt, ermittelt (siehe Bild 3.8). Die Position des WKS kann frei gewählt werden. In der Regel wird eine Raumecke als Ursprung genommen, und die Koordinatenachsen werden entlang der Wände ausgerichtet. Ähnlich wie bei der inneren Kalibrierung wird in einem Optimierungsverfahren der Abstand zwischen projizierten und im Bild detektierten Marken minimiert. Die Koordinaten der Marken im Bild werden hierbei durch

den Benutzer mit der Maus markiert. Die Vermessung der Marken in Bezug auf das WKS ergibt die Startwerte der Translation, die Rotation der Kamera in Bezug auf das Weltkoordinatensystem wird geschätzt. Im Minimierungsprozess werden lediglich die Werte für R_i und T_i optimiert. Die inneren Kameraparmeter werden konstant gehalten, da sie bei der inneren Kalibrierung schon exakt bestimmt worden sind.

Bei einem Stereoaufbau mit zwei oder mehr Kameras kann die externe Kalibrierung prinzipiell für jede Kamera einzeln nach dem oben beschriebenen Verfahren durchgeführt werden. Um die Genauigkeit der Stereomessungen zu erhöhen, bietet sich alternativ hierzu die Kalibrierung aller weiteren Kameras in Bezug auf die erste (Referenz-) Kamera an.

3.3.5 Umgebungsmodell

Zur Beschreibung des Raumes, in der sich die Kameras und das zu detektierende Objekt befinden, kann die Umgebung durch ein dreidimensionales CAD-Modell beschrieben werden. Die Genauigkeit bzw. Detailtreue der Modellierung hängt von der konkreten Anwendung ab und kann entsprechend angepasst werden. Bild 3.9 zeigt beispielhaft die Beschreibung eines Raumes, bei der neben der Modellierung von Wänden, Türen und Fenstern auch Teile des Inventars, wie z.B. Tische, Schränke und sonstige Gegenstände in das Umgebungsmodell mit einbezogen werden.

Die Modellierung der Wände erfolgt durch die Definition von dreidimensionalen Rhomboiden, die durch einen Eckpunkt $\vec{p}$ und zwei das Rhomboid aufspannende Richtungsvektoren $\vec{r}_1$ und $\vec{r}_2$ beschrieben werden. Darüber hinaus erlaubt das Modell die Definition von Kuboiden. Die Kuboide werden durch einen Eckpunkt $\vec{p}$ sowie drei das Kuboid aufspannende Richtungsvektoren $\vec{r}_1$, $\vec{r}_2$ und $\vec{r}_3$ beschrieben.

Die Modellierung des umgebenden Raumes und des Inventars fließt sowohl in den Interpretationsprozess als auch in die sich daran anschließende Verwendung der Messergebnisse durch die Akteure (*actors*) (siehe Bild 3.1) ein. Durch Wände oder Inventargegenstände wird der Sichtbereich der verwendeten Kameras eingeschränkt, und Objekte können von diesen ganz oder teilweise verdeckt werden. Im Interpretationsprozess wird daher vor dem Bildeinzug zunächst überprüft, ob die dreidimensionalen Suchräume, mit denen die einzelnen Objektmodellteile ausgestattet sind, in den einzelnen Kameras sichtbar sind. Bilder werden nur von den Kameras eingezogen, für die der zugehörige Sichtstrahl weder eine Wand noch einen Inventargegenstand schneidet. Mit Hilfe von Kuboiden können beispielsweise Alarmzonen beschrieben werden, die einen externen Alarmgeber aktivieren, sobald eine Modellinstanz innerhalb dieses Bereiches detektiert wird. Eine weitere Möglichkeit für die Verwendung des Wissens über die Umgebung ist z.B. die Visualisierung der Bewegung von Personen in einem zweidimensionalen Grundriss (Bild 3.9).

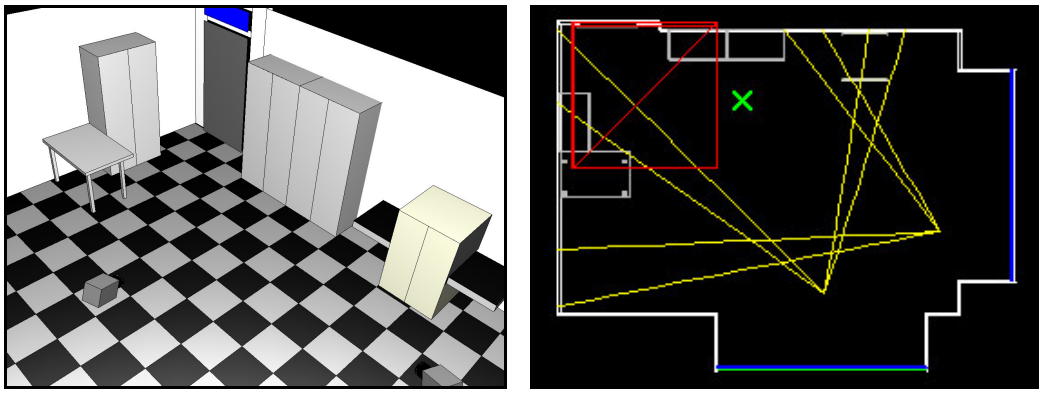


Bild 3.9: ***Umgebungsmodell****. Inventar und Wände werden durch Rhomboide und Kuboide modelliert (links). Im rechten Bild ist die Darstellung in einem zweidimensionalen Grundriss gezeigt. Das Modell erlaubt auch die Definition von Alarmzonen (rot markiert). Gelb eingezeichnet sind die Sichtbereiche der Kameras.*

Mit Hilfe des CAD-Modells können auch komplexere Objekte beschrieben werden, die beispielsweise aus dem verbreiteten DXF-Format konvertiert wurden. Bild 3.2 zeigt exemplarisch die Modellierung von Teilen der Karosserie eines Autos.

3.3.6 Merkmale

In Abschnitt 3.3.2 wurden die innere, die geometrische und die äußere Struktur des Objektmodells und dessen Aufbau aus mehreren Objektmodellteilen omp_i beschrieben. Zur genauen Charakterisierung kann jedem Objektmodellteil ein Merkmal zugeordnet werden, durch das es beschrieben wird und wodurch auch die bei der Segmentierung verwendeten Methoden festgelegt werden.

In STABIL++ wird zwischen drei verschiedenen Merkmalsarten unterschieden. Die Bildsegmentierung liefert zunächst *2D-Bildmerkmale*. Hierbei handelt es sich um geometrische Primitive wie z.B. Ellipsen, Kanten oder Kreisbögen, zu denen zusätzliche Attribute wie beispielsweise Farbe oder auch geometrische Maßzahlen bestimmen werden. Weiterhin ist jedem Bildmerkmal ein Bildpunkt $\vec{p}_{img} = (x, y)^T$ zugeordnet, der die (zweidimensionale) Position im Bild angibt. Die Positionsbestimmung im Bild erfolgt anhand eines oder mehrerer charakteristischer Punkte des Bildmerkmals. Bei der Verwendung eines Konturmodells für die menschliche Silhouette bietet sich beispielsweise der oberste Kopfpunkt an (vgl. Bild 5.1 auf Seite 106), bei der Segmentierung von ellipsenförmigen Regionen im Bild mit Hilfe eines Farbklassifikators kann der Ellipsenmittelpunkt verwendet werden.

Die Bildmerkmale werden mit den bekannten Kalibrierdaten für die Kameras in *3D-Szenenmerkmale* s_i überführt, die ebenfalls durch eine Reihe von Attributen charakterisiert sind und denen darüber hinaus eine (dreidimensionale) Position $\vec{p}_{WKS} = (x, y, z)^T$ im Weltkoordinatensystem zugeordnet ist. Ein Szenenmerkmal kann aus einem oder

mehreren Bildmerkmalen ermittelt werden und umfasst daher zusätzlich das Wissen darüber, aus welchen extrahierten Bildmerkmalen (*extraction data*) es bestimmt worden ist.

Die Charakterisierung der Objektmodellteile selbst erfolgt durch die so genannten *primären Modellmerkmale*. Für das Objektmodellteil omp_i ist das primäre Modellmerkmal f_i durch

$$f_i = \left\langle \{attr^i{}_1, ..., attr^i{}_n\}, ip_i, restr_i, qual_i, \vec{t} \right\rangle \tag{3.18}$$

definiert. Die Attribute $attr^i_j$ entsprechen den Attributen der Szenenmerkmale, sind jedoch eindeutig dem Objektmodellteil zugeordnet. Das Merkmal kapselt in einer Anzahl von Bildverarbeitungsoperatoren ip_i das Wissen darüber, wie seine Segmentierung im Kamerabild erfolgt und welche Restriktionen $restr_i$ dabei anzuwenden sind. Die Restriktionen beeinflussen auch die Qualitätsberechnung durch mehrere Gütefunktionen $qual_i$. Im Interpretationszyklus wird beim *Matching* in einem 3D/3D-Vergleich versucht, die extrahierten Szenenmerkmale den Modellmerkmalen zuzuordnen, um so eine gültige Hypothese für das Objektmodell zu erzeugen. Die Position des Szenenmerkmals $\vec{p}_{WKS}$ muss jedoch nicht mit der Position des Ursprungs des lokalen Koordinatensystems für das Objektmodellteil übereinstimmen, sondern kann zu dieser verschoben sein. Bei der Detektion des obersten Kopfpunktes durch das Silhouettenmodell (vgl. Abschnitt 4.4.6) beispielsweise muss der gefundene Punkt je nach Kopfgröße der Person um etwa 20-25 cm verschoben werden, um auf die Position des Koordinatensystem-Ursprungs für das Objektmodellteil *Kopf* schließen zu können. Diese Verschiebung wird durch den Vektor $\vec{t} = (t_x, t_y, t_z)$ berücksichtigt. Die Definition von $\vec{t}$ erfolgt im lokalen Koordinatensystem des jeweiligen OMP.

Die primären Modellmerkmale f_i dienen zur Lokalisation der einzelnen Objektmodellteile. Auf die Möglichkeit, die OMPs durch zusätzliche, so genannte *sekundäre Modellmerkmale* genauer zu beschreiben, wird hier nicht näher eingegangen, da diese nur zur Verifikation der aufgestellten Hypothesen dienen. Stattdessen sei auf [Rid99] verwiesen. Auf die in dieser Arbeit verwendeten Farb- und Kantenmerkmale wird in Abschnitt 3.5 und in Kapitel 4 näher eingegangen.

3.4 Interpretationsprozess

3.4.1 Bildeinzug und Suchräume

Das in Abschnitt 3.3.1 beschriebene Szenenmodell beinhaltet für jede Szene eine Reihe von Kameras, von denen im ersten Schritt des Interpretationszyklus zunächst ein Bild eingezogen wird. Die Art der Kamera bestimmt dabei die Bildquelle und die Art der Auswertung. Bei *Live*- oder *Memory*-Kameras wird das Videosignal einer Kamera einzeln oder als ganze Sequenz mit einem Framegrabber digitalisiert und in den Hauptspeicher transferiert. Die Auswertung erfolgt dann entweder unmittelbar auf den Einzelbildern (*Live*-Kamera) oder auf der gesamten Bildfolge (*Memory*-Kamera). Alternativ können gespeicherte Bildfolgen mit einer *File*-Kamera von Festplatte eingelesen werden.

Der Bildeinzug erfolgt nur für die Kameras, in denen die gesuchten Objekte voraussichtlich sichtbar sind. Die Sichtbarkeit wird zum einen durch den Sichtbereich der Kameras, des weiteren aber auch durch dreidimensionale Suchräume bestimmt, durch die der Suchbereich eingeschränkt wird. Bei der initialen Detektion werden ein oder mehrere kugelförmige Suchräume an denjenigen Stellen definiert, an denen Objekte erwartet werden. Bei der Überwachung von Räumen bietet sich beispielsweise die Positionierung der initialen Suchräume in der Nähe von Türen oder Fenstern an. Nach erfolgreicher Detektion wird der Suchraum auch durch die Objektmodellinstanzen selber bestimmt. Jedes Objektmodellteil verfügt über einen kugelförmigen (dreidimensionalen) Suchraum, dessen Größe von der Güte der Detektion abhängt und daher von Bild zu Bild variiert. Durch Projektion des 3D-Suchraumes kann der (zweidimensionale) Suchbereich im Bild für das betreffende OMP eingeschränkt und die Suche auf diesen Ausschnitt beschränkt werden. Bild 3.10 zeigt den Suchraum für das Objektmodellteil *Kopf*, wenn im Bild nach hautfarbenen Ellipsen gesucht wird. Die Segmentierung durch Farbklassifikation kann hier auf den eingekreisten Bereich begrenzt werden.

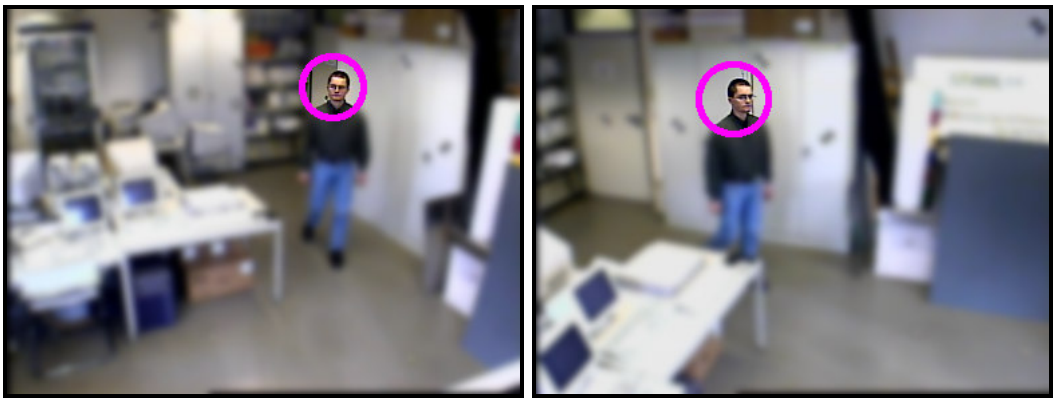


Bild 3.10: ***Projektion des 3D-Suchraumes für das Objektmodellteil Kopf ins Kamerabild****.*

Der Sichtbereich einer Kamera wird weiterhin durch das Umgebungsmodell eingeschränkt (siehe Abschnitt 3.3.5). Befindet sich zwischen dem Suchraum und der Kamera eine Wand oder ein Inventargegenstand, muss von der betreffenden Kamera kein Bild eingezogen werden. Bei der Überwachung mehrer nebeneinander liegender Räume kann die Auswertung bei der Verfolgung einer einzelnen Person so auf die relevanten Bilder beschränkt werden.

3.4.2 Restriktionsgesteuerte Modellsuche

Bei der Bildsegmentierung führen die einzelnen OMPs ihre Extraktion entsprechend der den primären Merkmalen zugeordneten Bildverarbeitungsoperatoren ip_i durch und führen so auf 2D-Bildmerkmale, aus denen durch Mono-Schätzung oder in einer Stereo-Zuordnung die 3D-Szenenmerkmale $\{s_i\}$ ermittelt werden (siehe auch Abschnitt 5.1). In einem 3D/3D-Strukturvergleichsverfahren[12] wird versucht, diese den primären Modellmerkmalen $\{f_i\}$ zuzuordnen und so gültige Hypothesen für das Objektmodell aufzustellen.

Beim Aufbau des modellgetriebenen Interpretationsbaumes wird für jedes OMP versucht, in einem rekursiven Verfahren –ausgehend vom Wurzelelement des hinterlegten Objektmodells– eine gültige Zuordnung seines primären Merkmals zu einem Szenenmerkmal zu finden. Die Tiefe des Interpretationsbaumes ist dabei durch die Anzahl der primären Modellmerkmale bestimmt. Auf jeder Ebene wird versucht, für ein OMP eine gültige Zuordnung seines primären Merkmals zu einem extrahierten Szenenmerkmal zu finden. Die Knoten des Baumes werden also durch Assoziationen

$$assoc_{ij} = \langle s_i, f_j \rangle \tag{3.19}$$

für die Zuordnung des Szenenmerkmals s_i zum Modellmerkmal f_j repräsentiert. Die Breite des Suchbaumes ist daher durch die Anzahl der möglichen Assoziationen bestimmt. Eine Assoziation kann nur aufgestellt werden, wenn die Basisattribute von s_i und f_j übereinstimmen. Für das Modell aus Bild 3.4 beispielsweise, bei dem die Charakterisierung der OMPs durch farbige Marken an den Gelenken erfolgt, werden Szenenmerkmale einer bestimmten Farbe nur Modellmerkmalen mit derselben Farbe zugeordnet. Ziel der Korrespondenzsuche bei der Aufstellung von Hypothesen ist es also, einen Satz gültiger Zuordnungen der 3D-Szenenmerkmale $s = (s_1, ..., s_n)$ zu den 3D-Objektmerkmalen $f = (f_1, ..., f_m)$ zu finden. Einer Hypothese entspricht daher ein

[12] engl. *matching*

vollständiger Pfad im Suchbaum, der sich ausgehend von der ersten Ebene, die das Wurzelelement repräsentiert, bis zu einem Blatt der untersten Ebene erstreckt.

Die Zeit, die für die Hypothesengenerierung benötigt wird, hängt entscheidend von der Größe des Suchbaumes, d.h. von der Anzahl der möglichen Assoziationen, ab. Durch die Verwendung verschiedener Basisattribute kann die Größe stark eingeschränkt werden. Bezeichne k die Anzahl der verschiedenen Basisattribute (z.B. verschiedene Farben), n_i die Anzahl der primären Merkmale des Objektmodells mit demselben Basisattribut i und m_i die Anzahl der Szenenmerkmale mit Basisattribut i. Die maximale Anzahl von Hypothesen berechnet sich dann gemäß

$$|H| = \prod_{i=1}^{k} \frac{m_i!}{(m_i - n_i)!} \tag{3.20}$$

Das Objektmodell in Bild 3.4 wird durch 14 primäre Merkmale charakterisiert. Bei Verwendung von nur einem Basisattribut (d.h. nur einer einzigen Farbe) könnten Assoziationen zwischen beliebigen s_i und f_j aufgestellt werden, so dass bei ebenfalls 14 extrahierten Szenenmerkmalen ($k = 1,\ n = 14,\ m = 14$) zunächst insgesamt $14! = 8{,}7 \cdot 10^{10}$ Hypothesen aufgestellt werden könnten. Durch die Verwendung der 4 verschiedenen Farben blau, magenta, grün und cyan ($k = 4,\ n_i = m_i = \{4,4,4,2\}$) kann die Anzahl um einen Faktor $3{,}2 \cdot 10^6$ auf $4! \cdot 4! \cdot 4! \cdot 2! = 27648$ eingeschränkt werden.

Eine weitere Einschränkung ergibt sich durch die Berücksichtigung von *Restriktionen* bei der Korrespondenzsuche. Je nach Anzahl der Assoziationen, die dabei berücksichtigt werden, lässt sich eine Einteilung in unäre (eine Assoziation), binäre (zwei Assoziationen) und tertiäre Restriktionen (drei Assoziationen) vornehmen. Bei der Interpretation werden Restriktionen auf Merkmalsebene und Modellebene berücksichtigt. Eine merkmalsbasierte Restriktion ist beispielsweise die Forderung, dass der gefundene Punkt für ein Objektmodellteil in dessen dreidimensionalem Suchraum liegen muss. Für farbige Ellipsen müssen darüber hinaus beispielsweise Größe und Form der extrahierten Bildmerkmale in definierten Grenzen liegen. Aufgrund der inneren und geometrischen Objektmodellstruktur ergeben sich weiterhin Einschränkungen für den Abstand der gefundenen 3D-Punkte, der z.B. der Knochenlänge des menschlichen Skeletts entspricht, oder auch Beschränkungen der Gelenkwinkel. Für eine vollständige Übersicht der verwendeten Restriktionen und deren Einfluss auf den Interpretationsprozess sei auf [Rid99] verwiesen.

Durch die Verwendung verschiedener Basisattribute und durch die Berücksichtigung geeigneter Restriktionen wird die Anzahl der möglichen Assoziationen also stark eingeschränkt, und eine vollständige Traversierung des Interpretationsbaums kann vermieden werden.

3.5 Farbe als Merkmal

Das in Abschnitt 3.3.2 beschriebene Objektmodell erlaubt es, jedes OMP mit einem Merkmal (*feature*) auszustatten, welches es charakterisiert. Nach dem Bildeinzug wird für jede Kamera cam_i, in der das OMP sichtbar ist, mit Hilfe geeigneter Bildverarbeitungsoperatoren eine Merkmalsextraktion durchgeführt, um so das OMP im Bild zu lokalisieren. Die zum Einsatz kommenden Operatoren hängen dabei von dem verwendeten Merkmal ab.

Im Rahmen des Systems STABIL++ wurde bislang hauptsächlich das Merkmal *Farbe* verwendet. Ein stochastischer Farbklassifikator nimmt eine Segmentierung der Bildpixel in n Farbklassen Ω_k sowie eine Rückweisungsklasse Ω_0 vor. Zur Detektion von Personen kann z.B. nach hautfarbenen Bereichen im Bild gesucht werden (Farbe *skin*), für die Analyse menschlicher Bewegungen werden die Gelenke mit farbigen Marken versehen (siehe Bild 3.2 und Bild 3.12).

Das farbliche Aussehen eines Objektes im Bild hängt von vielen verschiedenen Einflussfaktoren ab. Der letztlich mit einem CCD-Chip gemessene Farbwert wird beeinflusst von Hardwareeigenschaften (Charakteristik des CCD-Chips, A/D-Wandlung), Objekteigenschaften (Reflexionen, Orientierung) und den zum Aufnahmezeitpunkt herrschenden Umgebungsbedingungen (Anzahl der Lichtquellen, spektrale Zusammensetzung). Bei der Analyse von Bildfolgen variieren die genannten Parameter darüber hinaus auch noch zeitlich. Eine Modellierung all dieser Parameter und deren gegenseitigen Abhängigkeiten ist wegen der Vielzahl der relevanten Einflüsse nicht möglich. Um dennoch quantitative Aussagen machen zu können, wird die Farbe eines Objektes daher als Wahrscheinlichkeitsverteilung $p(\vec{c} \mid \Omega_k)$ modelliert, die die Verteilung des Farbvektors $\vec{c}$ für die Farbklasse Ω_k in einem geeigneten Farbraum beschreibt.

Der verwendete Farbraum sowie der Bayes‘sche Klassifikator werden in den folgenden Abschnitten beschrieben.

3.5.1 Klassifikation im i_2i_3 - Farbraum

Bei der Segmentierung eines Bildes durch Farbklassifikation wird für jedes Bildpixel entschieden, ob es zu einer der n Farbklassen Ω_k oder zur Rückweisungsklasse Ω_0 gehört. Um ein gutes Klassifikationsergebnis zu erhalten, muss die interne Darstellung der Daten so gewählt werden, dass sich die Farbklassen möglichst gut separieren lassen. Die Farbwerte, die meist als dreikanalige RGB-Werte vorliegen, müssen daher geeignet transformiert werden, um Korrelationen zwischen den Kanälen zu eliminieren. Eine Möglichkeit hierzu bietet die Karhunen-Loeve Transformation (KL-Transformation), die auch als Hauptachsentransformation (*principal component analysis*, PCA) bezeichnet wird [Hot33][Cas96], (siehe auch Abschnitt 4.3.2).

Ohta et al. [OKS80] und Hafner [Haf99] haben durch die empirische Auswertung einer Vielzahl von unterschiedlichen Bildern gezeigt, dass die Eigenvektoren der Kovarianzmatrix der Farbwerte nahezu unabhängig vom verwendeten Bildmaterial sind. Die Untersuchung einer Reihe von Standard-Farbräumen zeigte, dass die durch

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.21}$$

beschriebene Transformation der RGB-Werte in den $I_1I_2I_3$-Raum eine gute Separierung der Farbklassen ermöglicht und darüber hinaus ein schnelles Umrechnen zwischen den beiden Farbräumen erlaubt.

Der I_1-Kanal kodiert die Helligkeitsinformation und ist für die Farbsegmentierung nicht von Bedeutung. Für die Klassifikation werden daher nur die Komponenten I_2 und I_3 verwendet, die durch

$$i_2 = \frac{I_2}{R+G+B} \quad , \quad i_3 = \frac{I_3}{R+G+B} \tag{3.22}$$

normiert werden und den zweidimensionalen i_2i_3-Farbraum ergeben. Die Lage der Farben im i_2i_3-Raum zeigt Bild 3.11. Wenn nicht explizit anders angegeben, beziehen sich sämtliche Angaben zur Farbklassifikation in dieser Arbeit auf die Klassifikation im i_2i_3-Farbraum.

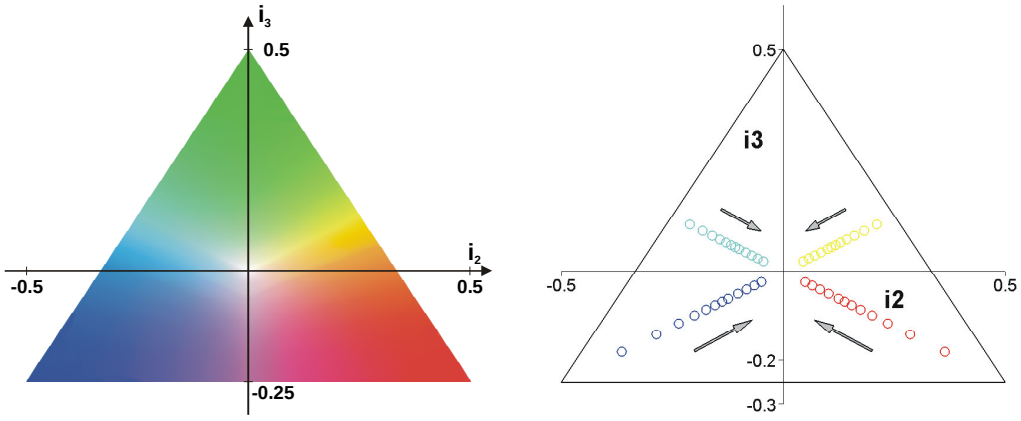


Bild 3.11: ***Der i_2i_3-Farbraum****. Links: Lage der Farben im i_2i_3-Farbraum. Rechts: schematische Darstellung der Verschiebung der Farbklassen in Richtung des Nullpunktes beim Öffnen der Blende.*

Empirische Untersuchungen einer Vielzahl von Bildern legen die Modellierung der Wahrscheinlichkeitsdichte durch eine *m*-dimensionale Gauß´sche Normalverteilung nahe [Haf99]:

$$p(\vec{c} \mid \Omega_k) = \frac{1}{(2\pi)^{n/2}\sqrt{\det C_k}} \exp(-\frac{1}{2}(\vec{c} - \bar{\mu}_k)^T C_k^{-1}(\vec{c} - \bar{\mu}_k)) \qquad (3.23)$$

Die Verteilung des *m*-dimensionalen Merkmalsvektors $\vec{c}$ wird für jede der *n* Farbklassen Ω_k durch den Mittelwert $\bar{\mu}_k$ und die Kovarianzmatrix C_k *(k=1..n)* charakterisiert. Zusätzlich zu den Farbklassen Ω_k *(k=1..n)* wird eine Rückweisungsklasse Ω_0 definiert, in die bei der Segmentierung alle Pixel klassifiziert werden, die nicht zu einer der Farbklassen gehören.

Der so definierte Klassifikator muss zunächst in einem überwachten Lernvorgang trainiert werden, um die Mittelwerte, Kovarianzmatrizen und a priori Wahrscheinlichkeiten p_k der einzelnen Farbklassen zu bestimmen. Prinzipiell wäre es möglich, für jede Farbklasse einen separaten Klassifikator zu trainieren. Zur Vermeidung von rechenintensiven Bildverarbeitungsoperationen bei der anschließenden Nachbearbeitung empfiehlt es sich jedoch, statt mehrerer unabhängiger Klassifikatoren mit je einer Farbklasse einen Klassifikator mit mehreren Klassen zu verwenden und die Zuordnung eines Pixels nach der Bayes´schen Entscheidungstheorie vorzunehmen [HM96][Haf99].

Hierzu werden zunächst die mit den a priori Wahrscheinlichkeiten p_i gewichteten Wahrscheinlichkeiten

$$p_i\, p(\vec{c} \mid \Omega_i) \qquad (3.24)$$

gebildet und mit

$$k = \arg\max_{i=1\ldots n} p_i\, p(\vec{c} \mid \Omega_i) \qquad (3.25)$$

die Klasse *k* ermittelt, zu der der Merkmalsvektor $\vec{c}$ am wahrscheinlichsten gehört. Die Zuordnung eines Pixels zu einer Klasse Ω_k mit *k=1...n* oder zur Rückweisungsklasse Ω_0 geschieht durch die Entscheidungsregel

$$\begin{aligned} &\delta(\Omega_k \mid \vec{c}) = 1 \quad \textit{falls} \quad p_k\, p(\vec{c} \mid \Omega_k) \geq \beta \sum_{i=1}^{n} p_i\, p(\vec{c} \mid \Omega_i) \\ &\delta(\Omega_0 \mid \vec{c}) = 1 \quad \textit{sonst} \end{aligned} \tag{3.26}$$

Der Rückweisungsparameter $\beta \in [0,1]$ wird aus den Kosten für Falschklassifikation, Rückweisung und korrekte Klassifikation gebildet. Details hierzu finden sich in [Haf99][HM96]. Durch β wird der Anteil an Pixeln festgelegt, der an den Klassenübergängen zurückgewiesen wird. Eine Rückweisungsklasse ist so mathematisch exakt definiert. Effekte lassen sich für Werte $\beta > 0{,}5$ beobachten.

Bei der Verwendung der Farbklassifikation zur Auswertung von Bildfolgen ist es notwendig, den Klassifikator an die sich im Lauf der Zeit verändernden Bedingungen wie z.B. Beleuchtungsänderungen zu adaptieren. Auf das dazu verwendete Verfahren des entscheidungsüberwachten Lernens wird hier nicht näher eingegangen, da es für diese Arbeit nicht relevant ist. Statt dessen sei auf die entsprechende Literatur verwiesen [Nie83][Haf99].

3.5.2 Grenzen der Farbklassifikation

Bei der Auswertung von Videobildfolgen kommt es zwischen zwei aufeinanderfolgenden Bildern oft nur zu graduellen Veränderungen in der Erscheinung einer bestimmten Farbe. Der oben erwähnte Ansatz des entscheidungsüberwachten Lernens zur Adaption der Parameter des Farbklassifikators führt unter kontrollierbaren Umgebungsbedingungen zu robusten Klassifikationsergebnissen und erlaubt die Verfolgung von Objekten in einer Sequenz. Für Umgebungen, in denen Umgebungsparameter wie beispielsweise die Beleuchtung kontrolliert werden können (z.B. in Innenräumen oder im Labor) und unter der Voraussetzung, dass das Gesicht der verfolgten Person stets sichtbar ist, reicht der bislang verwendete Ansatz daher aus. Schwierig ist jedoch die Auswertung von Bildern bei stark schwankenden Umgebungsbedingungen (z.B. in Außenbereichsanwendungen) oder wenn die verwendeten Kameras nicht hochwertig genug sind. Folgende Faktoren tragen dazu bei, dass die Segmentierung von Bildern allein auf Grundlage des Merkmals „Farbe“ nur unzureichende Ergebnisse liefert:

- Die Farbklassifikation ergibt bestmögliche Ergebnisse, wenn die zu untersuchende Szene gut ausgeleuchtet ist. Für schwach ausgeleuchtete Umgebungen wird der Rauschpegel der Eingabebilder jedoch sehr hoch, und die Segmentierung führt zu unbefriedigenden Ergebnissen. Eine schwache Ausleuchtung führt dazu, dass sich dunkle Pixel, die nahezu schwarz sind und deren RGB-Werte etwa 0 sind, vom Nullpunkt aus in sämtliche Richtungen im i_2i_3-Raum verstreuen und so fälschlicher-

weise als „farbig“ erkannt werden. Die Farbinformation geht bei dunklen Bildern verloren („In der Nacht sind alle Katzen grau“).

- Nur durch die Verwendung hochwertiger Hardware (Kameras, Framegrabber, etc.) lässt sich der gesamte Dynamikbereich ausnutzen. Bei Verwendung von Kameras mit entsprechend geringerer Qualität tragen Effekte wie z.B. Clipping zu einem verfälschten Ergebnis bei.
- Die FBAS[13]-Übertragung im Videobereich liefert nur sehr wenig Farbinformation. Am besten wäre daher die Verwendung hochwertiger RGB-Kameras, die jedoch teuer sind.
- Bei Beleuchtungsänderungen der Szene kann die Kontinuität der Adaption gestört werden. Es kommt daher u.U. zu einer Verschiebung der Klassen im Farbraum. Einen ähnlichen Effekt haben sich stark ändernde Bildinhalte, da der Klassifikator mit bestimmten a priori Wahrscheinlichkeiten trainiert wurde, sich diese aber durch geänderte Bildinhalte drastisch verschieben können.
- Bei Variation der Blende ist ein Wandern der Farbklassen im i_2i_3-Raum zu beobachten. Beim Öffnen verschieben sich die Farbklassen dabei nach innen in Richtung des Nullpunktes (siehe Bild 3.11). Die durch die Gleichungen (3.21) und (3.22) beschriebene Eliminierung der Helligkeitsinformation ist in der Praxis nicht immer gegeben.

Bild 3.12 zeigt beispielhaft den Effekt, den die Variation der Szenenbeleuchtung auf das Ergebnis der Klassifikation hat. Der Klassifikator wurde bei den im linken Bild gezeigten Beleuchtungsverhältnissen mit $n=4$ Farbklassen trainiert. Unter diesen definierten Bedingungen führt eine Klassifikation der Bildpixel zu guten Resultaten. Die zu einer Klasse gehörenden Pixel sind farbig markiert (grün ⇔ schwarz markiert, blau ⇔ rot markiert, magenta ⇔ grün markiert, cyan ⇔ weiß markiert). Bei Ausleuchtung derselben Szene mit natürlichem (Sonnen-)Licht kommt es jedoch zu einer Reihe von Fehlklassifikationen: Cyan beispielsweise wird überhaupt nicht mehr gefunden, rot wird fälschlicherweise der Farbklasse magenta zugeordnet.

Die beschriebenen Punkte führen dazu, dass ein einmal trainierter Klassifikator nur in einem relativ geringen Variationsbereich der Umgebungsbedingungen eingesetzt werden kann und bei geänderten Bedingungen entsprechend angepasst, also neu trainiert werden muss. Aufgrund der schwierigen Bedingungen beispielsweise bei Anwendungen im Außenbereich ist es daher wünschenswert, neben dem Merkmal „Farbe“ weitere

[13] Bildsignal zur Übertragung von Farbbildern. Farb- und Helligkeitsinformationen sowie Synchronisationssignale werden auf einer gemeinsamen Signalleitung übertragen. Das FBAS-Signal wird auch als Composite Video Signal bezeichnet.

Merkmale zu definieren, die eine robuste Interpretation auch bei stark schwankenden Umgebungsbedingungen erlauben.

Hierzu eignen sich insbesondere Merkmale, die nicht auf der Eigenschaft *Farbe* beruhen, sondern nach Kanten im Bild suchen. Hierdurch wird man einerseits unabhängiger von der Beleuchtung einer Szene, zum anderen wird durch die Verwendung farbunabhängiger Merkmale auch die Auswertung von Grauwertbildern möglich.

Die im Rahmen dieser Arbeit entwickelten kantenbasierten Merkmale und deren Einsatz zur Detektion formvariabler Objekte werden in den folgenden Kapiteln vorgestellt.

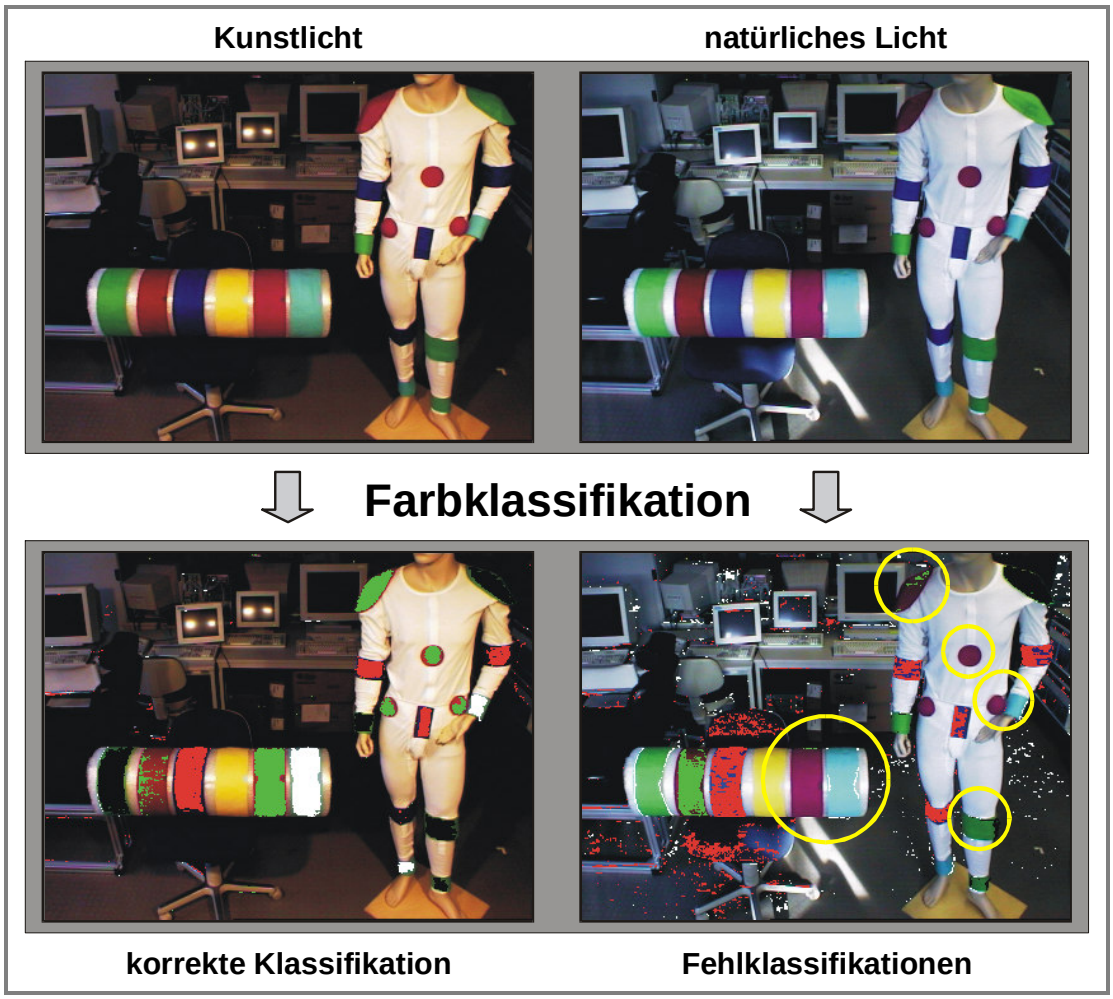


Bild 3.12: ***Einfluss der Szenenbeleuchtung auf das Ergebnis der Farbklassifikation****. Der Klassifikator wurde bei der Beleuchtung mit Kunstlicht (links) trainiert und führt unter diesen definierten Bedingungen zu korrekten Klassifikationsergebnissen (die zu einer Farbklasse gehörenden Pixel sind farbig markiert, z.B. weiße Markierung für Klasse cyan etc.). Bei Beleuchtung der Szene mit natürlichem Tageslicht (rechts) kommt es zu Fehlklassifikationen: Einige Farben werden nur noch teilweise oder gar nicht mehr gefunden, rot wird fälschlicherweise als magenta erkannt.*

4 Kantenbasierte Modellierung

Im vorhergehenden Kapitel wurden die Grenzen der Bildanalyse durch Farbsegmentierung aufgezeigt. Um eine weitgehende Unabhängigkeit von den Umgebungsbedingungen zu erreichen, müssen für eine robuste Objektdetektion verschiedene, sich ergänzende Verfahren zur Segmentierung verwendet werden. In diesem Kapitel werden daher kantenbasierte Modelle und Suchverfahren entwickelt, mit denen formvariable Objekte in Videosequenzen verfolgt werden können. Nach einer kurzen Einführung in die Kantenextraktion und die Theorie der Punktverteilungsmodelle (PDMs) werden zwei Modelle für die Silhouette des menschlichen Oberkörpers und das Gesicht vorgestellt. Diese kompakten statistischen Modelle ermöglichen zusammen mit den im abschließenden Kapitel erarbeiteten Suchverfahren und Algorithmen die Personenverfolgung innerhalb von STABIL++ und die Vermessung der Gesichtszüge.

4.1 Charakterisierung von Kanten und Kantendetektoren

Kanten repräsentieren physikalische oder geometrische Änderungen in einer Szene oder innerhalb eines Objektes und führen daher zu einer Änderung der Grauwerte im Bild, die durch Anwendung geeigneter Differentialoperatoren bestimmt werden kann. Die Charakterisierung von Kanten durch Maxima der ersten Ableitung bzw. Nulldurchgänge der zweiten Ableitung führt im einfachsten Fall auf die bekannten Sobel- bzw. Laplacefilter. Da die Ableitungen empfindlich auf Rauschen im Bild reagieren, sollten die Originalbilder vor der eigentlichen Kantendetektion mit einem geeigneten Filter geglättet werden, um so hochfrequente Anteile zu eliminieren. Der allgemeine Aufbau eines Kantendetektors aus Tiefpassfilter mit anschließender Gradientenbildung ist in Bild 4.1 dargestellt.

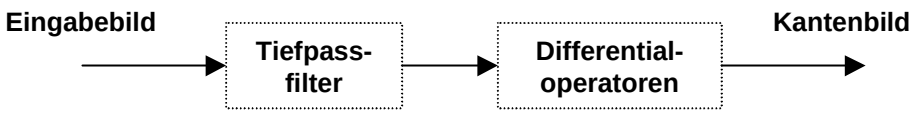


Bild 4.1: ***Allgemeiner Aufbau eines Kantendetektors****. Nach Glättung des Bildes zur Unterdrückung von Rauschen besteht die eigentliche Kantenextraktion aus der Anwendung geeigneter Differentialoperatoren.*

Der eigentlichen Kante, deren eindimensionaler Grauwertverlauf im Folgenden mit $g(x)$ bezeichnet wird, ist ein vom Kantenverlauf unabhängig angenommenes Rauschen $n(x)$ überlagert, das meist als weißes Rauschen mit den Erwartungswerten $E[n(x)]=0$ und

$E[n^2(x)]=\sigma^2$ modelliert wird. Bei der Tiefpassfilterung, für die häufig ein Gauß'scher Filter verwendet wird, wird das Bild mit einer geeigneten Funktion $s(x)$ gefaltet. Der geglättete Verlauf $g_0(x)$ setzt sich daher aus einem Anteil aus dem unverrauschten Kantensignal und einem Beitrag, der durch das Rauschen erzeugt wird, zusammen:

$$g_0(x) = g_N(x) * s(x) = \{g(x) + n(x)\} * s(x) \tag{4.1}$$

Der Operator * bedeutet hier die Faltung zweier Funktionen.

In der Literatur finden sich verschiedene Modelle für die Beschreibung von $g(x)$. Ein einfaches Modell beschreibt eine einzelne Kante an der Stelle x_0 durch einen Sprung des Grauwertes bei x_0 und eine gleichmäßige Änderung der Grauwerte rechts und links von x_0. Dieses Modell der asymmetrischen Stufenkante

$$g(x) = \begin{cases} k_1 x & , \quad x < x_0 \\ A/2 & , \quad x = x_0 \\ k_2 x + A & , \quad x > x_0 \end{cases} \tag{4.2}$$

ergibt für $x_0 = k_1 = k_2 = 0$ das mit Abstand am häufigsten verwendete Modell der symmetrischen Stufenkante bei $x = 0$. Werte $k_i \neq 0$ führen beim Laplace-Operator zu Ungenauigkeiten in der Lokalisation, wobei die Größe des Fehlers proportional zu $\Delta k=k_2-k_1$ ist [UM90]. Modelle, die nicht nur eine einzelne Kante beschreiben, sondern von mehreren Kanten in einem Intervall ausgehen, modellieren $g(x)$ meist als stationären stochastischen Prozess. Ein Beispiel hierfür wird z.B. von Shen und Castan [SC92] beschrieben.

Zum Auffinden von Bildkanten wurden verschiedene Verfahren entwickelt. Ein optimaler Kantendetektor muss eine Kante zuverlässig finden, d.h. die Wahrscheinlichkeit, eine tatsächlich vorhandene Kante zu detektieren, muss möglichst groß sein. Andererseits muss die Wahrscheinlichkeit, dass ein Grauwertverlauf, der keiner Kante entspricht, jedoch als solche detektiert wird, möglichst klein sein. Weiterhin sollte die vom Detektor gelieferte Position mit der tatsächlichen Position möglichst gut übereinstimmen. Das Detektorsignal sollte also bei x_0 ein lokales Maximum haben. Die erste Anforderung führt auf das Signal-Rausch-Verhältnis (*signal to noise ratio*, SNR), die zweite Anforderung liefert einen quantitativen Ausdruck für die Lokalisation *LOC* [Fau93][Can86].

Für die bei $x=0$ zentrierte symmetrische Stufenkante lassen sich *SNR* und *LOC* leicht aus der Impulsantwort $f(x)$ des Kantendetektors berechnen. Sie sind bis auf Konstanten,

die von der Höhe der Stufe und der Stärke des Rauschens abhängen, proportional zu folgenden Ausdrücken:

$$SNR \propto \frac{\int_{-\infty}^{0} f(x)dx}{\sqrt{\int_{-\infty}^{+\infty} f^2(x)dx}} \quad , \quad LOC \propto \frac{|f'(0)|}{\sqrt{\int_{-\infty}^{+\infty} f'^2(x)dx}} \tag{4.3}$$

Die meisten Ansätze zur Kantendetektion kombinieren *SNR* und *LOC* in geeigneter Weise zu einer Funktion, die mit Hilfe von Methoden aus der Variationsrechnung maximiert wird und so auf unterschiedliche Funktionen *f(x)* führt. Die Ansätze unterscheiden sich weiterhin darin, wie die Kriterien mit zusätzlichen Restriktionen kombiniert werden. Unterschiede ergeben sich weiterhin für Ansätze mit beschränkter oder unbeschränkter Fenstergröße des Faltungskerns. Der Ansatz von Canny [Can86] beispielsweise benutzt eine beschränkte Ausdehnung des Faltungskernes, während Shen und Castan in [SC92] einen Filter mit unbegrenzter Ausdehnung vorstellen, der optimal für Einzel- und Mehrfachkanten ist.

Nach der Extraktion der Bildkanten muss ein geeignetes Modell, das das zu suchende Objekt beschreibt, an die Bildstrukturen angepasst werden. Nach einer kurzen Zusammenfassung der Anforderungen, die ein Modell zur Beschreibung flexibler Objekte erfüllen muss, werden die in dieser Arbeit verwendeten Punktverteilungsmodelle in Abschnitt 4.3 eingeführt.

4.2 Anforderungen an das Modell

Das Aussehen von Personen in Videobildern variiert stark. Ein Modell, mit dem flexible Objekte wie beispielsweise die Silhouette von Personen oder das Gesicht in Bildfolgen detektiert und verfolgt werden können, darf daher nicht starr sein, sondern muss sämtliche auftretenden Formvariationen wiedergeben können. Es muss die verschiedenen Haltungen und Mimiken einer Person sowie das unterschiedliche Aussehen wiedergeben können, das sich aus verschiedenen Ansichten ergibt. Außerdem müssen auch Unterschiede zwischen verschiedenen Individuen modelliert werden. Die Bandbreite der Formen, die durch das Modell beschrieben werden, darf andererseits nur so groß sein, dass auch tatsächlich nur Personen bzw. Gesichter im Bild gefunden werden und keine anderen Objekte, d.h. entartete Formen müssen vermieden und vom Modell ausgeschlossen werden.

Zur Beschreibung der charakteristischen Silhouette von Personen bietet sich die Kopf-Schulter Partie an. Modelliert werden müssen hierbei die verschiedenen Größenproportionen unterschiedlicher Personen, die verschiedenen Ansichten sowie Bewegungen beispielsweise der Arme und des Kopfes.

Zur Erfüllung der genannten Anforderungen werden in dieser Arbeit die menschliche Silhouette und das Gesicht durch ein Punktverteilungsmodell (*point distribution model*, PDM) modelliert (siehe Bild 4.2). Diese Technik zur Erstellung kompakter statistischer Modelle von flexiblen Objekten wird im folgenden Abschnitt näher erläutert.

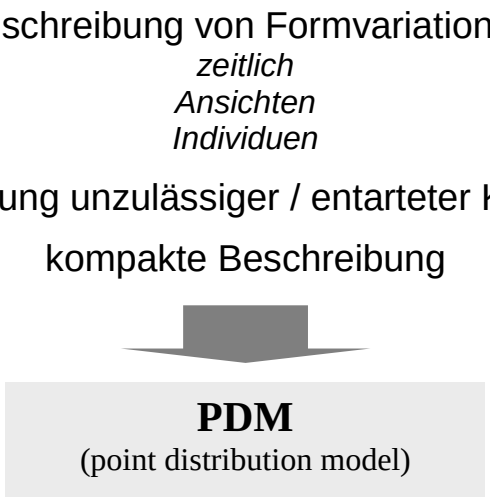


Bild 4.2: ***Anforderungen an ein Modell zur Beschreibung flexibler Objekte.***

4.3 Theorie der Punktverteilungsmodelle

4.3.1 Überblick

Die Technik, flexible Objekte mit Hilfe von Punktverteilungsmodellen (*point distribution models*, PDMs) zu beschreiben, wurde 1992 von Cootes et al. vorgestellt [CTC92]. Das zu modellierende Objekt wird durch eine Menge von Koordinaten repräsentiert, welche die Positionen charakteristischer Punkte der Kontur beschreiben. Aus einer hinreichend großen Anzahl von repräsentativen Trainingsbildern werden Beispiele für die verschiedenen Formen des zu modellierenden Objektes extrahiert. Jede Kontur S_i, die durch *n* Punkte beschrieben sei, kann als Vektor in einem *2n*-dimensionalen Koordinatenraum $\mathbb{R}^{2n}$ aufgefasst werden. Umgekehrt repräsentiert jeder *2n*-dimensionale Vektor eine Kontur, deren Punkte durch die Koordinaten definiert sind. Da die Form der Trainingskonturen zwar ähnlich ist, andererseits aber dennoch von Bild zu Bild variiert, bilden die Trainingskonturen im $\mathbb{R}^{2n}$ eine Punktwolke.

Durch eine Hauptachsentransformation (***p**rincipal **c**omponent **a**nalysis*, PCA) [Cas96] [CTC92] wird eine statistische Analyse dieser Punktwolke und somit der Punktkoordinaten vorgenommen (siehe Abschnitt 4.3.2). Man erhält so ein Modell, das durch eine mittlere Kontur $\bar{S}$ sowie einen Satz von *2n* orthogonalen Eigenvektoren $\vec{p}_k$ definiert wird. Durch Linearkombination der Eigenvektoren können neue Konturen erzeugt werden, die ähnlich zu den trainierten sind.

Um die PCA anwenden zu können, müssen die aus den Bildern extrahierten Trainingskonturen zunächst so zueinander ausgerichtet werden, dass alle Variationen, die nicht von einer Änderung der Form herrühren –d.h. Translation, Rotation und Skalierung– eliminiert werden. Mit der Methode der kleinsten Quadrate (*least squares* Ansatz) wird die Varianz der einzelnen Punktpositionen minimiert. Die Ausrichtung von Konturen (*shape alignment*) ist in Anhang B beschrieben.

Wichtig für die sinnvolle Anwendbarkeit der Hauptachsentransformation ist weiterhin, dass die Nummerierung der Konturpunkte in allen Trainingsbildern gleich ist und dass die Konturpunkte in allen Bildern die gleichen charakteristischen Objektpunkte beschreiben. Für ein Modell des menschlichen Oberkörpers sind dies z.B. der oberste Kopfpunkt, rechte und linke Schulter oder die beiden Ellenbogen (vgl. Abschnitt 4.4.6), für ein Gesichtsmodell beispielsweise die Mund- oder Augenwinkel (siehe Abschnitt 4.4.7).

Oft kann der größte Teil der Formvariation durch *t* wenige (*t«2n*) Eigenvektoren beschrieben werden. Die Eigenvektoren mit den betragsmäßig größten Eigenwerten beschreiben die Hauptachsen, entlang denen die Punktwolke die größte Ausdehnung hat. Je kleiner der zu einem Eigenvektor gehörende Eigenwert ist, desto geringer ist auch der Effekt auf die Formvariation. Eigenvektoren mit kleinen Eigenwerten können daher vernachlässigt werden. Das so erhaltene PDM stellt ein Modell dar, mit dem flexible Objekte kompakt repräsentiert werden können.

Bei der Auswertung der Trainingsbilder werden die Konturpunkte manuell definiert. Der Experte wird hierbei jedoch von einem intelligenten Tool unterstützt, das es erlaubt, Kantenverläufe in einem Bild halbautomatisch zu extrahieren. Der hierzu verwendete Algorithmus *intelligent scissors* (engl. intelligente Schere) [MB95][BM96] wird in Abschnitt 4.4.2 beschrieben. Außer der automatisierten Kantenextraktion wird weiterhin die genaue Positionierung der Konturpunkte auf dem Kantenverlauf durch eine Snap-Funktion erleichtert, und die äquidistante Interpolation zwischen den so definierten charakteristischen Landmarken ist automatisiert (vgl. Abschnitte 4.4.2 und 4.4.4).

Das erzeugte PDM kann anschließend zur Auswertung von Videobildfolgen eingesetzt werden (siehe Kapitel 5). Hierzu wird zunächst eine initiale Kontur S_{ini} an die Stelle im Bild projiziert, an der das zu detektierende Objekt erwartet wird. In einem iterativen Verfahren werden Position und Form der Kontur so lange angepasst, bis der Kantenverlauf im Bild hinreichend gut approximiert wird. Die Formänderung ist dabei immer konsistent mit dem Modell, d.h. die Bildkanten werden zwar so gut wie möglich appro-

ximiert, die gefundene Kontur nimmt aber immer nur Formen an, die noch zulässig sind, so dass entartete Formen vermieden werden.

4.3.2 Statistik des Trainingsdatensatzes

Gegeben sei eine Menge von *N* zueinander ausgerichteten (vgl. Anhang B) Trainingskonturen S_i. Jede dieser Konturen kann durch einen *2n*-dimensionalen Vektor beschrieben werden, der die *x*- und *y*- Koordinaten der *n* Konturpunkte enthält:

$$S_i = (x_{i1}, y_{i1}, \ \ldots \ , x_{in}, y_{in})^T \tag{4.4}$$

mit $i = 1 \ldots N$.

Die Menge der S_i bildet eine Punktwolke im $\mathbb{R}^{2n}$. Aus der mittleren Kontur

$$\bar{S} = \frac{1}{N} \sum_{i=1}^{N} S_i \tag{4.5}$$

und den einzelnen Trainingskonturen berechnet sich die zugehörige Kovarianzmatrix zu

$$C = \frac{1}{N} \sum_{i=1}^{N} dS_i dS_i^{\ T} \tag{4.6}$$

mit $dS_i = S_i - \bar{S}_i$.

Zur Ermittlung der orthogonalen Hauptachsen ist das Eigenwertproblem

$$C\vec{p}_k = \lambda_k \vec{p}_k \tag{4.7}$$

für $k = 1 \ldots 2n$ zu lösen.

Man erhält so ein System von *2n* paarweise zueinander orthogonalen Eigenvektoren $\vec{p}_k$ und zugehörigen Eigenwerten λ_k, wobei die Vektoren so angeordnet seien, dass $\lambda_k \geq \lambda_{k+1}$ für $k=1...2n-1$. Die Normierung wird so gewählt, dass $| \vec{p}_k | = \vec{p}_k^{\ T} \vec{p}_k = 1$ ist.

Die Eigenvektoren mit den größten Eigenwerten zeigen in die Richtungen, in der die analysierte Punktwolke die größte Ausdehnung hat, und der Betrag der zugehörigen Eigenwerte entspricht gerade die Varianz entlang dieser Richtungen.

4.3.3 Erzeugung neuer Konturen

Oft weisen die Trainingsdaten aufgrund linearer Abhängigkeiten entlang einiger Hauptachsen nur eine sehr geringe Variation auf, so dass die Formänderung der Konturen durch wenige ($t \ll 2n$) Eigenvektoren approximiert werden kann. Dabei wird t meist so gewählt, dass die Varianz

$$\lambda_t = \sum_{k=1}^{t} \lambda_k \tag{4.8}$$

die durch die ersten t Eigenvektoren beschrieben wird, einen bestimmten Anteil der totalen Varianz

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k \tag{4.9}$$

beschreibt (z.B. 95%)(vgl. Abschnitt 4.4.6 und 4.4.7). Die zugehörenden Eigenvektoren werden in einer Matrix P zusammengefasst:

$$P := (\vec{p}_1\sqrt{\lambda_1} \;\; \ldots \;\; \vec{p}_t\sqrt{\lambda_t}) \tag{4.10}$$

Mit Hilfe der mittleren Kontur $\bar{S}$ und den Eigenvektoren können durch Linearkombination neue Konturen S erzeugt werden, indem in einem Parametervektor $\vec{b} \in \mathbb{R}^t$ angegeben wird, wie weit die neue Kontur von $\bar{S}$ entlang der Hauptachsen abweicht:

$$S = \bar{S} + P\vec{b} \tag{4.11}$$

Aufgrund der durch Gleichung (4.10) gewählten Normierung beschreiben die Koordinaten b_i $(i = 1 \ldots t)$ für jede Hauptachse die Abweichung in Einheiten der Standardabweichung.

Ohne eine Beschränkung der Formparameter auf einen geeigneten Wertebereich lassen sich so nahezu beliebige Formen erzeugen, also auch Konturen, die zur Beschreibung des gesuchten Objektes nicht relevant sind (vgl. auch Abschnitt 5.4). Der zulässige Wertebereich (engl. *allowable shape domain*) muss daher so eingeschränkt werden, dass durch Variation der Parameterwerte in diesem Bereich nur zulässige Konturen erzeugt werden.

Naheliegend ist die Wahl eines kasten- oder ellipsoidförmigen Wertebereiches. Bei einer kastenförmigen *shape domain* wird der Wertebereich für jede Komponente von $\vec{b}$ durch die Angabe einer oberen und unteren Grenze (in Vielfachen der Standardabweichung) angegeben:

$$\underline{b}_i \le b_i \le \overline{b}_i \tag{4.12}$$

Bei Wahl eines ellipsenförmigen Wertebereichs ergibt sich als Bedingung:

$$\sum_{i=1}^{t} \frac{b_i^{\,2}}{d_i^{\,2}} \le 1 \tag{4.13}$$

d_k bezeichnet hier den Halbmesser entlang der k-ten Eigenvektors.

4.4 Modellerstellung

4.4.1 Schritte der Modellerstellung

Für die Erstellung der PDMs wird eine hinreichend große Anzahl von repräsentativen Trainingskonturen benötigt. Zur Generierung von Modellen, die die Variationsmöglichkeiten der menschlichen Silhouette und des Gesichts möglichst naturgetreu wiederspiegeln, werden hierzu Bilder realer Szenen und Gesichter verwendet. Aus diesen sind zunächst die Konturen der Personensilhouetten bzw. der Gesichter zu extrahieren, die nach gegenseitiger Ausrichtung zueinander durch PCA untersucht werden, um so zu einer kompakten Modellbeschreibung zu gelangen.

Mit Hilfe der Modelle wird anschließend in neuen Bildfolgen nach Personen und Gesichtern gesucht, um die Position der Person bzw. deren Gesichtszüge zu verfolgen. Durch Variation der Parameterwerte b_i können neue Konturen erzeugt werden (Abschnitt 4.3.3). Diese können jedoch aufgrund der Einschränkung des zulässigen Variationsbereiches nicht beliebige Formen annehmen, sondern nur solche, die konsistent mit

dem aus den Trainingsdaten erstellten Modell sind. In neuen Bildern, die nicht zum Training benutzt wurden, können daher nur solche Konturen gefunden werden, die sich aus dem PDM erzeugen lassen. Bei der Auswahl der Bilder, die zum Trainieren des PDM benutzt werden, ist deshalb darauf zu achten, dass ein möglichst großer Bereich an Formvariationen abgedeckt wird. Auf eine Auswahl verschiedener Personen ist hier ebenso zu achten wie auf die Variation von Körperhaltung bzw. Mimik, die Aufnahme verschiedener Ansichten und die Position der Kamera.

Für jede Trainingskontur muss eine relativ große Anzahl *n* von Punkten definiert werden. Für die in den Abschnitten 4.4.6 und 4.4.7 vorgestellten Modelle für die menschliche Silhouette und das Gesicht ist beispielsweise *n=35* bzw. *n=134*. Bei der Auswertung mehrerer hundert Trainingsbilder wäre eine rein manuelle Definition jedes einzelnen Punktes sehr zeitaufwändig und fehleranfällig. Darüber hinaus ist es für die Definition der einzelnen Konturen meist ausreichend, einige wenige, charakteristische Punkte –die so genannten Landmarken– anzugeben und Zwischenpunkte äquidistant entlang des Kantenverlaufs im Bild zu platzieren. Die Extraktion der Konturen aus den Trainingsbildfolgen geschieht daher mit Hilfe eines interaktiven Tools, das den Experten unterstützt und eine halbautomatische Punktgenerierung ermöglicht (siehe Abschnitt 4.4.2).

Bild 4.3 zeigt in einer Übersicht die zur Modellerstellung notwendigen Schritte. Unter *Shapeset* wird hier eine Menge von mehreren Trainingskonturen verstanden.

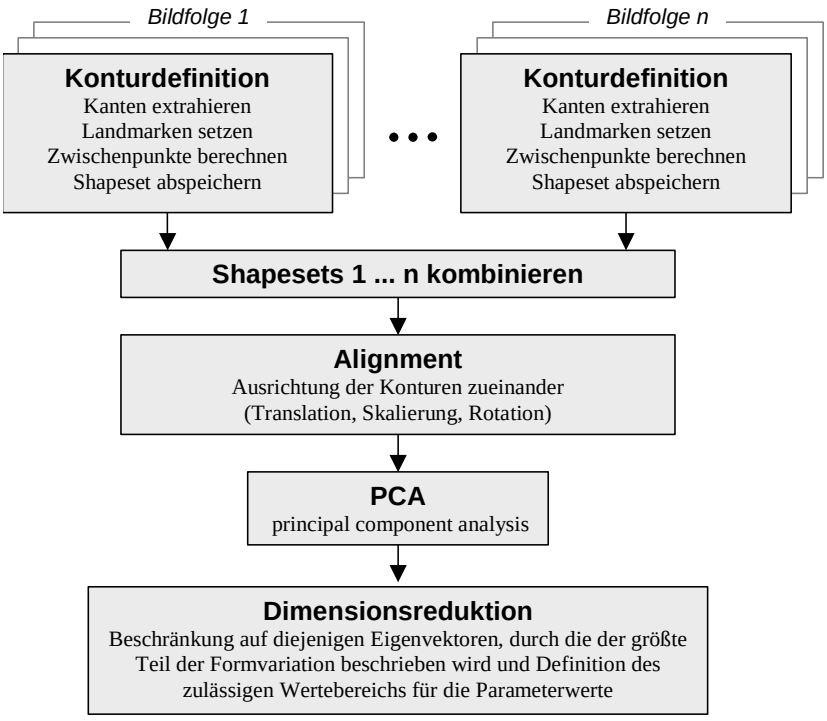


Bild 4.3: ***Schritte der Modellerstellung.***

Die aus den Trainingsbildern extrahierten Konturen werden anschließend durch Translation, Skalierung und Rotation so zueinander ausgerichtet, dass alle Variationen, die nicht auf eine *Form*änderung zurückzuführen sind, eliminiert werden (*shape alignment*, siehe Anhang B). Der so erhaltene Satz von normierten Konturen wird einer PCA unterworfen. Abschließend werden die Eigenvektoren mit den größten Eigenwerten ausgewählt, um so ein durch eine mittlere Form und mehrere Eigenvektoren beschriebenes, kompaktes Modell zu erhalten.

4.4.2 Intelligent Scissors

Bei der Extraktion der Trainingskonturen wird der Experte durch ein interaktives Tool unterstützt. Das Tool erlaubt es, zunächst den Kantenverlauf im Bild durch Angabe weniger Stützpunkte zu extrahieren und anschließend Landmarken auf der Kontur zu setzen. Die Berechnung äquidistanter Zwischenpunkte ist automatisiert. Im Vergleich zu einer rein manuellen Definition der Konturpunkte ist dieses Verfahren weitaus schneller, exakter und weniger fehleranfällig.

Die Kantenextraktion geschieht mit einem *intelligent scissors* genannten Verfahren, das von Mortensen und Barrett in [MB95][BM96] vorgestellt wurde. Ausgehend von einem mit der Maus markierten Startpunkt wird dabei durch dynamische Programmierung (DP) ein kostenoptimaler Pfad zu allen anderen Pixel berechnet. Bei Bewegung der Maus kann zu jedem Pixel der jeweils optimale Pfad angezeigt werden. Der Experte kann so interaktiv den besten Kantenverlauf aus allen optimalen Pfaden auswählen. Der Kantenverlauf im Bild wird hierbei zu jedem neuen Startpunkt erneut berechnet.

4.4.2.1 Lokale Kosten

Die Gesamtkosten $c_{total}(\cdot)$, um von einem Startpixel zu einem beliebigen anderen Bildpixel zu gelangen, sollten die Güte des Kantenverlaufs entlang dieses Pfades wiedergeben. Die lokale Kostenfunktion

$$c_{local}(p,q) = w_{mag}\,c_{mag}(q) + w_{dir}\,c_{dir}(p,q) + w_{ZC}\,c_{ZC}(q) \tag{4.14}$$

beschreibt die Kosten für den (gerichteten) Übergang vom Pixel p zum Nachbarpixel q und setzt sich als gewichtete Summe aus den in Tabelle 4.1 gezeigten Beiträgen zusammen, die im Weiteren detaillierter beschrieben werden.

Tabelle 4.1: ***Lokale Kostenfunktionen beim intelligent scissors Algorithmus.***

Kostenfunktion	*Gewichtung*	*Beschreibung*
c_{ZC}	w_{ZC}	Nulldurchgänge des Laplace-Operators
c_{mag}	w_{mag}	Gradientenstärke des Sobel-Operators
c_{dir}	w_{dir}	Gradientenrichtung des Sobel-Operators

Die Gewichtungen werden zu $w_{ZC} = 43\%$, $w_{mag} = 14\%$ und $w_{dir} = 43\%$ gewählt.

Nulldurchgänge des Laplace-Operators

Die Anwendung des Laplace-Operators

$$\nabla^2 img(x,y) = \frac{\partial^2}{\partial x^2} img(x,y) + \frac{\partial^2}{\partial y^2} img(x,y) \tag{4.15}$$

auf ein Bild *img(x,y)* wird digital implementiert als Faltung der Grauwerte mit der Maske

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.16}$$

Durch die Realisierung der 2. Ableitung des Grauwertverlaufes erzeugt der Filter scharfe Nulldurchgänge an Bildkanten. Da die Kantendetektion am besten auf unverrauschten Bildern mit starken Kanten anwendbar ist, wird das Originalbild zunächst mit einer Gaußmaske geglättet. Um die Nulldurchgänge im diskretisierten Bild zu bestimmen, wird ein Pixel immer dann als Kante akzeptiert, wenn der Grauwert im Laplacebild 0 ist oder mindestens eines der Nachbarpixel aus der 4-er Nachbarschaft ein anderes Vorzeichen hat. Durch Skelettierung ergibt sich schließlich eine Region mit Segmenten der Breite von einem Pixel.

Die aus dem resultierenden Bild img_{ZC} berechnete Kostenfunktion

$$c_{ZC}(q) = \begin{cases} 0 & , \quad img_{ZC}(q) = 0 \\ 1 & , \quad img_{ZC}(q) \neq 0 \end{cases} \tag{4.17}$$

ergibt also geringe Kosten für ein Pixel mit guten Kanteneigenschaften, was in diesem Fall einem Nulldurchgang des Laplace-Operators entspricht.

Amplitude des Sobel-Operators

Durch Faltung des Bildes mit den Masken

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \textit{bzw.} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4.18}$$

wird die 1. Ableitung in *x*- bzw. *y*-Richtung approximiert. Je stärker eine Bildkante ist, desto größer wird der Wert des Sobel-Filters sein. Aus dem Gradienten $\vec{G} = (G_x, G_y)$ berechnet sich die Kostenfunktion für die Gradientenstärke gemäß

$$c_{mag}(q) = 1 - \frac{|\vec{G}(q)|}{g_{mag}^{\max}} \tag{4.19}$$

Hierbei ist $g_{mag}^{\max}$ der maximale Grauwert des Kantenbildes, und es gilt in der Regel $g_{mag}^{\max} = 255$. Für starke Kanten, d.h. große Werte für $|\vec{G}|$, ergeben sich geringe Kosten, entsprechend hoch werden die Kosten bei schwacher Ausprägung der Kante.

Richtung des Sobel-Operators

Zusätzlich zu den Nulldurchgängen des Laplace-Operators und der Gradientenstärke wird mit dem Beitrag c_{dir} zur lokalen Kostenfunktion die Richtung des Grauwertgradienten $\vec{G}$ berücksichtigt:

$$c_{dir} = \frac{2}{3\pi}\left\{\arccos\left(\vec{D}(p)\cdot\vec{L}(p,q)\right) + \arccos\left(\vec{D}(q)\cdot\vec{L}(p,q)\right)\right\} \tag{4.20}$$

mit

$$\vec{L}(p,q) = \begin{cases} q-p & , \quad \vec{D}(p)\cdot(q-p) \geq 0 \\ p-q & , \quad \vec{D}(p)\cdot(q-p) \leq 0 \end{cases} \tag{4.21}$$

$\vec{D}$ ist hier der um 90° im mathematisch positiven Sinn gedrehte Einheitsvektor normal zum Gradientenvektor $\vec{G}$. Die gerichtete Kante $\vec{L}$ zwischen den Pixeln *p* und *q* wird so gewählt, dass $\vec{D}(p)$ und $\vec{L}$ einen spitzen Winkel bilden (siehe Bild 4.4).

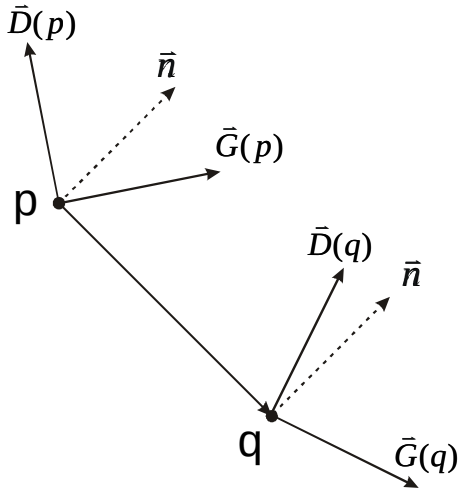


Bild 4.4: ***Berücksichtigung der Gradientenrichtung $\vec{G}$ bei der Berechnung der Kostenfunktion c_{dir}*** *beim Übergang von Pixel p zu Pixel q*

Die Winkel ergeben sich aus den entsprechenden Skalarprodukten. Es ist also $\arccos\{\vec{D}(p)\cdot\vec{L}(p,q)\} \in [0..\frac{\pi}{2}]$ und $\arccos\{\vec{D}(q)\cdot\vec{L}(p,q)\} \in [0..\pi]$.

Die Kostenfunktion c_{dir} ist umso kleiner, je besser die Gradientenrichtung mit der Normalen $\vec{n}$ der Verbindung zwischen den beiden Pixeln übereinstimmt.

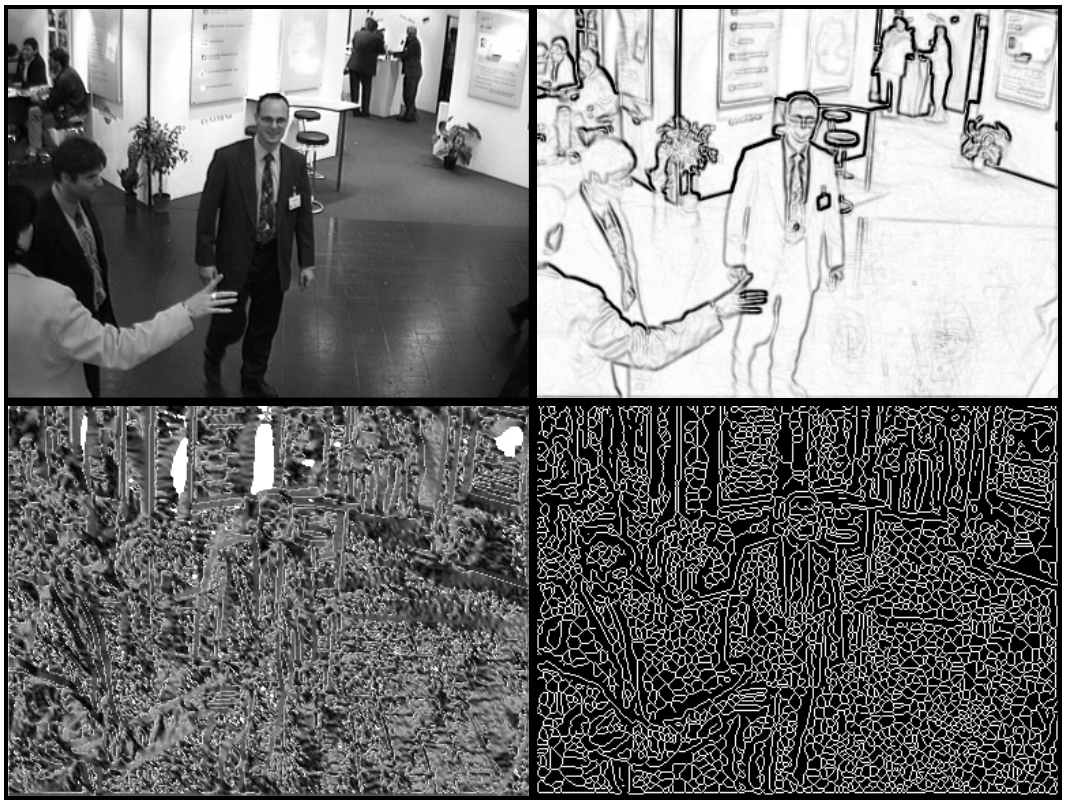


Bild 4.5: ***Bilder für die Berechnung der Kostenfunktion $c_{local}(p,q)$****. Oben links: Originalbild. Oben rechts: Amplitude des Sobelfilters (c_{mag}). Unten links: Richtungskosten beim Übergang zum Pixel links oben (c_{dir}). Unten rechts: Nulldurchgänge des Laplace-Operators (c_{ZC}).*

4.4.2.2 Algorithmus

Die Berechnung des kostenoptimalen Pfades von einem Startpixel (engl. *start point*) zu einem beliebigen Bildpixel kann als Suche in einem gerichteten Graphen aufgefasst werden. Die Knoten des Graphen werden von den Bildpixeln gebildet, die Kosten für den Übergang zwischen zwei Knoten beschreibt die Funktion $c_{local}(p,q)$ aus Gleichung (4.14). Mit dem auf Seite 65 dargestellten Algorithmus wird durch dynamische Programmierung (DP) der optimale Pfad zu *allen* anderen Bildpixeln ermittelt. Ausgehend vom Startpixel werden dazu zwei Matrizen aufgebaut, welche die kumulativen Kosten $c_{total}(\cdot)$ enthalten sowie einen Richtungsvektor, der beschreibt, welches der 8 Nachbarpixel den Pfad mit den geringsten Kosten bildet. In jedem Schritt werden die Nachbarpixel desjenigen Pixels in eine so genannte aktive Liste L aufgenommen, das die geringsten kumulativen Kosten hat (Expansion). Wenn kein Nachbar mit niedrigeren Gesamtkosten gefunden wird, fällt das Pixel aus der aktiven Liste heraus. Die expandierten Punkte breiten sich ausgehend vom Startpixel aus, wobei die Wellenfronten entlang von Bildkanten schneller propagieren als in Richtungen mit hohen Übergangskosten (siehe Bild 4.6).

Die Berechnung der global optimalen Pfade erfordert auf einem Pentium III Prozessor mit einer Taktfrequenz von 800 MHz bei Bildern der Größe 768*576 Pixel nur Bruchteile einer Sekunden. Bei Bewegung des Mauszeigers kann daher unmittelbar nach der Definition des Startpunktes zu jedem Pixel der optimale Pfad angezeigt werden. Damit wird bei der Modellerstellung eine interaktive Auswahl des Kantenverlaufes möglich.

Algorithmus *Intelligent Scissors* [MB95]

```
// variables
seed_point          // start point, defined by mouse click
L                   // list of active pixels, sorted by total cost
N(q)                // 8 neighbours of pixel q
e(q)                // boolean matrix indicating if q has been expanded
predecessors(q)     // matrix of pointers describing minimum cost path
c_total(q)          // total cost from seed point to q

// initialization
c_total(seed_point) ← 0:                    // total cost = 0 for start point
L ← seed_point:                             // add start point to active list

// algorithm
while (L ≠ ∅)  // still points to be processed
{
    q_min ← min(L):                         // get minimum cost pixel from active list L
    delete_min(L):                          // remove minimum cost pixel from L
    e(q_min) ← TRUE:                        // mark pixel as expanded

    for all n∈N(q_min) with (e(q_min)==FALSE) do
    {
       c_tmp ← c(q_min)+c_local(q_min,n):   // total cost to neighbour
       if (n∈L)    // if n is active already
       {
         if (c_tmp < c_total(n))
         {
             // new path is cheaper than old one
             c_total(n) ← c_tmp:            // store new total cost for n
             predecessors(n) ← q_min:       // store back pointer
             decrease_key(L,n,c_tmp):       // decrease costs for n
         }
       }
       else
       {
         c_total(n) ← c_tmp:                // store total cost for n
         predecessors(n) ← q_min:           // store back pointer
         insert(L,n):                       // add n to active list
       }
    }
}
```

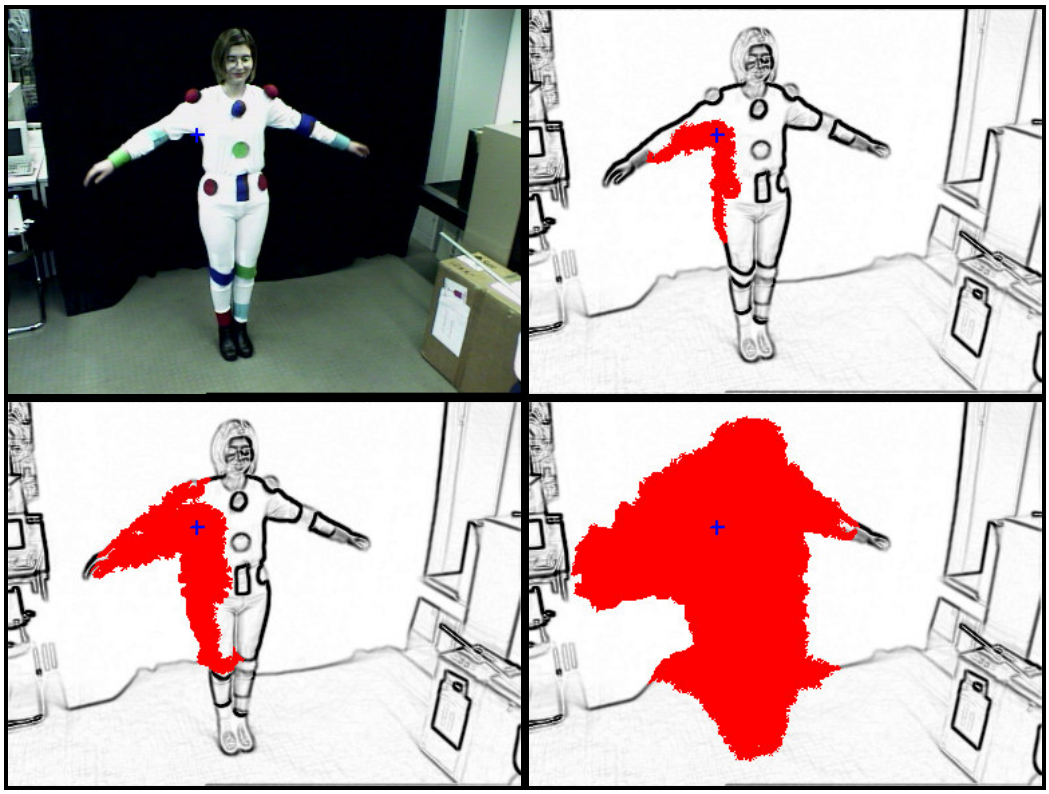


Bild 4.6: ***Ausbreitung der Wellenfronten der expandierten Pixel*** *(rot). Ausgehend von einem Startpixel (blau markiert) verläuft die Ausbreitung bevorzugt entlang eines Pfades mit geringen Kosten, d.h. entlang von Bildkanten. Originalbild (oben links), expandierte Pixel nach 2000 (oben rechts), 6000 (unten links) und 28000 (unten rechts) Expansionen.*

4.4.3 Auswahl der Trainingsbilder

Für die Erstellung der Punktverteilungsmodelle muss eine hinreichend große Anzahl von Trainingsbildern ausgewertet werden, aus denen möglichst viele Trainingskonturen extrahiert werden. Der Experte wird hierbei von einem interaktiven Tool unterstützt, das eine halbautomatische Definition der Landmarken erlaubt und das für die Kantenextraktion den in Abschnitt 4.4.2 beschriebenen *intelligent scissors*-Algorithmus verwendet.

Bei der Auswahl der Trainingsbilder ist darauf zu achten, dass die Formen der extrahierten Konturen möglichst den kompletten Variationsbereich abdecken, der auch später in neuen, nicht im Trainingsdatensatz vorhandenen Bildern wiedergefunden werden soll. Mit dem erstellten PDM kann zwar zwischen einzelnen Konturen interpoliert werden, völlig neue Formen können hingegen nicht erzeugt werden.

In den folgenden Abschnitten werden zwei Modelle vorgestellt: ein Modell für die menschliche Silhouette, das die Kopf-Schulter Partie bis hin zu den Ellenbogen umfasst

sowie ein komplexes, aus mehreren Teilen aufgebautes Modell des menschlichen Gesichtes.

Bei der Erstellung des Modells zur Beschreibung der für Personen typischen Kopf-Schulter Partie wurden Bilder verschiedener Individuen ausgewertet, auf denen die betreffenden Personen aus unterschiedlichen Blickrichtungen zu sehen sind. Verschiedene Frontal- und Seitenansichten wurden dabei ebenso berücksichtigt wie leicht gebeugte und zur Seite geneigte Haltungen.

Für das in Abschnitt 4.4.7 vorgestellte Gesichtsmodell wurden zum einen Bilder ausgewertet, auf denen die Kopfhaltung der aufgenommenen Personen variiert. Um auch die Bewegung der Gesichtsteile relativ zueinander zu modellieren, fließen in das Modell darüber hinaus Trainingsdaten mit Aufnahmen unterschiedlicher Mimiken ein.

4.4.4 Definition von Landmarken und Zwischenpunkten

Die Auswertung der Trainingsbilder erfolgt in zwei Schritten. Zunächst werden mit Hilfe eines interaktiven Tools halbautomatisch die Objektkanten extrahiert. Hierbei wird mit dem *intelligent scissors* Algorithmus ausgehend von einem mit der Maus markierten Startpixel der optimale Pfad zu allen anderen Bildpixeln ermittelt. Bei anschließender Bewegung der Maus wird zu dem jeweils überfahrenen Pixel der kostengünstigste Pfad angezeigt, so dass dieser ausgewählt werden kann, wenn er hinreichend gut mit den Objektkanten übereinstimmt.

Nachdem der das Objekt beschreibende Kantenverlauf segmentiert wurde, werden in einem zweiten Schritt die Landmarken und Zwischenpunkte gesetzt. Bei der Definition der Landmarken ist darauf zu achten, dass deren Positionierung in allen Bildern konsistent erfolgt und dass die Punkte immer den gleichen charakteristischen Teil der Kontur beschreiben (z.B. oberster Kopfpunkt, linke Schulter, linker Mundwinkel, etc.). Zwischen diesen fixen Landmarken wird eine hinreichend große Anzahl von Zwischenpunkten automatisch äquidistant entlang des Konturverlaufs interpoliert.

Connection Types

Damit einteilige von mehrteiligen Konturen unterschieden werden können, wird jedem Konturpunkt ein so genannter Verbindungstyp (*connection type*) zugeordnet, der beschreibt, ob es sich um einen Endpunkt für ein Konturteil handelt und ob das Konturteil geschlossen bzw. offen ist. Der Typ der Verbindung kann drei verschiedene Werte annehmen (siehe Tabelle 4.2). Ein Punkt mit *connection type* 0 kennzeichnet den letzen Punkt eines offenen Konturteils, *connection type* 1 stellt einen Punkt mit Verbindung zu einem weiteren Punkt dar, Typ 2 kennzeichnet den letzten Punkt eines geschlossenen Konturteils. Der letzte Punkt eines geschlossenen Konturteils wird durch eine Gerade mit dem ersten Punkt des Konturteils verbunden.

Tabelle 4.2: ***Verbindungstypen (connection types) zwischen den einzelnen Modellpunkten*** *zur Definition von ein- und mehrteiligen Konturen.*

connection type	Bedeutung
0	beende Konturteil
1	Verbindung zum nächsten Punkt, falls nicht letzter Punkt
2	beende und schließe Konturteil

Der Typ der Verbindung bestimmt zum einen die visuelle Darstellung der Kontur, zum anderen wird daraus bei der PDM-Suche die Orthogonale zur Kontur ermittelt, entlang der die Kantensuche erfolgt (siehe Abschnitt 4.5.7).

Eine Kontur mit *connection types* 1101112 beispielsweise besteht aus insgesamt sieben Punkten, die einen offenen Konturteil (definiert durch die drei *connection types* 110) sowie einen aus vier Punkten bestehenden geschlossenen Teil (definiert durch 1112) bilden. Für einen inneren Konturpunkt x_i, d.h. einen Punkt, der nicht den Beginn oder Abschluss eines offenen Konturteils bildet, wird die Senkrechte zum Konturverlauf aus der Winkelhalbierenden der Strecken $\overline{x_i x_{i-1}}$ und $\overline{x_i x_{i+1}}$ ermittelt. Bei einem Anfangs- oder Endpunkt genügt die Betrachtung der Verbindung zum Nachfolger/Vorgänger.

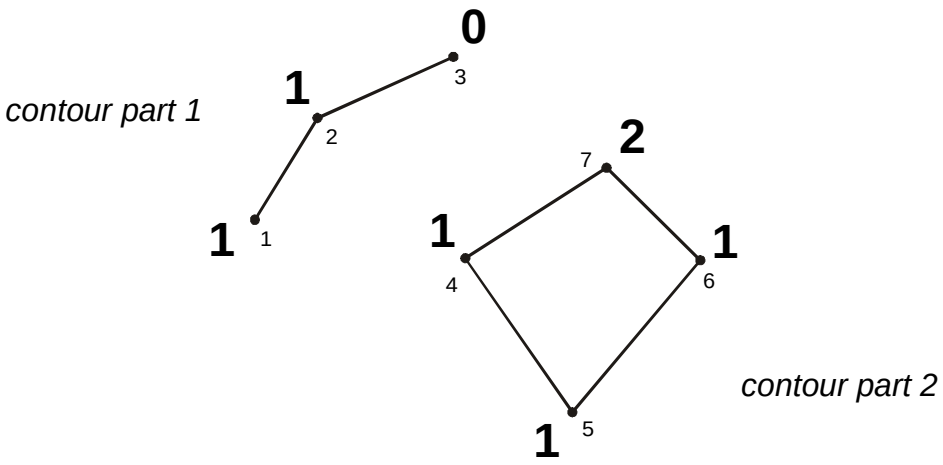


Bild 4.7: ***Zweiteilige Kontur****. Durch die Zuweisung der connection types 1101112 (große Indizes) zu den einzelnen Konturpunkten (kleine Indizes) wird ein offener und ein geschlossener Konturteil definiert.*

4.4.5 Dimensionsreduktion

Nach der Definition von fixen Landmarken und Zwischenpunkten für eine hinreichend große Anzahl an Trainingskonturen werden diese durch Skalierung, Translation und

Rotation zueinander ausgerichtet. Die Ausrichtung von Konturen ist in Anhang B beschrieben. Auf den so normierten Koordinaten des Trainingsdatensatzes wird dann nach dem in Abschnitt 4.3.2 beschriebenen Verfahren eine Hauptachsentransformation (*PCA*) durchgeführt. Diese liefert eine mittlere Modellkontur $\overline{S}$ und $2n$ Eigenvektoren und Eigenwerte, mit denen gemäß Gleichung (4.11) neue Konturen erzeugt werden können.

Zur Beschreibung der Variation der Trainingsdaten genügt eine kleine Anzahl $t \ll 2n$ an Eigenvektoren. Um durch Linearkombination der Eigenvektoren nur zulässige Formen zu erhalten, muss der Variationsbereich für die Parameterwerte b_i eingeschränkt werden (siehe Abschnitt 4.3.3). Die obere und untere Schranke für die zulässige Variation kann global für sämtliche verwendeten Eigenvektoren gesetzt werden. Die Wahl individueller Schranken $\overline{b}_i$ und $\underline{b}_i$ kann in der implementierten Klassenstruktur aber auch für jeden Eigenvektor getrennt erfolgen. Dies ist insbesondere bei dem mehrteiligen Gesichtsmodell, das in Abschnitt 4.4.7 vorgestellt wird, sinnvoll, da so charakteristische Bewegungen einzelner Gesichtsteile oder Gesichtspartien gezielt kontrolliert werden können.

4.4.6 Modell der menschlichen Silhouette

Für die Erstellung eines Modells der menschlichen Silhouette wurden 10 Videobildfolgen ausgewertet, in denen verschiedene Personen in unterschiedlichen Ansichten zu sehen sind. Insgesamt wurden 202 Trainingskonturen extrahiert, die wegen der Symmetrie des menschlichen Oberkörpers zusätzlich an der Vertikalen gespiegelt wurden. Zum Trainieren des Modells standen somit insgesamt 404 Trainingssilhouetten zur Verfügung. Bild 4.8 zeigt einige der verwendeten Trainingsbilder mit eingezeichneter Silhouette.

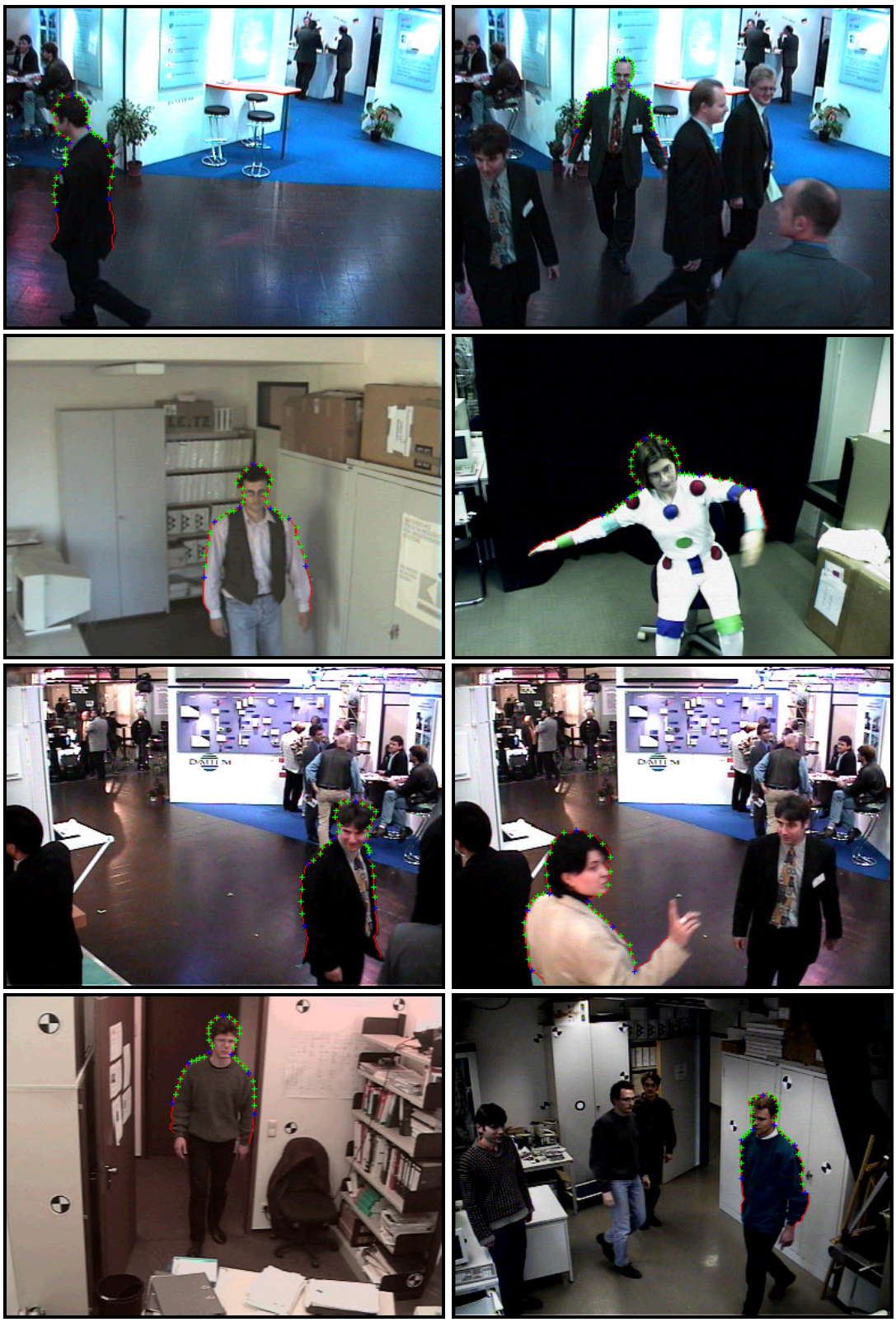


Bild 4.8: ***Trainingsbilder für das Modell der menschlichen Silhouette****. Blau eingezeichnet sind die mit der Maus definierten fixen Landmarken, zwischen denen die grün markierten Punkte automatisch äquidistant entlang des Kantenverlaufs platziert werden.*

Als fixe Landmarken, die in jedem Bild eindeutig markiert werden müssen, wurden die sieben in Bild 4.9 rot markierten Punkte verwendet (linker Ellenbogen, linke Schulter, linker Halspunkt, Scheitel, rechter Halspunkt, rechte Schulter, rechter Ellenbogen). Auf jeder Seite werden zwischen Ellenbogen und Schulter sowie zwischen Schulter und Halspunkt jeweils 4 äquidistante Zwischenpunkte entlang des Kantenverlaufs gesetzt (in Bild 4.9 grau markiert), zwischen Halspunkt und Scheitel beträgt die Anzahl der Zwischenpunkte jeweils 6. Eine Silhouette wird aus insgesamt 35 Punkten gebildet. Jeder Konturpunkt hat den *connection type* 1, womit die Kontur aus nur einem einzigen offenen Konturteil besteht.

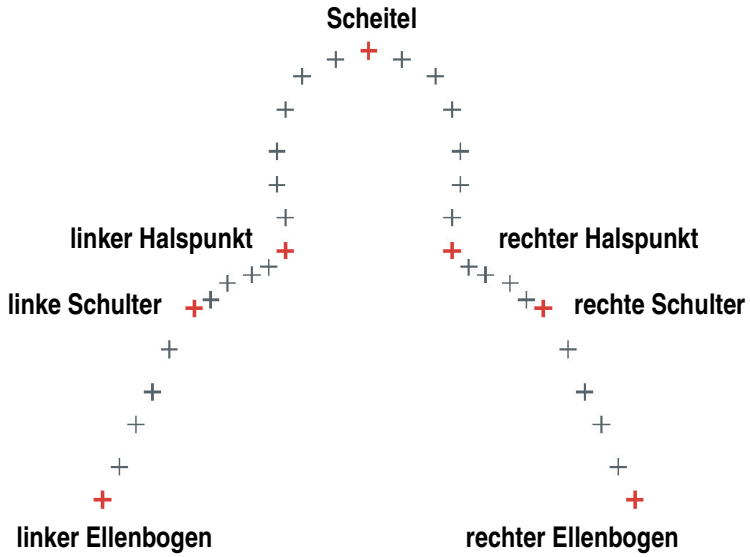


Bild 4.9: ***Modell der menschlichen Silhouette****. Gezeigt ist die mittlere Silhouette des erzeugten PDM. Die rot eingezeichneten Punkte markieren die fixen Landmarken, d.h. die Punkte, die in jedem Trainingsbild korrespondieren müssen und die mit der Maus markiert werden. Die grauen Zwischenpunkte werden automatisch äquidistant platziert.*

Da die Silhouetten aus jeweils 35 Punkten bestehen, werden bei der Hauptachsentransformation 70 Eigenvektoren ermittelt, die nach der Größe der zugehörigen Eigenwerte geordnet werden. Die Eigenwerte beschreiben die Varianz des Trainingssatzes entlang der Hauptachsen. Eine Übersicht über die absolute Größe der Einzelvarianzen sowie den prozentualen Anteil an der Gesamtvarianz der ersten 10 Eigenvektoren gibt Tabelle 4.3.

Tabelle 4.3: ***Eigenwerte/Varianzen für das Modell der menschlichen Silhouette****. (abs. = absoluter Wert; % = prozentualer Anteil an der Gesamtvarianz)*

Eigenvektor	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Eigenwert abs.	787.7	188.1	113.5	95.1	62.5	23.1	14.7	11.0	8.8	6.7
Eigenwert %	58.8	14.0	8.5	7.1	4.7	1.7	1.1	0.8	0.7	0.5
Σ%	58.8	72.8	81.3	88.4	93.1	94.8	95.9	96.7	97.4	97.9

Bild 4.10 verdeutlicht, dass der prozentuale Anteil der einzelnen Eigenwerte an der Gesamtvarianz rasch abnimmt. Die Punktwolke, die sich aus den Trainingsdaten im $\mathbb{R}^{2n} = \mathbb{R}^{70}$ ergibt, lässt sich also durch eine geringe Anzahl ($t \ll n$) Eigenvektoren beschreiben. Die Ausdehnung in Richtung der übrigen Achsen kann vernachlässigt werden. Im gezeigten Beispiel kann z.B. allein durch Variation des ersten Eigenvektors über 50% der Formvariation beschrieben werden.

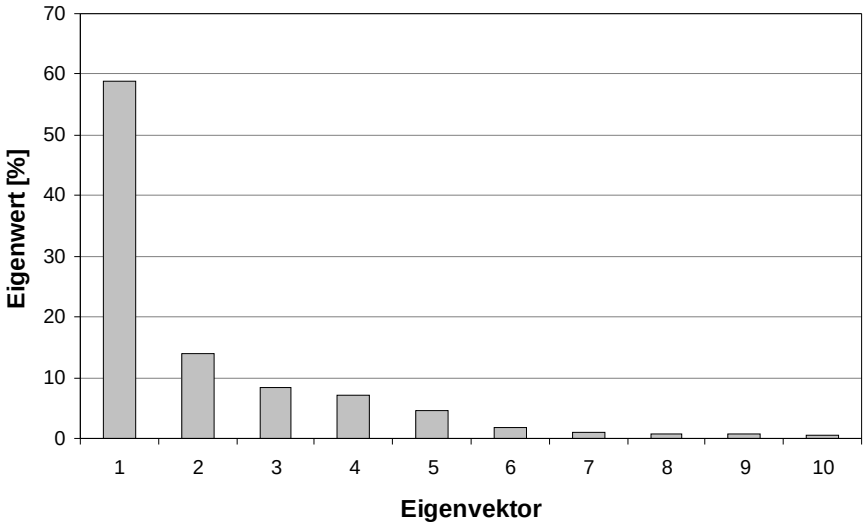


Bild 4.10: ***Prozentualer Anteil der Eigenwerte für das Modell der menschlichen Silhouette.*** *Die ersten 10 Eigenvektoren modellieren 97,9% der Gesamtvariation λ_T.*

Die Gesamtvarianz für das vorgestellte Modell beträgt

$$\lambda_T = \sum_{k=1}^{70} \lambda_k = 1339.3 \tag{4.22}$$

Bei der Berücksichtigung der ersten 6 ($t=6$) Eigenvektoren beträgt die Variation, die durch diese Vektoren beschrieben wird

$$\lambda_6 = \sum_{k=1}^{6} \lambda_k = 1270.0 \tag{4.23}$$

so dass der Anteil an der Gesamtvarianz 95 % beträgt.

Der Effekt, den die einzelnen Eigenvektoren auf die Form der Silhouette haben, ist in Bild 4.11 veranschaulicht. Dargestellt ist jeweils die mittlere Silhouette $\bar{S}$ des Modells (rot eingezeichnet) mit einem Parametervektor $\vec{b} = (0 \ldots 0)$ sowie die Silhouetten, die sich ergeben, wenn die Parameterwerte in einem Bereich von b_i=*–3.0...+3.0* Standardabweichungen variiert werden.

Die gezeigten Silhouetten verdeutlichen Folgendes:

- Die Eigenvektoren mit den größten Eigenwerten rufen die größten Formvariationen hervor. Mit abnehmender Größe der Eigenwerte nimmt auch der Einfluss auf die Form ab, so dass es ausreicht, sich bei der Beschreibung der Formänderungen auf eine kleine Anzahl zu beschränken.
- Jeder Eigenvektor beschreibt eine charakteristische Bewegung bzw. Formänderung der menschlichen Silhouette. So beschreibt z.B. Eigenvektor 1 das Heben und Senken der Arme, Eigenvektor 2 die Kopfneigung bzw. den Übergang in die Seitenansicht und Eigenvektor 5 die Kopfgröße relativ zur Größe des Rumpfes.
- Sinnvolle Formen lassen sich nur dann erzeugen, wenn der zulässige Wert der Parameterwerte b_i beschränkt wird. Bei zu großer Variation kommt es zu entarteten Formen. So können z.B. mittels des Eigenvektors 1 die Arme zu weit nach innen gebeugt werden. Entartete Formen werden insbesondere auch dann erzeugt, wenn mehrere Parameterwerte gleichzeitig in einem großen Bereich variiert werden (vergleiche hierzu auch Abschnitt 5.4).

Für die Anwendung des Modells zur Personendetektion werden die ersten 6 Eigenvektoren berücksichtigt, durch die *95%* der Gesamtvariation beschrieben werden. Der zulässige Variationsbereich (engl. *allowable shape domain*, siehe auch Abschnitt 4.3.3) wird je nach Anwendung auf etwa –2.5 ... +2.5 Standardabweichungen beschränkt.

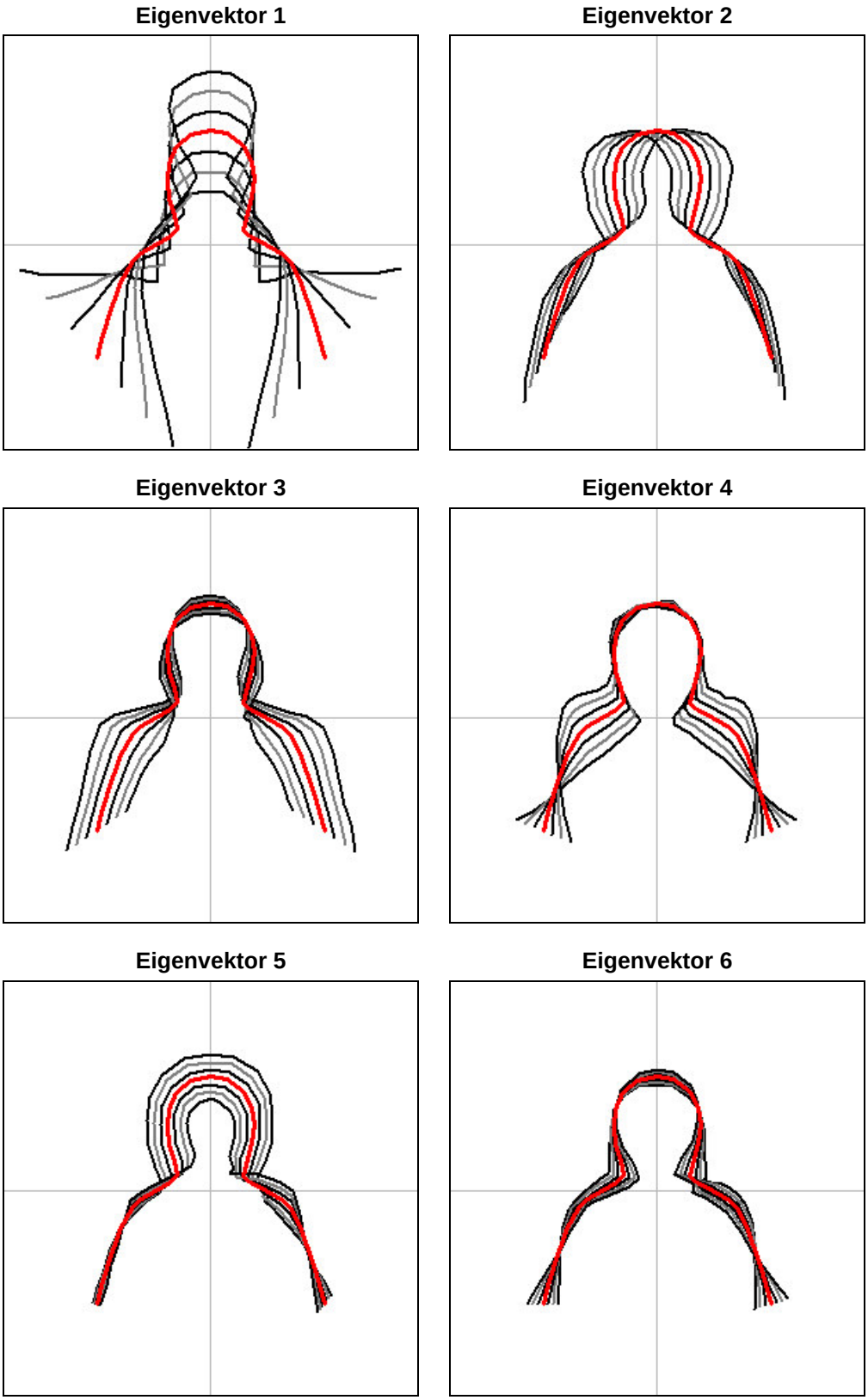


Bild 4.11: ***Veränderung der Silhouettenform*** *bei Variation der ersten 6 Parameterwerte im Bereich –3.0 bis +3.0 Standardabweichungen. Jeder Eigenvektor beschreibt charakteristische Formvariationen der mittleren Silhouette (rot).*

4.4.7 Gesichtsmodell

Im Bereich der Sicherheitstechnik ist es häufig wünschenswert, von verdächtigen Personen formatfüllende Portraitaufnahmen zu machen oder Personen anhand der aufgenommenen Bilder automatisch zu identifizieren. Mit dem im vorigen Abschnitt vorgestellten Modell für die menschliche Silhouette ist es möglich, die Position des Kopfes im Raum zu bestimmen. Für eine exakte Bestimmung der Lage des Gesichts und im Hinblick auf eine zukünftige automatisierte Identifikation einer Person im Rahmen von STABIL++ ist es jedoch notwendig, die Gesichtszüge genauer zu bestimmen. Deshalb wurde im Rahmen dieser Arbeit ein Modell für das menschliche Gesicht erstellt.

Das Aussehen eines Gesichtes im Bild wird von vielen verschiedenen Faktoren bestimmt. Ein Modell muss zum einen die Größen- und Formunterschiede im Aussehen verschiedener Personen berücksichtigen. Weitere Variationsmöglichkeiten ergeben sich darüber hinaus durch Drehen, Heben und Senken des Kopfes sowie durch Formveränderung einzelner Gesichtsteile und Variation der relativen Positionen der Teile zueinander.

Im Rahmen dieser Arbeit wurde eine Datenbank mit über 1000 Bildern von verschiedenen Personen erstellt. Im Hinblick auf eine zukünftige Anwendung des Modells zur automatisierten Mimikerkennung wurden neben dem neutralen Gesichtsausdruck auch verschiedene weitere Gemütsausdrücke der Testpersonen aufgenommen. Im Einzelnen wurden folgende Variationen berücksichtigt (siehe Bild 4.12):

- *verschiedene Personen*

  Es wurden Aufnahmen von 16 verschiedenen Personen im Alter von 23-64 Jahren gemacht. Die Anzahl der männlichen und weiblichen Testpersonen war in etwa gleich groß.

- *Kopfhaltung*

  Drehen sowie Heben und Senken des Kopfes nach rechts, links, oben, unten, rechts oben, links oben, rechts unten und links unten

- *Mimiken*

  Um einen Klassifikator für verschiedene Mimiken trainieren zu können, wurden Aufnahmen mit neutralem, freudigem, überraschtem sowie zornigem/wütenden Gesichtsausdruck gemacht. Hierdurch wird auch die Formvariation einzelner Gesichtsteile sowie deren relative Lageveränderung teilweise berücksichtigt.

- *Bewegung einzelner Gesichtsteile*

  Augen auf/zu, Pupillen nach links/rechts und oben/unten bewegen, Mund öffnen/schließen, verschiedene Mundformen beim Sprechen der Vokale a-e-o-u, grinsen, Kussmund, Augenbrauen nach oben und unten bewegen

Für die Modellerstellung wurden 212 repräsentative Bilder ausgewählt, aus denen die Trainingskonturen extrahiert wurden. Bei der Auswahl wurde darauf geachtet, dass sämtliche Personen und Mimiken im Trainingsdatensatz vorhanden sind. Es wurden etwa gleich viele Bilder mit gehobenem wie gesenktem Blick verwendet, damit die mittlere PDM-Kontur keine Neigung nach oben oder unten aufweist. Da das menschliche Gesicht im Wesentlichen symmetrisch zur Vertikalen ist, wurden die erzeugten Konturen zusätzlich an dieser gespiegelt, so dass insgesamt 424 Trainingskonturen für die Erzeugung des Modells zur Verfügung standen. Durch die Spiegelung wurde auch erreicht, dass in das Modell gleich viele Bilder mit Blick nach links wie nach rechts einfließen.

Das Modell (siehe Bild 4.13) ist aus 10 Konturteilen aufgebaut: Augenbraue links, Auge links, Pupille links, Augenbraue rechts, Auge rechts, Pupille rechts, Nase, Oberlippe, Unterlippe und Kinn. Durch die Zuweisung entsprechender *connection types* werden Nase und Kinn als offen, die restlichen Konturteile als geschlossen definiert. Die Bezeichnung ‚links' und ‚rechts' bezieht sich auf die Position im Bild. Auf eine Modellierung der Ohren wurde verzichtet, da diese schon bei leichter Drehung des Kopfes nicht mehr sichtbar sind. Ebenso wurde auf die Stirnpartie mit Haaren verzichtet, da diese sehr starken Veränderungen unterliegt.

Eine Modellkontur besteht aus 30 fixen Landmarken, die in jedem Trainingsbild eindeutig markiert werden müssen, sowie einer Reihe von äquidistant platzierten Zwischenpunkten, deren Anzahl von der Größe des zugehörigen Modellteils abhängt. Eine Übersicht über die für die einzelnen Modellteile verwendete Anzahl von Punkten gibt Tabelle 4.4.

Tabelle 4.4: ***Spezifikation der Landmarken und Zwischenpunkte für das Gesichtsmodell.***

Konturteil	*Landmarken*	*Zwischenpunkte*	*Gesamtzahl*
Augenbraue links	2	11	13
Auge links	2	8	10
Pupille links	4	4	8
Augenbraue rechts	2	11	13
Auge rechts	2	8	10
Pupille rechts	4	4	8
Nase	5	10	15
Oberlippe	3	14	17
Unterlippe	3	14	17
Kinn	3	20	23
Σ	**30**	**104**	**134**

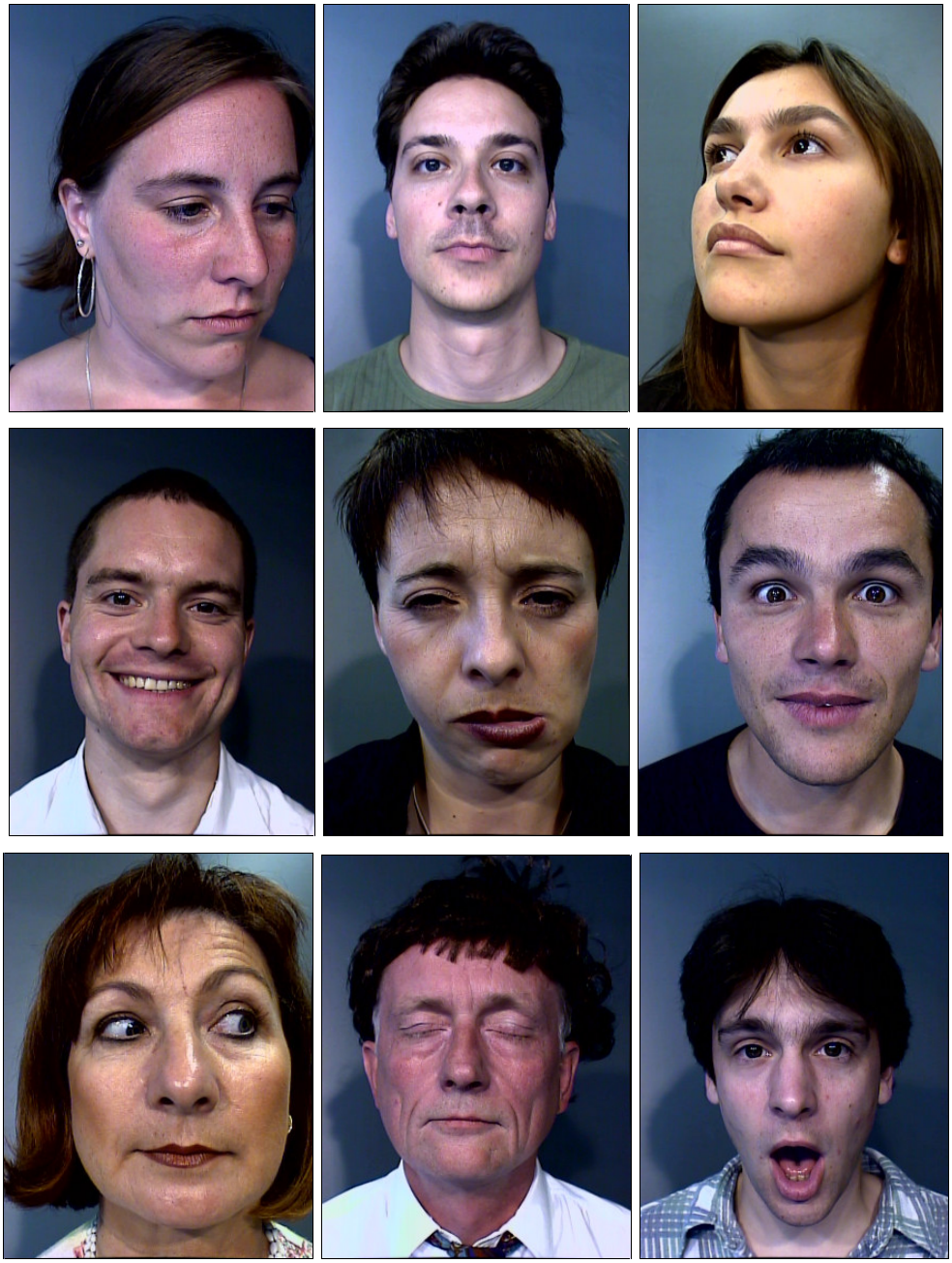


Bild 4.12: ***Trainingsbilder für das Gesichtsmodell****. Oben: verschiedene Blickrichtungen. Mitte: unterschiedliche Mimiken (Freude, Wut, Überraschung). Unten: Bewegung der Gesichtsteile relativ zueinander (Pupillenbewegung, Augen schließen/öffnen, Mund auf/zu).*

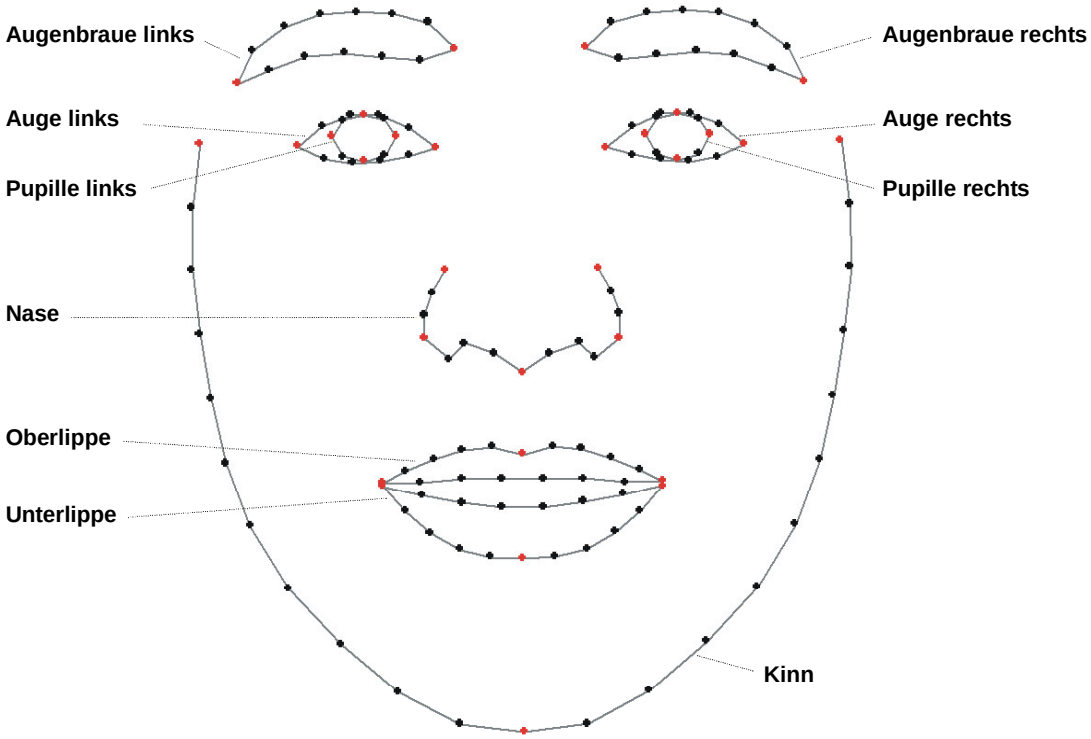


Bild 4.13: ***Gesichtsmodell mit Bezeichnung der einzelnen Konturteile****. Gezeigt ist die mittlere Gesichtsform des erstellten Modells. Rot markiert sind die in jedem Trainingsbild eindeutig zu markierenden Landmarken, die schwarzen Punkte werden äquidistant platziert.*

Die insgesamt 134 Modellpunkte führen bei der Hauptachsentransformation zur Berechnung von 268 Eigenvektoren und Eigenwerten. Eine Übersicht über die absolute Größe und den prozentualen Anteil an der Gesamtvarianz der ersten 17 Eigenwerte geben Tabelle 4.5 sowie Bild 4.14.

Tabelle 4.5: ***Eigenwerte/Varianzen für das Gesichtsmodell*** *(abs. = absoluter Wert; % = prozentualer Anteil an der Gesamtvarianz)*

Eigenvektor	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Eigenwert abs.	20467	9257	5668	2999	2460	1743	1636	1256	810
Eigenwert %	39.3	17.8	10.9	5.8	4.7	3.4	3.1	2.4	1.6
Σ %	39.3	57.1	68.0	73.8	78.5	81.9	85.0	87.4	89.0

Eigenvektor	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*
Eigenwert abs.	592	529	456	453	379	269	224	221
Eigenwert %	1.1	1.0	0.9	0.9	0.7	0.5	0.4	0.4
Σ %	90.1	91.1	92.0	92.9	93.6	94.1	94.5	94.9

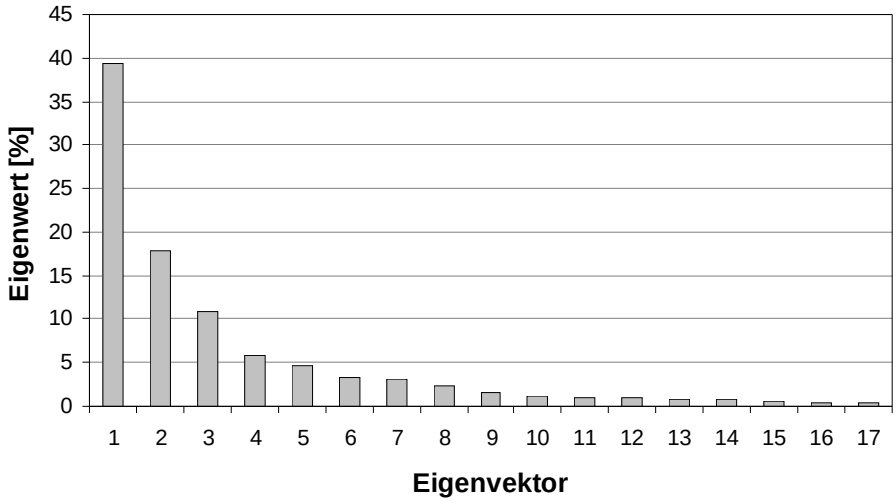


Bild 4.14: ***Prozentualer Anteil der Varianzen für das Gesichtsmodell.*** *Die ersten 17 Eigenvektoren modellieren 95,0% der Gesamtvariation λ_T.*

Die Größe der Eigenwerte nimmt auch hier schnell ab, so dass sich der Trainingsdatensatz im $\mathbb{R}^{2n} = \mathbb{R}^{268}$ wiederum durch eine kleine Anzahl an Eigenvektoren approximieren lässt. Die Gesamtvarianz für das Gesichtsmodell beträgt

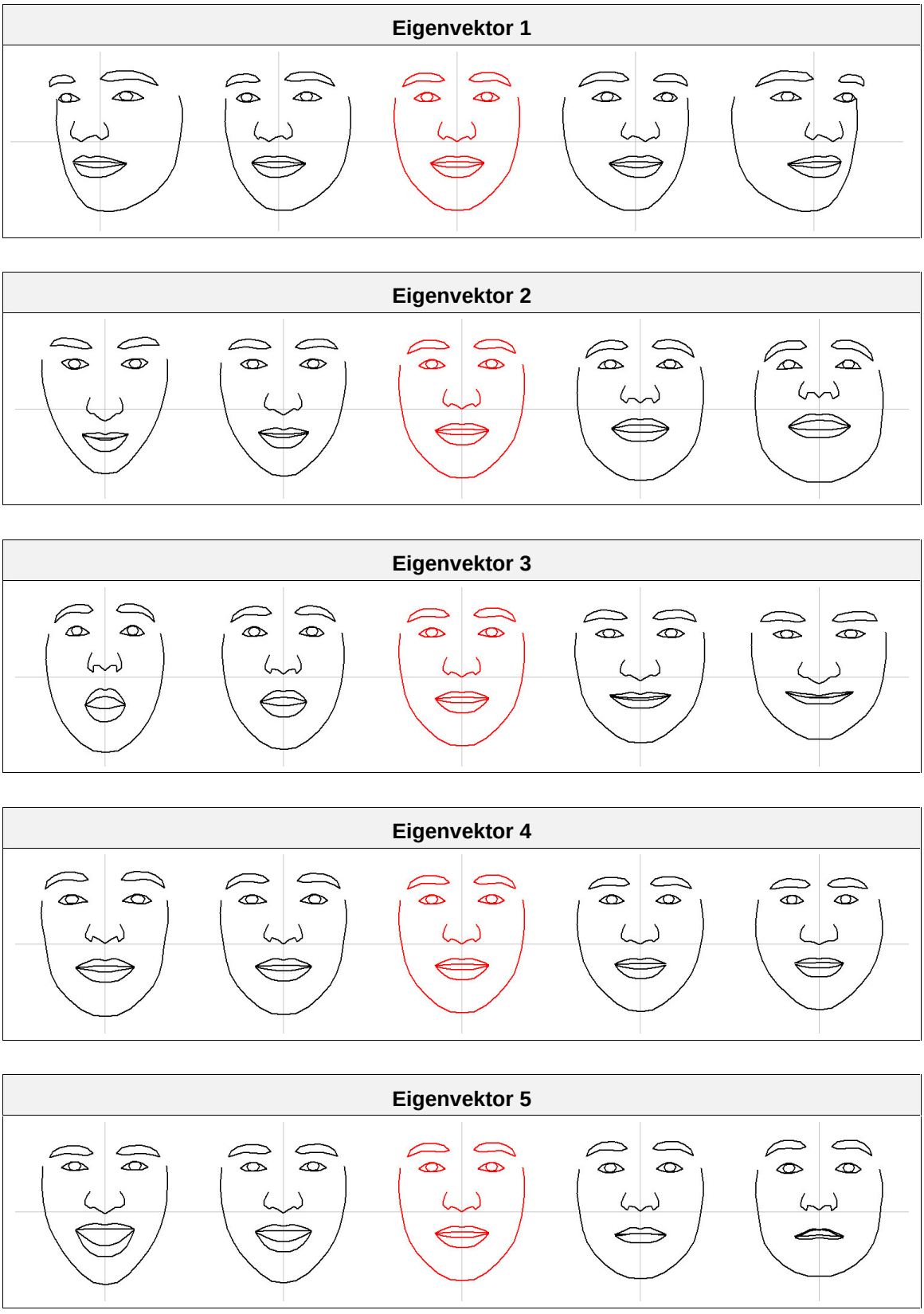
$$\lambda_T = \sum_{k=1}^{268} \lambda_k = 52029 \tag{4.24}$$

Bei der Berücksichtigung der ersten 17 ($t=17$) Eigenvektoren beträgt die Variation, die durch diese Vektoren beschrieben wird

$$\lambda_{17} = \sum_{k=1}^{17} \lambda_k = 49419 \tag{4.25}$$

Der Anteil an der Gesamtvarianz beträgt dann 95 %, so dass es wie beim Silhouettenmodell für den Oberkörper genügt, sich bei der Suche auf diese vergleichsweise kleine Anzahl von Eigenvektoren zu beschränken.

Den Effekt, den die einzelnen Eigenvektoren auf die Gesichtsform haben, zeigt Bild 4.15. Aus Gründen der Übersichtlichkeit ist die Variation nur für die ersten 10 Eigenvektoren und Parameterwerte in einem Bereich von $b_i = -2.0 \ldots +2.0$ Standardabweichungen gezeigt. Die entsprechenden Bilder für die Eigenvektoren 11-17 befinden sich in Anhang C.


Eigenvektor 1
Eigenvektor 2
Eigenvektor 3
Eigenvektor 4
Eigenvektor 5

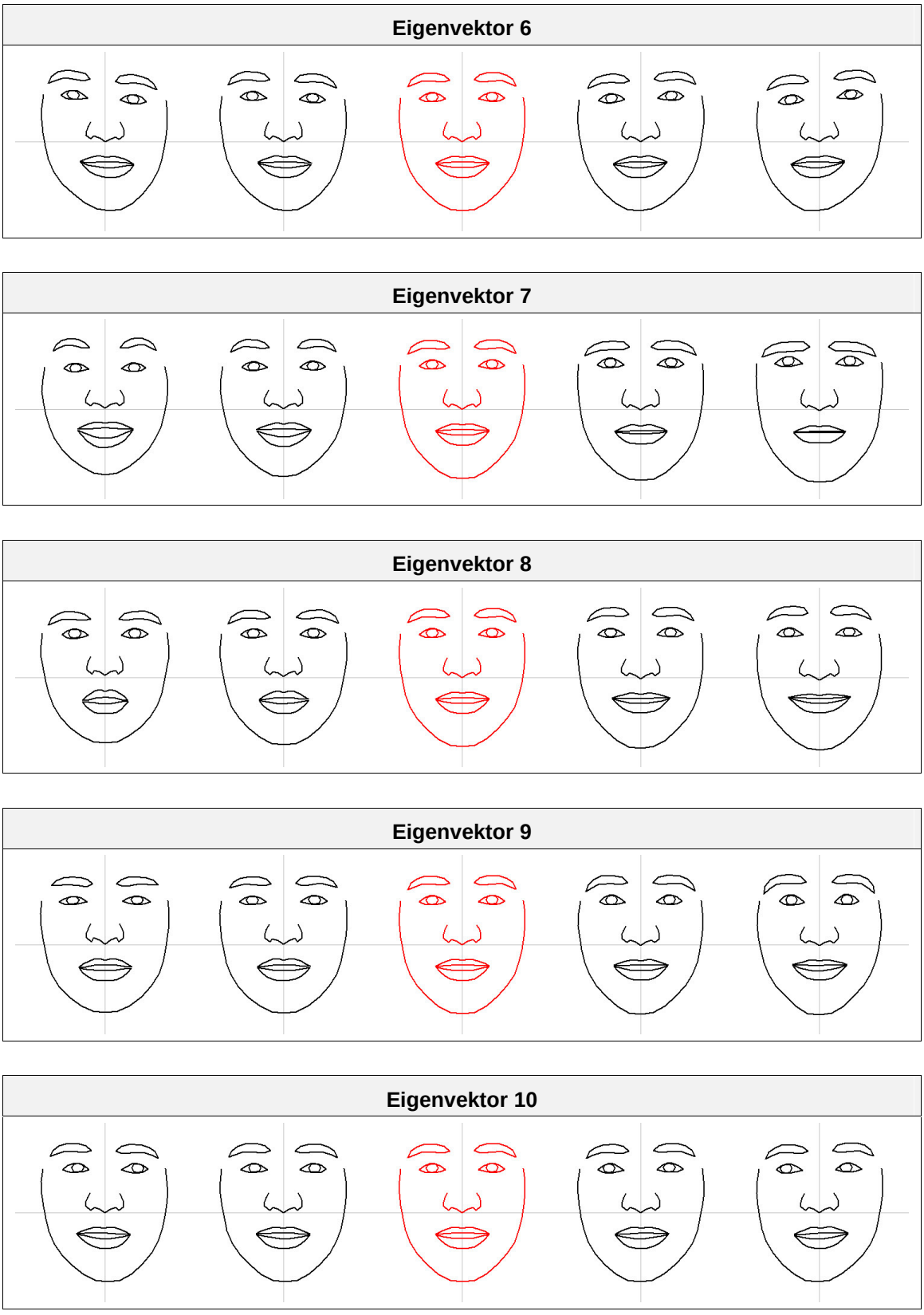


Bild 4.15: ***Veränderung der Gesichtsform bei Variation der ersten 10 Parameterwerte*** *im Bereich –2.0 ... +2.0 Standardabweichungen. Jeder Eigenvektor beschreibt charakteristische Formvariationen des mittleren Gesichts (rot).*

Wie auch beim Modell für die menschliche Silhouette beschreibt jeder Eigenvektor eine charakteristische Veränderung der mittleren Modellkontur, wobei die Größe der Veränderung mit kleiner werdendem Eigenwert abnimmt. Eigenvektor 1 modelliert die Drehung des Gesichts nach rechts und links. Eigenvektor 2 beschreibt das Heben und Senken des Kopfes sowie die Breite des Gesichts. Für die übrigen Eigenvektoren ist eine Zuordnung oft nicht eindeutig. So wird beispielsweise das Öffnen und Schließen des Mundes durch die Eigenvektoren 5 und 7 beschrieben, wobei hier jedoch auch, genauso wie bei Eigenvektor 4, ein Effekt auf die Variation der Augenbrauen zu sehen ist.

Tabelle 4.6: ***Effekt der Eigenvektoren auf die Gesichtsform****. Bei Variation des Parameterwertes b_i innerhalb der Grenzen $\underline{b}_i$ ... $\overline{b}_i$ (in Vielfachen der Standardabweichungen) erhält man gerade noch zulässige Gesichtsformen, wenn die übrigen Parameterwerte b_j mit $j \neq i$ auf 0 gesetzt werden.*

Eigenvektor	***Eigenwert***	$\underline{b}_i$	$\overline{b}_i$	***Effekt***
1	20467	-2.0	2.0	Kopf links/rechts drehen
2	9257	-2.0	3.0	Kopf heben/senken
3	5668	-3.0	1.6	grinsen / Mund zu ‚o' formen Augenbrauen heben/senken
4	2999	-2.0	1.8	Augenbrauen heben/senken und Abstand vergrößern/verkleinern
5	2460	-2.5	1.0	Mund auf/zu; Augenbrauen senken/heben
6	1743	-2.5	2.5	Asymmetrie links/rechts Mund + Augen rauf/runter
7	1636	-2.5	2.5	Augenbrauen rauf/runter; Mund auf/zu
8	1256	-3.5	3.0	Lächeln / Mund zu ‚u' formen Augenbrauen rauf/runter
9	810	-3.0	3.0	Augenabstand, Augenbrauen innen/außen
10	592	-3.0	3.0	Pupillen rechts/links
11	529	-3.0	3.0	Augenbrauen klein/groß; Augen auf/zu Pupillen rauf
12	456	-3.0	3.0	Mund breit/schmal; Nase lang/kurz
13	453	-3.5	3.5	Pupillen links/rechts Mund links/rechts verschieben
14	379	-4.0	4.0	Augen auf/zu
15	269	-4.0	4.0	Verlauf der Augenbrauen
16	224	-4.0	4.0	Augen auf/zu
17	221	-4.0	4.0	Asymmetrie von Mund und Augen

Tabelle 4.6 zeigt für jeden Eigenvektor den zulässigen Wertebereich, innerhalb dessen b_i variiert werden kann, so dass noch zulässige Gesichtsformen entstehen, wenn die übrigen Werte b_j mit $j \neq i$ auf 0 gesetzt werden. Da bei der kombinierten Variation aller Parameterwerte jedoch wiederum unzulässige Formen entstehen, muss der zulässige Wertebereich für die Anwendung weiter eingeschränkt werden und kann je nach Bedarf für jeden Vektor separat angepasst werden.

4.5 Modellsuche im Bild

Die vorgestellten Punktverteilungsmodelle für die menschliche Silhouette und das Gesicht werden in Kapitel 5 zur Auswertung von Videobildfolgen eingesetzt. Der Einsatz des Silhouettenmodells für den Oberkörper erfolgt dabei innerhalb von STABIL++, das um Klassen und Methoden zur Realisierung von kantenbasierten Merkmalen erweitert wurde. Mit STABIL++ sollen in Bildsequenzen Personen detektiert und deren Silhouette verfolgt werden. Es werden mehrere zueinander kalibrierte Kameras cam_i verwendet. Durch Berücksichtigung der Stereoinformationen kann so auf die 3D-Position der Person im Raum geschlossen werden. Umgekehrt kann die geschätzte 3D-Position dazu verwendet werden, die ungefähre 2D-Lage der Silhouette im Bild vorherzusagen.

Das Auffinden von Modellinstanzen im Bild erfordert mehrere Schritte. Zunächst wird eine initiale Kontur S_{ini} an die Stelle im Bild platziert, an der mit einem Objekt gerechnet wird. Da die erwartete Position nur grob angegeben werden kann, wird im Folgenden ausgehend von den Punkten der initialen Kontur deren Form in einem iterativen Verfahren so lange verformt, bis sich eine hinreichend gute Übereinstimmung mit den Bildkanten ergibt. Dabei darf die neu erzeugte Kontur nur Formen annehmen, die konsistent mit dem Modell sind, d.h. deren Parameterwerte b_i im zulässigen Bereich liegen (vgl. Abschnitt 4.3.3). Darüber hinaus ist das Ergebnis einer Suche zu verwerfen, wenn im Kantenbild keine Strukturen vorhanden sind, die mit dem Modell übereinstimmen. Hierzu wird ein geeignetes Qualitätsmaß definiert, welches eine Bewertung der Güte der Anpassung des Modells zulässt.

4.5.1 Iterationsschritt bei der PDM-Suche

Bei jedem Iterationsschritt (siehe Algorithmus S. 84) werden Position und Form der Kontur in Konsistenz mit dem zugrunde liegenden PDM an die Bildstrukturen angepasst. In jedem Konturpunkt wird entlang der Normalen zum Konturverlauf zunächst nach Kanten im Bild gesucht, um so neue, verbesserte Positionen für die Modellpunkte zu erhalten. Die sich ergebende Kontur S_{edge} ist im Allgemeinen nicht konsistent mit dem PDM, d.h. die Form kann aufgrund der Beschränkung der Eigenwerte und Eigenvektoren nicht durch das PDM beschrieben werden. Deshalb werden die Formparameter

der Kontur so angepasst, dass die gefundenen Kantenpunkte einerseits möglichst gut approximiert werden, die sich ergebenden Parameterwerte b_i andererseits aber dennoch im zulässigen Bereich liegen.

Hierzu werden durch eine geeignete Transformation $T(s,\Theta,t_x,t_y)$ zunächst Skalierung *s*, Rotation Θ und Translation (t_x,t_y) der Kontur so verändert, dass S_{edge} möglichst gut approximiert wird (*shape alignment*, siehe Anhang B). Die dann noch bestehenden Unterschiede zu S_{edge} werden durch eine Veränderung der Formparameter b_i minimiert ($\vec{b} \rightarrow \vec{b} + d\vec{b}$, siehe Abschnitt 4.5.9). Die sich aus $T(s,\Theta,t_x,t_y)$ und $\vec{b} + d\vec{b}$ ergebende neue Kontur wird im nächsten Iterationsschritt als Startkontur verwendet.

Die Anpassung der Kontur an das Kantenbild wird so lange wiederholt, bis sich eine hinreichend gute Übereinstimmung mit den Bildstrukturen ergibt oder eine vorgegebene maximale Anzahl an Iterationen erreicht ist. Ist die Qualität der resultierenden Kontur größer als eine definierte Mindestqualität, so wird sie akzeptiert. Konnte hingegen keine Modellinstanz gefunden werden, wird das Ergebnis verworfen.

Algorithmus *Iterationsschritt bei der PDM-Suche*

```
// variables
S_edge                          // edge shape
S_res                           // result shape
T(s,Θ,t_x,t_y):                 // transformation for shape alignment
b                               // parameter vector
db                              // parameter vector adjustments

// algorithm: search()
b    ← shape_parameters(S_res):                  // get original shape parameters
S_edge ← edge_shape(S_res):                      // search edges along shape normals
T(s,Θ,t_x,t_y) ← shape_alignment(S_res,S_edge): // find transformation to map S_res to S_edge
transform(S_res,T(s,Θ,t_x,t_y)):                 // set transformation
db   ← parameter_adjustment(S_res,S_edge):       // calculate shape parameter adjustment
S_res ← S_res(b+db):                             // set new parameters
```

4.5.2 History Shapes

Für die iterative PDM-Suche im Bild sollte S_{ini} die Objektkanten möglichst gut approximieren. Zur initialen Detektion wird die mittlere Modellkontur $\bar{S}$ verwendet, deren Position und Größe heuristisch gewählt werden. Bei der Re-Detektion dagegen kann auf bereits gefundene und als *history shapes* gespeicherte Modellinstanzen zurückgegriffen werden. Aus den *history shapes* kann die erwartete Silhouettenform durch das in Abschnitt 4.5.4 beschriebene Verfahren prädiziert werden.

Bei der Verwendung mehrerer Kameras variiert aufgrund der verschiedenen Blickrichtungen das Aussehen der Kontur. Bei der Personenverfolgung in STABIL++ (vgl. Abschnitte 4.5.3 und 5.1) werden die bereits gefundenen Instanzen daher für jede Kamera separat verwaltet. Zusammen mit der Vorhersage der 3D-Position für das lokale Koordinatensystem des Kopfes ergibt sich so eine Schätzung für S_{ini}, die den Kantenverlauf im folgenden Bild oft schon gut approximiert. Damit wird die Suche schneller und wesentlich robuster.

4.5.3 PDM-Suche in STABIL++

Die Objektmodellteile (OMPs) des in einer Szene hinterlegten Objektmodells sind mit Merkmalen (*features*) ausgestattet, die sie charakterisieren und die im Laufe des Interpretationsprozesses in den Bildern wiedergefunden werden müssen (vgl. Abschnitt 3.3.6). Jedes Merkmal verfügt über eine Reihe von Bildverarbeitungsoperatoren, die es ihm ermöglichen, sich in den verschiedenen Kamerabildern zu extrahieren.

Zur Initialisierung der PDM-Suche wird zunächst eine initiale Kontur S_{ini} an eine geeignete Stelle ins Bild positioniert, die in etwa dem erwarteten Kantenverlauf im Bild entsprechen sollte. Von dieser Stelle ausgehend werden Form und Position iterativ an die Bildstrukturen angepasst. Für die Initialisierung müssen für jede Kamera, in der das OMP sichtbar ist, Position, Größe und Form von S_{ini} ermittelt werden. Bei der Berechnung der Größe und Form von S_{ini} wird zwischen initialer Detektion und Re-Detektion bereits gefundener Modellinstanzen unterschieden. Bei der initialen Detektion wird für die Größe der im hinterlegten Objektmodell angegebene (mittlere) Wert verwendet. Die Komponenten des Parametervektors $\vec{b}$ werden mit 0 initialisiert, es wird also die mittlere Kontur $\overline{S}$ des PDM verwendet. Wurden jedoch bereits Modellinstanzen gefunden, so kann bei der Re-Detektion auf dieses Wissen zurückgegriffen werden. Form und Position von S_{ini} können dann aus den *history shapes* prädiziert werden (vgl. Abschnitt 4.5.2).

S_{ini} wird für die anschliessende iterative Suche verwendet. Die Suche wird so lange fortgesetzt, bis entweder die Qualität der approximierten Silhouette eine Mindestqualität übersteigt oder bis eine bestimmte Anzahl an Iterationen erreicht ist. Die gefundene Silhouette S_{res} wird akzeptiert und als *history shape* der entsprechenden Kamera gespeichert, wenn die Qualität der Anpassung größer als die Mindestqualität ist. Der Gesamtalgorithmus für die PDM-Suche in STABIL++ ist auf Seite 86 dargestellt.

Algorithmus *PDM-Suche in STABIL++*

```
// variables
S_ini                 // initial shape for PDM search
S_res                 // result shape from search
quality               // quality of shape fit to image data
min_quality           // minimum quality of shape fit to image data
iterations            // maximum number of iterations during search
history_shapes[]      // sequence of found shapes for each camera
i                     // counter variable

// algorithm
for each camera cam_i do
{
    quality ← -1.0:                          // initialization

    S_ini ← compute_initial_shape(cam_i):    // compute initial PDM shape

    S_res ← presearch(S_ini):                // optimize start position

    // search until quality is high enough or maximum number of iterations is reached
    for ( i←0; (i<iterations-1) AND (quality<min_quality); i++ )
    {
       S_res ← search(S_res):                // PDM search iteration
       quality ← calc_quality(S_res):        // calculate shape quality
    }

    if (quality >= min_quality)
       history_shapes[cam_i] ← S_res         // accept search result
}
```

Die Optimierung von Position und Form der Startsilhouette erfolgt in zwei Schritten: Zunächst wird durch Projektion der geschätzten 3D-Position des OMP ins Bild die Position ermittelt, an der die Silhouette erwartet wird. Im zweiten Schritt werden Größe und Form angepasst.

Positionierung der initialen Silhouette

Die PDM-Suche konvergiert am schnellsten, wenn die Startsilhouette so positioniert wird, dass sie den Kantenverlauf im Bild möglichst gut approximiert (siehe Bild 4.17). Jedes OMP verfügt über einen dreidimensionalen Suchraum, der die ungefähre Position des Ursprungs seines lokalen Koordinatensystems im Raum angibt und mit dem der Bereich eingeschränkt werden kann, innerhalb dessen das OMP erwartet wird. Zusätzlich zu dem 3D-Suchraum ist jedes OMP mit einem Vorhersagefilter für seine 3D-Position ausgestattet, der die erwartete Position für jeden Interpretationszyklus aus be-

reits gemessenen *history*-Werten ermittelt[14]. Für die Vorhersage der Position kommt zum einen ein einfacher Filter zum Einsatz, der die neue Position linear aus den *history*-Einträgen extrapoliert und annimmt, dass sich das OMP mit gleicher Richtung und Geschwindigkeit weiterbewegt. Alternativ besteht die Möglichkeit, die Vorhersage über einen Kalman-Filter durchzuführen. Für jeden Interpretationsschritt werden Position und Größe des Suchraums neu bestimmt, und es wird ein Qualitätsmaß berechnet, das die Güte der Schätzung angibt. Die Qualität der Schätzung berücksichtigt einerseits die Qualität der *history*-Einträge, des weiteren wird angenommen, dass die Güte der Vorhersage mit größer werdendem Vorhersagezeitraum abnimmt. Weitere Details sind in [Rid99] beschrieben.

Die Position der initialen Kontur für die PDM-Suche ergibt sich durch Projektion des geschätzten 3D-Punktes für die Position des OMP in das jeweilige Kamerabild.

Größenanpassung

Die Größenanpassung berücksichtigt, dass eine Person mit abnehmender Entfernung zur Kamera größer erscheint. Eine *history shape* muss daher entsprechend skaliert werden.

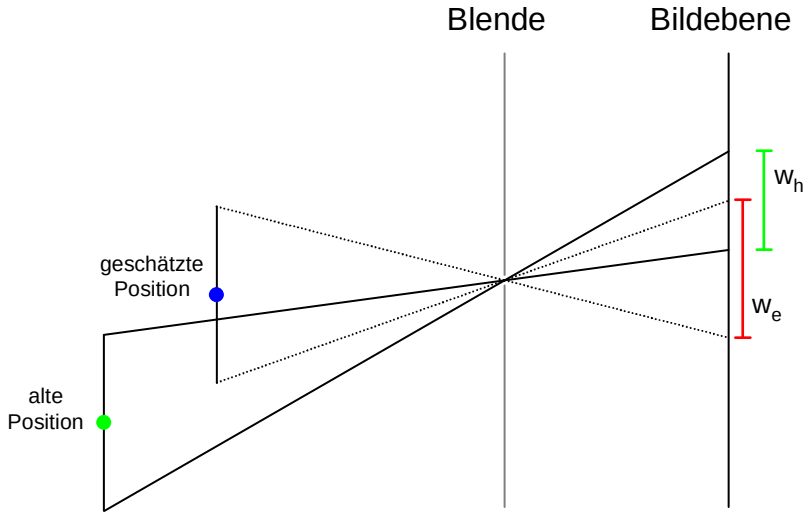


Bild 4.16: ***Berechnung des Skalierungsfaktors aufgrund der Bewegung der Person (scale from motion)****. An der alten und neuen (geschätzten) Position für ein Objektmodellteil wird eine Kugel (hier nur eindimensional dargestellt) ins Bild projiziert. Der Skalierungsfaktor, der sich aufgrund der Bewegung der Person ergibt, beträgt* w_e/w_h.

Bild 4.16 verdeutlicht die Ermittlung des entsprechenden Skalierungsfaktors (*scale from motion*). An der alten 3D-Position (grün markiert) und an der geschätzten neuen Positi-

[14] Die *history* für die 3D-Position der einzelnen OPMs ist nicht mit den in Abschnitt 4.5.2 vorgestellten *history shapes* zu verwechseln.

on (blau markiert) wird eine Kugel (in der Zeichnung aus Gründen der Übersichtlichkeit nur eindimensional als Strecke skizziert) ins Bild projiziert. Aus den entsprechenden Größen im Bild ergibt sich der Skalierungsfaktor zu w_e/w_h.

Bild 4.17 zeigt die Positionierung der initialen Silhouette. Die im letzten Bild gefundene Silhouette (grün) wird an die prädizierte Position für das Objektmodellteil Kopf (blau markiert) verschoben und entsprechend der Bewegung der Person skaliert. Die so erhaltene Silhouette (rot) wird für die iterative Modellsuche verwendet. Das Bild zeigt auch das verstärkte Auftreten von Kanten im Inneren der Kontur, da sich die Person vor einem stark strukturierten Hintergrund bewegt und vom aktuellen Kamerabild ein Hintergrundbild subtrahiert wurde.

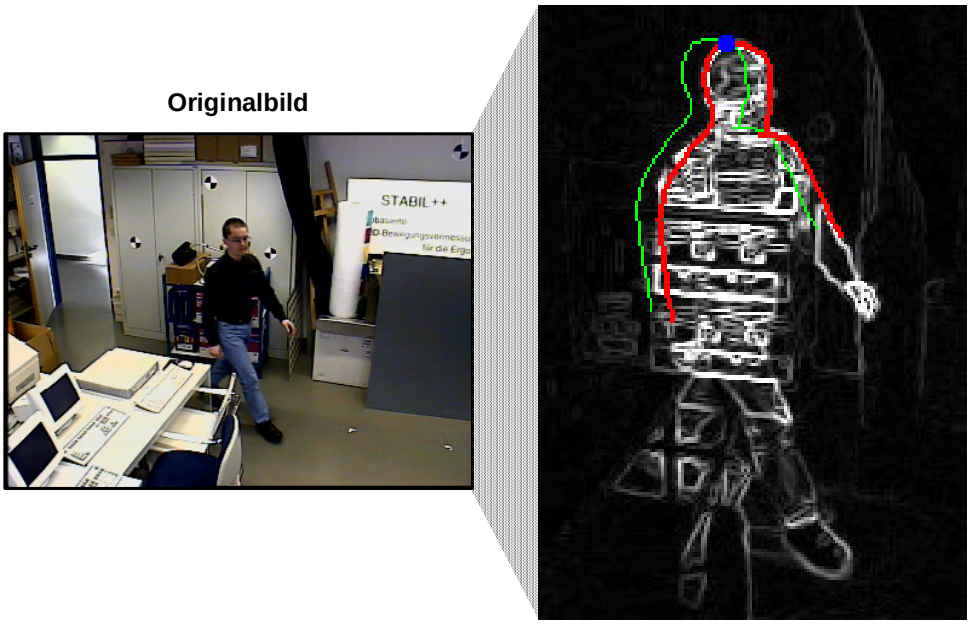


Bild 4.17: ***Positionierung der initialen Silhouette****. Die zuletzt gefundene und in den history shapes gespeicherte Silhouette (grün) wird entsprechend der geschätzten neuen Position des Kopfes (blau markiert) verschoben und in der Größe angepasst. Im Bild ist auch das verstärkte Auftreten von Kanten im Inneren der Kontur zu sehen, da eine Hintergrundschätzung verwendet wird und sich die Person vor stark strukturiertem Hintergrund bewegt.*

4.5.4 Prädiktion der Modellparameter

Um den realen Konturverlauf im Bild für die PDM-Suche bestmöglich zu approximieren, wird zusätzlich zur Vorhersage von Position und Größe der initialen Kontur auch die erwartete Kontur-Form im nächsten Bild prädiziert. Die Beobachtung, dass die einzelnen Eigenvektoren des Modells charakteristische Formvariationen der mittleren Kontur beschreiben (siehe Seite 74 und 81), legt nahe, dies über eine Vorhersage der Modellparameter zu tun. Der Parametervektor $\vec{b}$ wird dazu für jede Kamera nach einem

Geschwindigkeits-Beschleunigungsmodell unter Berücksichtigung der *history*-Einträge geschätzt. Je nach Größe der *history* wird auf 2, 3 oder 4 Einträge zurückgegriffen und die Geschwindigkeit $\vec{v}$, die Beschleunigung $\vec{a}$ sowie die zeitliche Variation von $\vec{a}$ der Parametervektoren berücksichtigt. Aus der kinematischen Gleichung

$$\vec{b}_e = \vec{b}_0 + \left(\frac{\partial \vec{b}}{\partial t}\right)_{t=t_0} t + \frac{1}{2}\left(\frac{\partial^2 \vec{b}}{\partial t^2}\right)_{t=t_0} t^2 + \frac{1}{6}\left(\frac{\partial^3 \vec{b}}{\partial t^3}\right)_{t=t_0} t^3 \tag{4.26}$$

ergibt sich für die Schätzung $\vec{b}_e$:

$$\vec{b}_e = \begin{cases} 2\vec{b}_0 - \vec{b}_{-1} & 2\ \textit{history} - \text{Einträge} \\ \frac{5}{2}\vec{b}_0 - 2\vec{b}_{-1} + \frac{1}{2}\vec{b}_{-2} & 3\ \textit{history} - \text{Einträge} \\ \frac{8}{3}\vec{b}_0 - \frac{5}{2}\vec{b}_{-1} + \vec{b}_{-2} - \frac{1}{6}\vec{b}_{-3} & 4\ \textit{history} - \text{Einträge} \end{cases} \tag{4.27}$$

Hierin ist $\vec{b}_0$ der zuletzt gemessene Parameter-Vektor, $\vec{b}_{-1}, \vec{b}_{-2}$ und $\vec{b}_{-3}$ sind die Vektoren der weiter zurückliegenden *history*-Einträge. Die Zeitpunkte, zu denen die Bilder aufgenommen wurden, sind bei dieser Betrachtung als äquidistant angenommen.

Bild 4.18 veranschaulicht die Schätzung von $\vec{b}$ bei Verwendung von 2, 3 oder 4 *history*-Einträgen. Um bei der Prädiktion nur Konturformen zu erhalten, die mit dem Modell konsistent sind, wird der errechnete Parameter-Vektor so normiert, dass die Koordinatenwerte im zulässigen Variationsbereich liegen (vgl. Abschnitt 4.3.3).

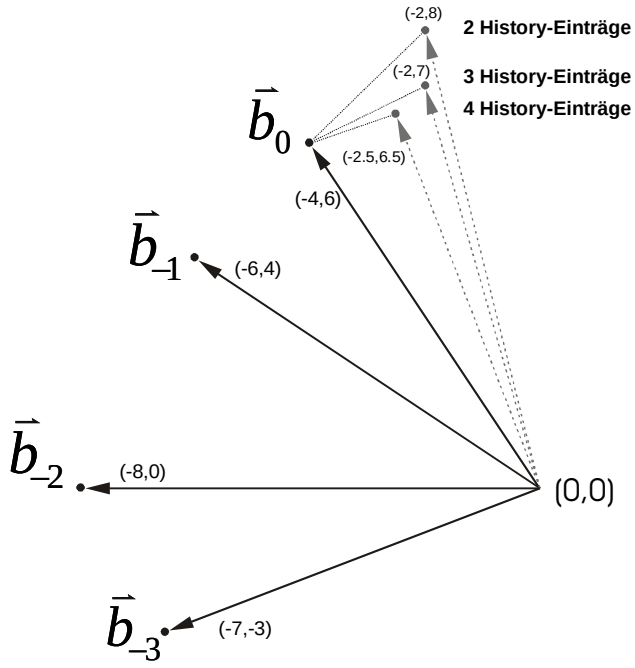


Bild 4.18: ***Prädiktion des Parametervektors aus den History-Einträgen*** $\vec{b}_i$ *. Die beiden letzten Vektoren ergeben durch lineare Prädiktion den Vektor (-2,8). Bei Berücksichtigung der Beschleunigung durch zusätzliche Verwendung von* $\vec{b}_{-2}$ *wird der Vektor (-2,7) vorhergesagt. Die Verwendung von 4 history-Einträgen führt auf (-2.5,6.5).*

4.5.5 Vorsuche bei einteiligen Konturen

In STABIL++ führt die Vorhersage der 3D-Position eines OMP dazu, dass die ermittelte Position von S_{ini} im Bild nicht exakt mit den Bildstrukturen übereinstimmt. Insbesondere kommt es häufig zu seitlichen Verschiebungen, so dass Form und Größe der Startsilhouette die Strukturen im Kantenbild zwar prinzipiell gut approximieren, die Silhouette insgesamt aber verschoben ist. Da auf der anderen Seite die lokale Anpassung des PDM die Position nur schwer verändern kann, wird daher zunächst auch die unmittelbare Umgebung des projizierten Ausgangspunktes untersucht.

Bei guter Positionierung der Startsilhouette wird eine Silhouette im Bild schon nach einer Iteration mit einer guten Qualität approximiert. In einer Vorsuche (*presearch*) wird S_{ini} daher zunächst um einen horizontalen und vertikalen Versatz an 8 Nachbarpositionen verschoben (siehe Bild 4.19). Die Größe des Versatzes wird relativ zur Größe von S_{ini} gewählt.

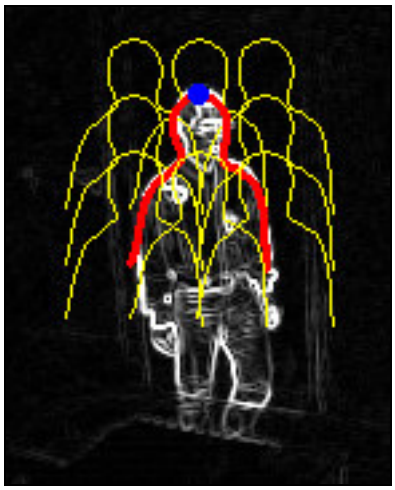


Bild 4.19: ***Vorsuche (presearch) bei einteiligen Konturen****. Zur Ermittlung der optimalen Ausgangsposition für die PDM-Suche wird in der Umgebung der geschätzten Position für den Kopf (blauer Punkt) jeweils eine Iteration durchgeführt. Der Abstand der verschobenen Silhouetten (gelb) zur Ausgangssilhouette (rot) ist hier übertrieben groß dargestellt.*

An jeder der insgesamt 9 Positionen wird S_{ini} daraufhin in *einem* Iterationsschritt an die Kantenstrukturen angepasst. Für die sich anschließende iterative Suche wird S_{ini} an die Position verschoben, für die die Vorsuche den besten Qualitätswert (siehe Abschnitt 4.5.10) ergeben hat.

Algorithmus *Vorsuche bei einteiligen Konturen*

```
// variables
S_ini                   // initial shape for PDM search
S_res                   // result shape; used for iterative search after presearch
S_tmp                   // temporary shape
quality                 // quality of shape fit to image data
q_tmp                   // temporary variable

// initialization
S_ini     ← compute_initial_shape():   // compute initial PDM shape
quality   ← -1.0:

// algorithm
for all 9 neighbour vectors v_i of S_ini do
{
     S_tmp ← translate(S_ini,v_i):          // shift initial shape to neighbour position
     S_tmp ← search(S_tmp):                 // one PDM search iteration
     q_tmp ← calc_quality(S_tmp):           // calculate quality of found shape
     if ( q_tmp > quality )
     {
        S_res ← S_tmp:                      // store found shape for usage in iterative search
        quality ← q_tmp:                    // quality of best neighbour position after 1 iteration
     }
}
```

4.5.6 Vorsuche bei mehrteiligen Konturen

Beim Gesichtsmodell unterliegen Größe und Form sowie die relativen Abstände der einzelnen Konturteile zueinander vielfältigen Variationsmöglichkeiten. Neben den Unterschieden zwischen verschiedenen Personen hängt das Aussehen eines Gesichtes im Bild von der Kameraposition und dem Gesichtsausdruck der betreffenden Person ab. Form und Position der einzelnen Gesichtsteile werden daher von der mittleren Modellsilhouette (siehe Bild 4.13) abweichen (vgl. auch Abschnitt 5.2.4).

Die in Abschnitt 4.5.1 beschriebene Suche während eines Iterationsschrittes ist eine lokale Suche, bei der die Bildstrukturen insbesondere dann gut approximiert werden, wenn die Position des Modells grob mit der Position im Bild übereinstimmt. Verschiebungen der Kontur oder von Konturteilen in vertikaler und horizontaler Richtung können von der Suche hingegen nur begrenzt erreicht werden. Aufgrund der vielen Freiheitsgrade tritt dieses Problem beim Gesichtsmodell für jedes einzelne Konturteil auf. Bei einteiligen Konturen, wie z.B. beim Modell für die menschliche Silhouette, wird die Suche auf 8 Nachbarpositionen in der Nähe der initialen Schätzung ausgedehnt und vor der eigentlichen iterativen Suche an jeder Position *eine* Iteration durchgeführt (siehe Bild 4.19), um so eine optimale Ausgangsposition zu finden. Ein analoges Verfahren führt auch bei mehrteiligen Konturen, wie z.B. dem Gesichtsmodell, zu verbesserten Suchergebnissen. Vor der eigentlichen iterativen Suche wird die Position der einzelnen Gesichtsteile optimiert (siehe Algorithmus auf Seite 93). Jedes Konturteil wird dabei um einen Vektor proportional zu seiner Größe an 8 Nachbarpositionen verschoben. Entlang der Senkrechten zum Konturverlauf wird nach Kanten im Bild gesucht und für jeden Punkt nach Gleichung (4.31) ein Qualitätsmaß ermittelt. Durch Summation der Punktqualitäten ergibt sich für jede der insgesamt 9 Positionen ein Qualitätsmaß. Jedes Konturteil wird schließlich an die Position mit der höchsten Qualität verschoben.

Da die so definierte Kontur im Allgemeinen nicht konsistent mit dem PDM-Modell ist, wird die Form mit dem in Abschnitt 4.5.9 beschriebenen Verfahren durch eine gültige Modellinstanz approximiert. Die eigentliche iterative Suche wird schließlich mit dieser Modellinstanz durchgeführt.

Eine Erweiterung dieses Verfahrens könnte darin bestehen, jedes Konturteil oder eine Gruppe von Konturteilen (beispielsweise das komplette Auge, bestehend aus Pupille und Auge) durch ein eigenes Punktverteilungsmodell zu beschreiben. Die an der besten Position ermittelten Kantenpunkte könnten dann durch eine für das Konturteil bzw. die Gruppe von Konturteilen gültige Modellinstanz approximiert werden.

Algorithmus *Vorsuche bei mehrteiligen Konturen*

```
// variables
S_ini          // initial shape for PDM iterations
S_res          // result shape; used for iterative search after presearch
S_adj          // shape with adjusted shape parts
e              // best edges found for part
v_best         // best neighbour vector for shape part adjustment
quality        // best neighbour edge quality for shape part
q_tmp          // temporary variable

// initialization
S_ini ← compute_initial_shape():        // compute initial PDM shape
S_adj ← S_ini:

// algorithm
for each contour part part_i do
{
    quality ← -1.0:                     // initialization

    for all 9 neighbour vectors v_ij for part_i do
    {
       translate_part(S_adj,part_i,v_ij):   // shift part to neighbour position
       e     ← search_edges(part_i):        // search strongest edge for each point
       q_tmp ← calc_quality(e):             // calculate quality for edges of part i

       if ( q_tmp > quality )
       {
         v_best ← v_ij:                 // store best neighbour vector
         quality ← q_tmp:               // quality for best neighbour position
       }

       translate_part(S_adj,part_i,-v_ij):  // shift part back to initial position
    }

    translate_part(S_adj,part_i,v_best):    // shift part to best neighbour position
}

S_res ← approximate_shape(S_ini,S_adj):     // map S_ini onto S_adj in
                                            // consistency with PDM model
```

4.5.7 Kantensuche

Da die Konturpunkte den Umriss des im Bild zu suchenden Objektes beschreiben, werden die einzelnen Konturpunkte bei einer guten Anpassung des Modells an die Bildstrukturen auf einer Bildkante zu liegen kommen. Um für jeden Punkt eine verbesserte Position zu finden, wird in jedem Konturpunkt entlang einer Geraden senkrecht zum Konturverlauf nach Kanten im Bild gesucht. Die Berechnung des Kantenbildes erfolgt je nach Anwendung unterschiedlich und wird in den jeweiligen Abschnitten separat beschrieben.

Größe des Suchintervalls

Die Länge des Suchintervalls ist nicht für jeden Konturpunkt gleich groß, sondern wird in Abhängigkeit von der Größe des jeweiligen Konturteils gewählt. Die Punkte, die ein Konturteil definieren, ergeben sich aus den *connection types* der Konturpunkte (siehe Tabelle 4.2).

Bezeichnen $x_{min}(n)$, $x_{max}(n)$, $y_{min}(n)$ und $y_{max}(n)$ die minimalen bzw. maximalen Koordinatenwerte für das Konturteil n, so sind Breite $w(n)$ und Höhe $h(n)$ definiert als

$$w(n) = x_{\max}(n) - x_{\min}(n) + 1 \quad , \quad h(n) = y_{\max}(n) - y_{\min}(n) + 1 \tag{4.28}$$

Die Größe $s(n)$ des Konturteils ist definiert als das geometrische Mittel von Breite und Höhe:

$$s(n) = \sqrt{w(n) \cdot h(n)} \tag{4.29}$$

Der Suchradius

$$r_{search}(n) = c \cdot s(n) \tag{4.30}$$

wird relativ zu $s(n)$ gewählt und kann je nach PDM und Anwendung angepasst werden. In jedem Punkt wird jeweils um $r_{search}(n)$ nach innen und außen gesucht (siehe Bild 4.20).

Bild 4.20 zeigt die Größe der Suchintervalle für die verschiedenen Konturteile am Beispiel des 10-teiligen Gesichtsmodells. Die Abhängigkeit des Suchradius von der Größe des jeweiligen Konturteils führt dazu, dass für Konturpunkte, die zu einem vergleichsweise großen Konturteil gehören (wie z.B. die Kinnpartie), innerhalb einer größeren Umgebung nach Kanten gesucht wird. Bei kleinen Modellteilen, wie z.B. den Pupillen, bleibt die Suche auf ein kleineres Intervall beschränkt. Die Konstante c aus Gleichung (4.30) wurde in Bild 4.20 zu $0{,}13$ gewählt. Bei der Suche nach Silhouetten des menschlichen Oberkörpers liefert ein empirisch ermittelter Wert von $c = 0{,}2$ gute Resultate (siehe auch Bild 4.23).

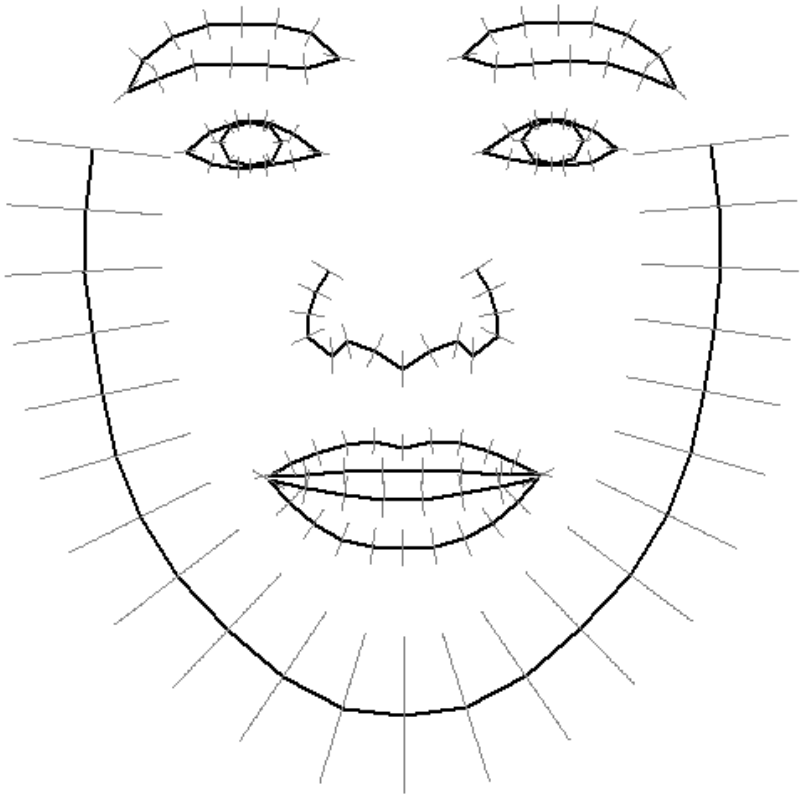


Bild 4.20: ***Suchintervall für die Kantensuche*** *am Beispiel des Gesichtsmodells. Die Größe der Suchintervalle (grau) wird abhängig von der Größe des jeweiligen Konturteils gewählt (im Beispiel ist c=0,13).*

Blowing

Das System STABIL++ erlaubt es, bei der Bildauswertung durch die Verwendung eines adaptiven Hintergrundschätzers eine Segmentierung der Bildpixel in eine Vordergrund- und Hintergrundregion vorzunehmen [RMK95]. Bei Subtraktion eines Hintergrundbildes vom aktuellen Kamerabild kommt es zu einem vermehrten Auftreten von Kanten, wenn sich ein Objekt vor strukturiertem Hintergrund bewegt oder wenn das Objekt selbst eine starke Textur aufweist. Bild 4.17 zeigt das dadurch verstärkte Auftreten von Kanten im Inneren des zu detektierenden Objektes.

Bei der Detektion und Verfolgung von Personen in Videosequenzen mit dem in Abschnitt 4.4.6 vorgestellten Modell für die menschliche Silhouette bietet es sich an, die Suche nach Kanten im Bild von außen durchzuführen und die initiale Silhouette etwas zu vergrößern. Diese als *blowing* bezeichnete Skalierung kann entweder vom Schwerpunkt aus erfolgen (*blowing from center of gravity*) oder senkrecht zum Konturverlauf (*orthogonal blowing*)(siehe Bild 4.21). Hierdurch kommt es bei der Berechnung der Punktqualität bei der Kantensuche mittels Gleichung (4.31) zu einer Verschiebung der

Qualitätskurve nach außen. Äußere Pixel mit demselben Abstand und Grauwert im Kantenbild werden also stärker gewichtet als Pixel, die sich im Inneren der Silhouette befinden. So kann vermieden werden, dass beispielsweise die Arme nach innen gezogen werden und sich an starke Kanten anpassen, die z.B. von Schränken oder Türen herrühren. Vorteile ergeben sich auch bei der Anpassung des Kopfes, bei dem es oft zu einer stark ausgeprägten Kante am Haaransatz kommt. Durch die Suche von außen ergibt sich eine bessere Anpassung der Silhouette an die meist schwächer ausgeprägte Kante, die vom Übergang Haare-Hintergrund herrührt. Die Anpassung an den Übergang Haare-Gesicht kann vermieden werden.

Im Fall des Modells für die menschliche Silhouette ist die Skalierung der Silhouette senkrecht zum Konturverlauf der Skalierung vom Schwerpunkt aus vorzuziehen. Bei der Skalierung vom Schwerpunkt aus kommt es insbesondere bei den Armen zu einer Verschiebung der Punkte parallel zum Konturverlauf, die dazu führt, dass die Arme bei wiederholter Suche immer länger werden. Dieser Effekt kann durch *orthogonal blowing* vermieden werden (vgl. Bild 4.21).

Die Größe der Skalierung wird relativ zur Größe der Silhouette ermittelt, die sich nach Gleichung (4.29) ergibt. Empirisch ermittelte Werte zwischen *0,1* und *0,2* führen zu einer robusten Anpassung an die gewünschten Bildstrukturen.

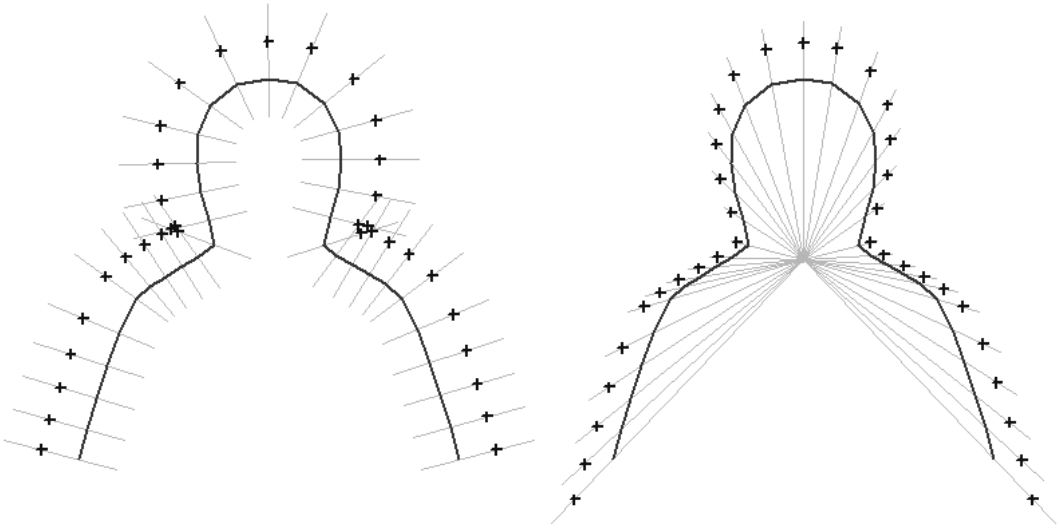


Bild 4.21: ***Blowing****. Da bei Subtraktion eines Hintergrundbildes verstärkt Kanten im Inneren einer Silhouette auftreten, können die Startpunkte für die Kantensuche nach außen verschoben werden. Links: Verschiebung senkrecht zum Konturverlauf. Rechts: Verschiebung entlang vom Schwerpunkt ausgehender Vektoren.*

Berechnung der Güte einzelner Punkte

Die einfachste Möglichkeit, eine möglichst gute Kante zu bestimmen, wäre, dasjenige Pixel auf der Suchgeraden zu ermitteln, für das die Kantenstärke den größten Wert hat. Diese Vorgehensweise führt jedoch dazu, dass weit entfernt liegende Kanten genauso

berücksichtigt werden wie Kanten, die nahe am Ausgangspunkt liegen und daher zu bevorzugen wären. Um weit entfernt liegende Ausreißer zu vermeiden, wird jeder Punkt *p* auf der Suchgeraden mit einer Qualitätsfunktion bewertet, die umso größer ist, je höher der Grauwert $g_{edge}(p)$ im Kantenbild ist und je näher die gefundene Position an der Ausgangsposition liegt:

$$q_{edge}(p) = \frac{g_{edge}(p)}{1 + \frac{d(p)}{r_{search}}} \tag{4.31}$$

Hierbei ist $d(p)$ der Abstand des Pixels *p* vom Ausgangspunkt.

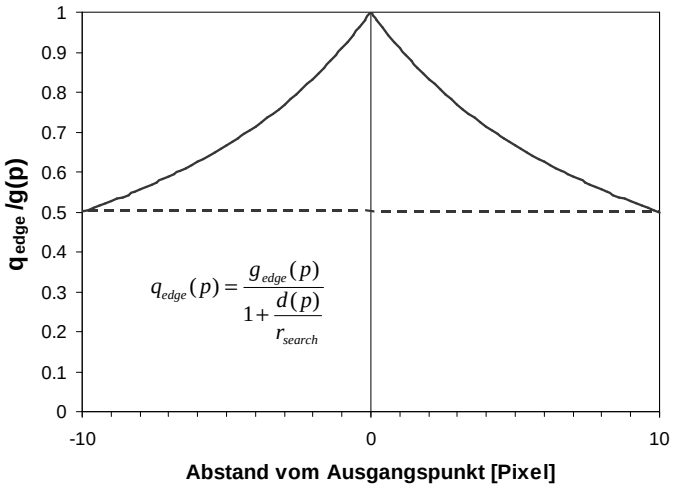


Bild 4.22: ***Berechnung der Punktqualität****. Die Güte einer Kante ist umso größer, je höher der Grauwert $g_{edge}(p)$ des betreffenden Pixels p im Kantenbild ist und je näher p am Ausgangspunkt liegt. Am Rand des Suchbereiches (im Beispiel: $r_{search}=10$) erreicht die Punktqualität einen minimalen Wert von 0.5.*

Der Punkt mit der höchsten Qualität auf der Suchgeraden wird schließlich als Kandidat für die verbesserte Punktposition gewählt. Die so gefundenen Punkte bilden die (im allgemeinen nicht mit dem PDM konsistente!) Form S_{edge}, an welche die Kontur durch *shape alignment* und Veränderung der Formparameter angepasst wird.

Die Punktqualitäten von S_{edge} werden darüber hinaus als Gewichtsfaktoren für die im folgenden Abschnitt beschriebene Ausrichtung von zwei Konturen verwendet. Bei der Anpassung der Formparameter (siehe Abschnitt 4.5.9) muss die Inverse der Matrix P^TWP durchgeführt werden (siehe Gleichung (4.45)). Die durch die Gewichtsfaktoren gebildete Diagonalmatrix *W* in Gleichung (4.44) muss daher vollen Rang haben, d.h.

$$rang(W) \overset{!}{=} 2n \tag{4.32}$$

Für den Fall, dass entlang der gesamten Suchgeraden im Bild die Pixel des Kantenbildes den Grauwert 0 haben, oder wenn die Kontur ganz oder teilweise außerhalb des Bildes liegt, wird jedem Punkt daher eine Mindestqualität $q_{edge}^{\min} = 1/g_{edge}^{\max}$ zugewiesen, wobei der maximale Grauwert des Kantenbildes in der Regel $g_{edge}^{\max} = 255$ beträgt. Die Position des neuen Punktes bleibt dabei unverändert.

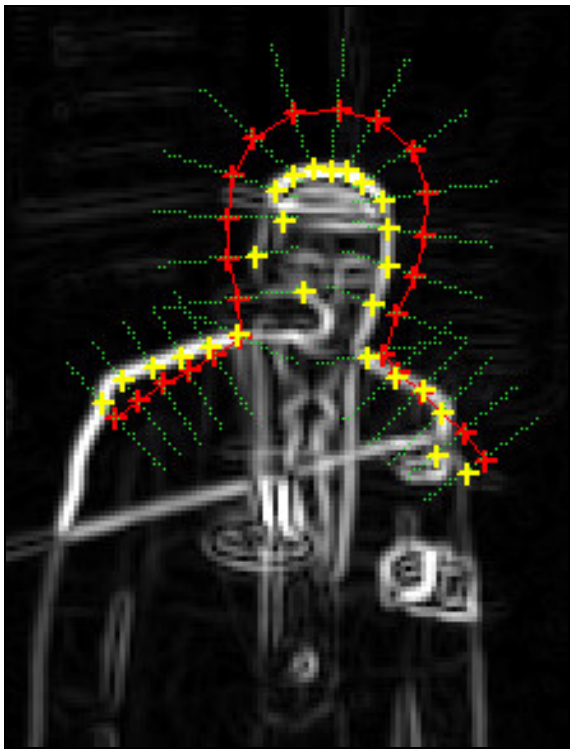


Bild 4.23: ***Kantensuche****. Ausgehend von der initialen Silhouette (rot) wird bei einem Iterationsschritt für jeden Konturpunkt entlang der Senkrechten zum Konturverlauf (grün) nach Kanten im Bild gesucht; die gefundenen Kantenpunkte sind gelb dargestellt.*

Bei der Suche nach Personensilhouetten erscheint es nicht sinnvoll, das Punktverteilungsmodell um zusätzliche a priori Informationen über den Verlauf der Grauwerte entlang der Suchgeraden zu erweitern. Das Aussehen ist in Abhängigkeit von der Kleidung der Person und der Strukturierung des Hintergrundes starken Variationen unterworfen. Daher sind keine sinnvollen Heuristiken möglich. Beim Gesichtsmodell hingegen können zusätzliche Informationen ins Modell mit aufgenommen werden, um so eine bessere Approximation der Bildstrukturen zu erreichen. Die Punktqualität $q_{edge}(p)$ wird in diesem Fall, entsprechend der zusätzlich im Modell hinterlegten Information über den Grauwertverlauf, modifiziert (vgl. Abschnitt 5.2.3).

4.5.8 Ausrichtung an den Kantenpunkten

Die Punkte der bei der Kantensuche gefundenen Kontur S_{edge} geben für jeden Modellpunkt eine verbesserte Position an, durch welche die Kontur an Objektkanten im Bild angepasst werden kann. Die Form von S_{edge} ist jedoch im Allgemeinen nicht konsistent mit dem Modell, sondern kann durch dieses aufgrund der Beschränkung der Parameterwerte und der Anzahl der Eigenvektoren nur angenähert werden. Ziel ist es daher, neue Formparameter $\vec{b}$ zu bestimmen, die zu einer möglichst guten Übereinstimmung mit S_{edge} führen. Um alle nicht die Form beschreibenden Parameter zu eliminieren, wird die Kontur zunächst so skaliert, rotiert und verschoben, dass die gewichtete Summe der Abstandsquadrate korrespondierender Punkte minimiert wird (vgl. auch [CTC92]).

Die Ausrichtung von zwei Konturen S_1 und S_2 wird durch eine Transformation

$$S_2 = T(s, \Theta, t_x, t_y)(S_1) \tag{4.33}$$

mit Skalierungsfaktor s, Rotationswinkel Θ und Translation (t_x,t_y) beschrieben. Für jeden Konturpunkt $(x\ y)^T$ gilt:

$$T(s, \Theta, t_x, t_y)\begin{pmatrix} x \\ y \end{pmatrix} = s\begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{4.34}$$

Mit dem durch

$$\vec{c}_j = \frac{1}{n}\sum_{i=1}^{n} \vec{r}_{ji} \quad mit \quad \vec{r}_{ji} = \begin{pmatrix} x_{ji} \\ y_{ji} \end{pmatrix} \tag{4.35}$$

definierten Schwerpunkt einer Kontur S_j lässt sich der Abstand eines transformierten Punktes von S_1 zum korrespondierenden Punkt von S_2 durch

$$\begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = s\begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix}\begin{pmatrix} x_{1i} - c_{1x} \\ y_{1i} - c_{1y} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} c_{1x} \\ c_{1y} \end{pmatrix} - \begin{pmatrix} x_{2i} \\ y_{2i} \end{pmatrix} \tag{4.36}$$

ausdrücken. Im Folgenden wird aus Gründen der Übersichtlichkeit ohne Beschränkung der Allgemeinheit ein im Schwerpunkt von S_1 zentriertes Koordinatensystem, d.h.

$c_{1x} = c_{1y} = 0$, angenommen. Rotationswinkel und Translationsvektor lassen sich dann anschaulich leicht interpretieren.

Bei der Berechnung der Summe der Abstandsquadrate werden die einzelnen Konturpunkte mit einem Gewichtsvektor $\vec{w} = (w_1 \ldots w_n)^T$ unterschiedlich gewichtet. Zusammen mit den Definitionen $a=s\cdot\cos(\Theta)$ und $b=s\cdot\sin(\Theta)$ führt dies auf den zu minimierenden Ausdruck

$$\begin{aligned} D &= \sum_{i=1}^{n} w_i (dx_i^{\,2} + dy_i^{\,2}) \\ &= \sum_{i=1}^{n} w_i (ax_{1i} - by_{1i} + t_x - x_{2i})^2 + \sum_{i=1}^{n} w_i (bx_{1i} + ay_{1i} + t_y - y_{2i})^2 \end{aligned} \tag{4.37}$$

Die Einträge von $\vec{w} = (w_1 \ldots w_n)^T$ sind die Punktqualitäten, die bei der Kantensuche entlang der Senkrechten zum Konturverlauf berechnet werden (Gleichung (4.31)). So wird erreicht, dass Punkte, deren neue Position einer guten Bildkanten entspricht, stärker ins Gewicht fallen als Punkte, deren neue Position auf einer schlechten Kante liegt.

Für die Minimierung von *D* wird ein *least squares* Ansatz verwendet, der auf die gesuchten Transformationsparameter *s*, Θ, t_x und t_y führt (siehe Anhang B).

4.5.9 Anpassung der Formparameter

Bei der in Abschnitt 4.5.7 beschriebenen Kantensuche wird für jeden Punkt $(x_i\ y_i)$ einer Kontur *S* ein Verschiebungsvektor $(\Delta x_i\ \Delta y_i)$ bestimmt, in dessen Richtung die Anpassung erfolgen sollte. Die gefundene Kontur S_{edge} lässt sich als

$$S_{edge} = S + \Delta S \tag{4.38}$$

schreiben, mit $\Delta S = (\Delta x_1 \Delta y_1 \ldots \Delta x_n \Delta y_n)^T$. *S* und S_{edge} sind hierbei im Bildkoordinatensystem, d.h. als Pixelwerte angegeben.

Im Folgenden wird mit S_{PDM} diejenige Kontur bezeichnet, die *S* im lokalen Koordinatensystem des Punktverteilungsmodells beschreibt. Dieses Modellkoordinatensystem kann z.B. im Schwerpunkt der mittleren Kontur zentriert sein oder sich aus den Bildkoordinaten ergeben, aus denen das PDM erzeugt worden ist.

S_{PDM} wird dabei aus Verformung der mittleren PDM-Kontur durch den zu *S* gehörenden Parametervektor $\vec{b}$ (vgl. Formel (4.11)) gebildet. Mit dem in Abschnitt 4.5.8 beschrie-

benen Alignment wird eine Transformation $T(s,\Theta,t_x,t_y)$ berechnet, die S_{PDM} möglichst gut auf S_{edge} abbildet. Die verbleibenden Unterschiede zu den gefundenen Bildkanten, die nicht durch geeignete Skalierung, Rotation und Translation von S_{PDM} eliminiert werden können, werden im folgenden Schritt durch eine Anpassung der Formparameter minimiert. Gesucht ist also eine Änderung dS_{PDM} der Modellkontur, so dass gilt:

$$T(s,\Theta,t_x,t_y)(S_{PDM}+dS_{PDM})=S_{edge} \tag{4.39}$$

Mit der inversen Transformation

$$T^{-1}(s,\Theta,t_x,t_y)=T(\frac{1}{s},-\Theta,-\frac{1}{s}(t_x\cos\Theta+t_y\sin\Theta),\frac{1}{s}(t_x\sin\Theta-t_y\cos\Theta)) \tag{4.40}$$

kann die nötige Formänderung berechnet werden:

$$dS_{PDM}=T^{-1}(s,\Theta,t_x,t_y)S_{edge}-S_{PDM} \tag{4.41}$$

Der Vektor dS_{PDM} gibt also an, wie die Punkte von S im lokalen Modellkoordinatensystem verschoben werden müssen, damit unter der Transformation T eine Übereinstimmung mit S_{edge} erreicht wird. Da diese Verschiebungen im Allgemeinen jedoch nicht konsistent mit den Beschränkungen für die Form des PDM sind, wird dS_{PDM} gemäß

$$dS_{PDM}\approx P\vec{db} \tag{4.42}$$

durch die Eigenvektoren des PDM approximiert.

Bei dem zur Minimierung verwendeten *least squares* Ansatz wird jeder Punkt mit einem Gewicht w_i gewichtet. Die Gewichte entsprechen gerade den bei der Kantensuche berechneten Werten für die Punktqualität (siehe Gleichung (4.31)).

Der Ansatz

$$(dS_{PDM}-P\vec{db})^T W(dS_{PDM}-P\vec{db}) \quad\Rightarrow\quad \text{min!} \tag{4.43}$$

bei der die Diagonalelemente der Gewichtsmatrix

$$W = \begin{pmatrix} w_1 & & & & & \\ & w_1 & & 0 & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & 0 & & \cdot & & \\ & & & & w_n & \\ & & & & & w_n \end{pmatrix} \tag{4.44}$$

mit den Punktqualitäten besetzt sind, führt durch Bilden der partiellen Ableitungen $\partial/\partial b_j$ schließlich auf den Ausdruck zur Berechnung der Anpassung der Formparameter:

$$d\vec{b} = (P^T W P)^{-1} P^T W dS_{PDM} \tag{4.45}$$

Der Vektor $d\vec{b}$ gibt also an, wie die Parameter der Kontur S_{ini} verändert werden müssen, so dass die Kantenstrukturen im Bild –beschrieben durch S_{edge}– in einer mit dem Modell konsistenten Weise bestmöglich approximiert werden. Bild 4.24 zeigt das Ergebnis einer solchen Approximation. Durch die Beschränkung des zulässigen Variationsbereiches der Parameterwerte wird der Effekt von Ausreißern im Kantenbild, wie z.B. in Bild 4.24 der Punkt unterhalb des linken Auges, abgeschwächt.

Bei der Modellsuche im Bild werden die Schritte *Kantensuche*, *Alignment* und *Anpassung der Formparameter* so lange wiederholt, bis die Kontur mit den Bildstrukturen hinreichend gut übereinstimmt oder eine Maximalzahl an Iterationen überschritten wird. Die Güte der Anpassung wird mit der im nächsten Abschnitt beschriebenen Qualitätsberechnung ermittelt.

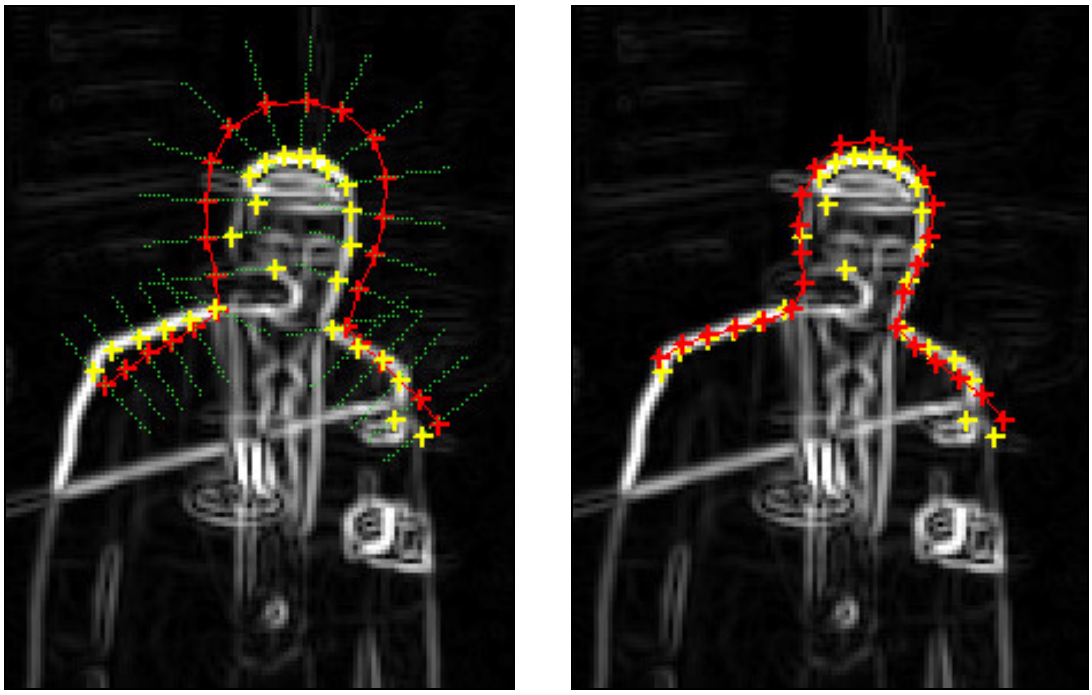


Bild 4.24: ***Iterationsschritt bei der Modellsuche****. Die Formparameter der initialen Silhouette (linkes Bild, rot) werden in jedem Iterationsschritt so verändert, dass die im Bild gefundenen Kantenpunkte (gelb) möglichst gut approximiert werden. Die Form der neuen Silhouette (rechtes Bild, rot) ist dabei konsistent mit dem Modell.*

4.5.10 Qualitätsberechnung

Um zu entscheiden, ob die iterative Suche mit einem weiteren Iterationsschritt fortgesetzt wird, oder ob die Suche beendet und eine Kontur $S = (x_1, y_1, ..., x_n, y_n)$ als gefundene Objektmodellinstanz akzeptiert werden kann, muss ein geeignetes Qualitätsmaß definiert werden, mit dem die Güte der Anpassung des Modells an die Bildstrukturen beurteilt werden kann. Als einfaches Maß für die Güte der Approximation wird in dieser Arbeit der durch

$$q(S) = \frac{1}{n \cdot g_{edge}^{\max}} \sum_{i=1}^{n} g(x_i, y_i) \tag{4.46}$$

definierte, auf den Wertebereich [0,1] normierte mittlere Grauwert im Kantenbild entlang des Konturverlaufs verwendet. Die Suche kann beendet werden, sobald $q(S)$, für eine Kontur einen zuvor definierten Schwellwert übersteigt. Der maximal mögliche Grauwert im Kantenbild $g_{edge}^{\max}$ beträgt in der Regel *255*.

5 Experimente

Im vorigen Kapitel wurden zwei Punktverteilungsmodelle für die menschliche Silhouette und das Gesicht vorgestellt. Im Rahmen der vorliegenden Arbeit wurde das System STABIL++ dahingehend erweitert, dass neben der bislang verwendeten Detektion mit dem Merkmal *Farbe* zusätzlich die Objektdetektion und –verfolgung mittels Kantenmerkmalen möglich wird. Dies gestattet zum einen eine robuste, von der Beleuchtung der Umgebung weitgehend unabhängige Segmentierung, darüber hinaus ist es damit möglich, die Auswertung auf im Vergleich zu 24-Bit Farbbildern weniger speicherintensiven 8-Bit Grauwertbildern vorzunehmen.

In diesem Kapitel werden im ersten Teil Experimente gezeigt, die demonstrieren, wie mit dem Silhouettenmodell Personen in Bildern detektiert und in Videobildfolgen verfolgt werden können. Innerhalb des Systems STABIL++ kann so auf die 3D Position der Person im Raum geschlossen werden. Hierbei werden die Ergebnisse aus der kantenbasierten Methode den Resultaten aus der bislang verwendeten Segmentierung mit dem Merkmal *Hautfarbe* gegenübergestellt und mit diesen verglichen.

Zur genauen Lokalisation des Gesichtes und im Hinblick auf eine zukünftige Erweiterung von STABIL++ um ein Modul zur Gesichtserkennung wird im zweiten Teil das in Abschnitt 4.4.7 vorgestellte Modell für die Lokalisation des Gesichtes eingesetzt. Anhand von Beispielen wird die Möglichkeit zur Gesichts*erkennung* mit diesem Ansatz aufgezeigt. Bei der Auswertung von Bildfolgen ergeben sich Trajektorien im Raum der Modellparameter. Es wird demonstriert, wie diese zur Charakterisierung einer bestimmten Mimik verwendet werden können.

5.1 Personendetektion

Für die Detektion von Personen wird das in Abschnitt 3.3.2 vorgestellte Menschmodell verwendet. Das Objektmodellteil (OMP) für den Kopf wird entweder durch ein Merkmal PDM (*feature PDM*) charakterisiert, wenn die Detektion mit dem Silhouettenmodell für den menschlichen Oberkörper erfolgt, oder aber durch das Merkmal Hautfarbe (*feature skin*), wenn im Bild hautfarbene Bereiche segmentiert werden.

Jedes OMP ist mit einem dreidimensionalen Suchraum ausgestattet. Im Interpretationsprozess wird nur von den Kameras ein Bild eingezogen, in denen die Suchbereiche sichtbar sind. Die Auswertung kann so auf diese relevanten Bilder beschränkt werden. Jedes Merkmal kapselt das Wissen darüber, wie seine Extraktion in den Bildern erfolgt. Für das Merkmal *PDM* wird mit dem in Abschnitt 4.5 beschriebenen Verfahren nach

Silhouetten im Bild gesucht, *feature skin* segmentiert zunächst mit der in Abschnitt 3.5.1 beschriebenen Farbklassifikation hautfarbene Bereiche im Bild und approximiert anschließend die gefundene Region durch eine Ellipse.

STABIL++ erlaubt sowohl die Auswertung von monokularen Bildfolgen als auch die Interpretation von Stereosequenzen, wobei die Anzahl der verwendeten Kameras prinzipiell nicht beschränkt ist. Für die Experimente in diesem Abschnitt wurden Aufnahmen verwendet, in denen die betrachtete Szene mit einer oder zwei Kameras aufgenommen wurde. Sowohl im Mono- als auch im Stereo-Fall kann für ein Objektmodellteil eine 3D-Position in Bezug auf das Weltkoordinatensystem (WKS) ermittelt werden. Der Übergang 2D-3D erfolgt dabei ausgehend von der Position des extrahierten Merkmals im Bild. Für das elliptische Farbmerkmal, das bei der Personendetektion mit *feature skin* verwendet wird, wird der Mittelpunkt der an die hautfarbenen Bereiche angepassten Ellipse bestimmt. Hieraus wird der zugehörige Sichtstrahl ermittelt, auf dem die 3D Position des Objektmodellteils im WKS liegt.

Wenn zur Charakterisierung des Objektmodellteils das Merkmal PDM verwendet wird, können ein oder mehrere Punkte der Silhouette als Referenzpunkte (*reference points*) verwendet werden, aus denen die Position des Merkmals im Bild bestimmt wird. Bei der Verwendung von einem Referenzpunkt wird mit den Parametern der Kamerakalibrierung der zugehörige Sichtstrahl unmittelbar berechnet. Bei mehreren Referenzpunkten wird hierzu zunächst der Schwerpunkt im Bild ermittelt. Bild 5.1 zeigt exemplarisch die Verwendung von einem, zwei oder drei Referenzpunkten.

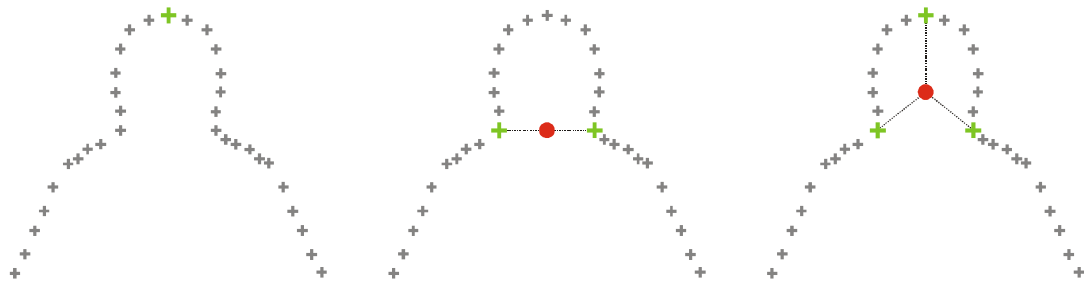


Bild 5.1: ***Referenzpunkte für den 3D-Übergang****. Zur Berechnung des Sichtstrahles können ein oder mehrere 2D Referenzpunkte (grün) verwendet werden. Bei mehr als einem Referenzpunkt wird zunächst der Schwerpunkt im Bild bestimmt (rot). Aus dem so erhaltenen Punkt wird der Sichtstrahl für die jeweilige Kamera ermittelt.*

Für monokulare Sequenzen kann die 3D-Position eines OMP und damit auch des Objektmodells unter Verwendung von Modellwissen und zusätzlichen Restriktionen geschätzt werden (siehe Abschnitt 5.1.2). Kann sich das OMP hingegen in mehreren Bildern extrahieren, ist eine exakte Stereo-Vermessung der Position möglich (siehe Abschnitt 5.1.3). Der 3D-Punkt, der sich aus den Referenzpunkten ergibt, bildet die Position des Merkmals im Weltkoordinatensystem. Diese kann gegenüber dem Ursprung des lokalen Koordinatensystems des OMP um einen Vektor $\vec{t}$ verschoben sein. Um die

Position des Objektmodellteils im Weltkoordinatensystem zu erhalten, muss der ermittelte Punkt daher entsprechend verschoben werden. Die Länge von $\vec{t}$ hängt dabei offensichtlich von den Positionen der verwendeten Referenzpunkte ab.

Bei der Anpassung der Silhouette an die Bildstrukturen wird insbesondere die Kopfpartie robust approximiert. Als Referenzpunkte eignen sich daher der oberste Kopfpunkt und die beiden Halspunkte. In einigen Fällen kann es allerdings trotz *blowing* (siehe Abschnitt 4.5.7) zu einer Anpassung der Silhouette an die Kante kommen, die sich aus dem Übergang Gesicht-Haare ergibt, und nicht –wie gewünscht– an die Kante, die den Übergang Haare-Hintergrund beschreibt. Die sich daraus ergebende Ungenauigkeit kann durch die Verwendung von linkem und rechtem Halspunkt als Referenzpunkte vermieden werden.

Im Folgenden wird zunächst der experimentelle Aufbau skizziert. Anschließend werden verschiedene Möglichkeiten der Detektion (monokulare Schätzung, Stereodetektion, Übergabe zwischen zwei Kameras) vorgestellt. Die Ergebnisse der Detektion mit dem Merkmal *PDM* werden dabei denen aus der Detektion mit dem Merkmal *Farbe* gegenübergestellt.

5.1.1 Experimenteller Aufbau

Bild 5.2 zeigt schematisch den experimentellen Aufbau zur Aufnahme und Auswertung von Sequenzen zur Personendetektion.

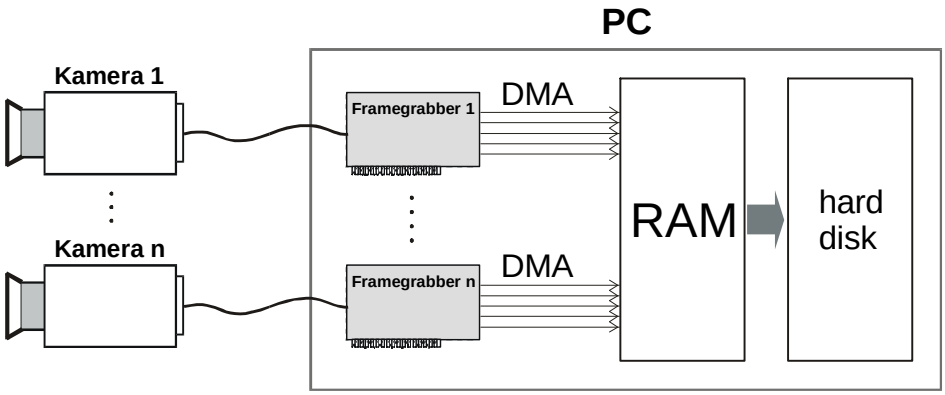


Bild 5.2: ***Experimenteller Aufbau zur Aufnahme von Sequenzen zur Personendetektion****. Die Bilder werden zunächst einzeln oder als ganze Sequenz über direct memory access (DMA) in den Hauptspeicher transferiert und können anschließend unmittelbar ausgewertet oder gegebenenfalls gespeichert werden.*

Der zu überwachende Bereich bzw. die betrachtete Szene wird im monokularen Fall mit einer, bei Stereosequenzen mit zwei oder mehr Kameras beobachtet. Das analoge Videosignal wird durch einen oder mehrere Framegrabber digitalisiert, und die Bilddaten werden mittels *direct memory access* (DMA) in den Hauptspeicher des Aufnahmerechners transferiert. Die Auswertung der Bilder kann entweder unmittelbar nach der Auf-

nahme der Einzelbilder erfolgen (*live* Modus), oder es besteht alternativ die Möglichkeit, eine Bildsequenz zunächst komplett im Hauptspeicher abzulegen und die Auswertung anschließend auf der kompletten Bildfolge durchzuführen. Für eine offline-Auswertung können die Einzelbilder der Sequenz darüber hinaus dauerhaft auf Festplatte gespeichert werden.

Die in dieser Arbeit gezeigten Sequenzen zur Personendetektion und –verfolgung wurden mit CCD Farbkameras vom Typ JAI-2040 (Sequenzen in Bild 5.4 und Bild 5.11) und JAI CV-S 3200 (Sequenz in Bild 5.8) aufgenommen. Die Digitalisierung erfolgte mit mehreren low cost 32-Bit PCI Framegrabbern vom Typ FALCON der Firma IDS, die in einem handelsüblichen PC mit Intel Pentium II Prozessor betrieben wurden. Die (offline-) Auswertung der Bildfolgen und die in den folgenden Abschnitten angegebenen Messungen erfolgten auf einem PC mit Intel Pentium III Prozessor mit einer Taktfrequenz von 800 MHz unter dem Betriebssystem Windows NT 4.0.

Bei Anwendungen im Bereich der Sicherheitstechnik werden Videobildfolgen häufig mit reduzierter Auflösung oder in einem komprimierten Format gespeichert. Auch die in den Testsequenzen verwendeten Bilder wurden mit einer in horizontaler und vertikaler Richtung halbierten Auflösung aufgenommen und haben das Format 384*288 Pixel. Für die Aufnahmen wurde das SVHS-Signal verwendet. Bei der gewählten Auflösung reicht es aus, von jedem Videoframe nur ein Halbbild (*field*) zu verwenden. Um Fehler durch einen Versatz zwischen den *fields odd* und *even* bei der Stereo-Zuordnung (vgl. Bild 5.7) zu vermeiden, wurde jeweils nur das erste oder zweite Halbbild digitalisiert. Hierdurch kommt es jedoch zu einer gewissen Zeitverzögerung zwischen den Aufnahmezeitpunkten der beiden Kamerabilder. Die Aufnahme eines korrespondierenden Bildpaares dauert im Mittel etwa 120 *ms*, da die Bilder der beiden Kameras sequentiell und nicht zeitlich parallel aufgenommen werden können und sich der Framegrabber bei jeder Aufnahme wieder neu auf das Videosignal synchronisieren muss. Bewegt sich eine Person beispielsweise mit einer Geschwindigkeit von 1 *m/s*, kommt es dadurch aufgrund der zeitlichen Differenz von 60 *ms* zu einem Positionsunterschied von 6 *cm* zwischen beiden Bildern. Der Effekt ließe sich durch die Verwendung von Framegrabbern reduzieren, die das synchrone Aufnehmen mehrerer Bilder erlauben. Eine weitere Verbesserung und damit eine Erhöhung der Genauigkeit ist zu erwarten, wenn die Videosignale beider Kameras synchronisiert werden, d.h. wenn eine Kamera als Master und alle übrigen im *slave*-Modus betrieben werden.

Kantenextraktion

Die in diesem Abschnitt gezeigten Bildsequenzen wurden unter relativ konstanten Umgebungsbedingungen im Innenbereich aufgenommen. Für die Auswertung stand ein Hintergrundbild zur Verfügung, das vom jeweils aktuellen Kamerabild subtrahiert wird und das vor der Aufnahme der eigentlichen Sequenz von der leeren Szene, d.h. ohne die Anwesenheit von Personen, aufgenommen wurde. Für die Kantenextraktion genügt da-

her die Anwendung eines einfachen Sobelfilters auf das mit einem Gaußfilter geglättete Differenzbild.

5.1.2 Monokulare Detektion

Für monokulare Videosequenzen kann ein 3D-Punkt für die Kopfposition geschätzt werden, wenn das Wissen aus Objektmodell genutzt wird und zusätzlich Annahmen über die mögliche Höhe des Kopfes gemacht werden. Bewegt sich eine Person aufrecht durch die beobachtete Szene, kann beispielsweise davon ausgegangen werden, dass sich die detektierte Kopfposition immer in einer konstanten Höhe über der *xy*-Ebene des Weltkoordinatensystems befindet. Aus der in der Szenenbeschreibung hinterlegten Modelldefinition kann die Höhe für das Merkmal PDM des Kopfes rekursiv aus dem Modellbaum berechnet werden.

Der Sichtstrahl, der von der detektierten Position des Merkmals im Bild ausgeht, kann im Weltkoordinatensystem mit der Kamerakonstanten *b* und den Parametern der externen Kalibrierung gemäß

$$\vec{S}_{WKS} = {}^{WKS}T_{cam}\vec{S}_{cam} = {}^{WKS}T_{cam}\begin{pmatrix} x_c \\ y_c \\ b \end{pmatrix} \tag{5.1}$$

aus den Koordinaten (x_c,y_c) im Kamerakoordinatensystem berechnet werden (vgl. auch Abschnitt 3.3.3).

Der Schnittpunkt dieses Sichtstrahls mit einer Ebene parallel zur *xy*-Ebene des Weltkoordinatensystems in Höhe der erwarteten Kopfposition liefert eine Schätzung für die 3D-Position $\vec{p}_{center}^{\,feature}$ des Merkmals in Weltkoordinaten (siehe Bild 5.3). Die Position des lokalen Koordinatensystems für das OMP Kopf ergibt sich schließlich aus der Verschiebung von $\vec{p}_{center}^{\,feature}$ um den im Objektmodell hinterlegten Translationsvektor $\vec{t} = (0,0,t)$.

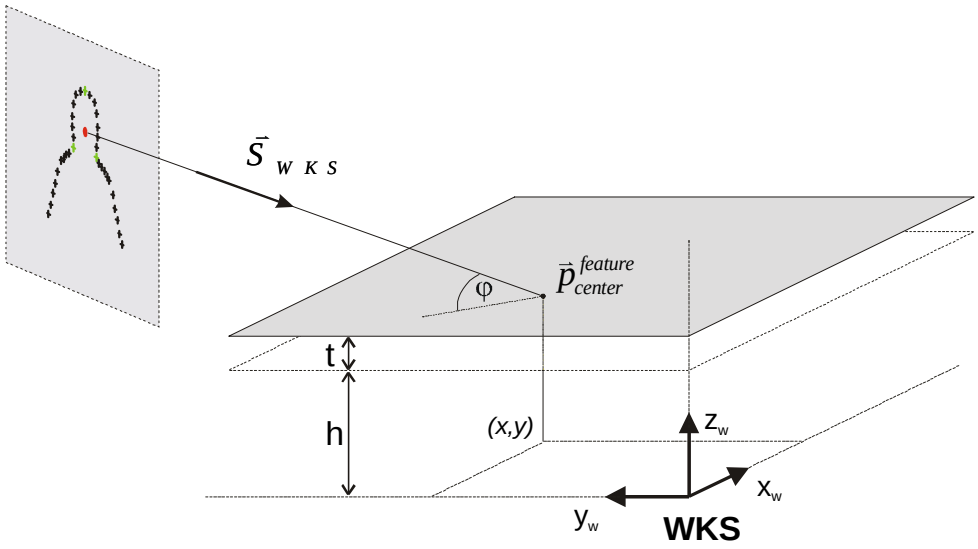


Bild 5.3: ***Monokulare Positionsbestimmung****. Der Sichtstrahl, der von der Position des extrahierten Merkmals im Bild ausgeht, liefert beim Schnitt mit einer Ebene in Höhe des Objektmodellteils den Ursprung des Merkmals (feature center). Dieser Punkt muss noch um die Translation t korrigiert werden, um so auf den Ursprung des lokalen Koordinatensystems des Objektmodellteils zu kommen.*

Die Genauigkeit der Mono-Schätzung hängt zum einen davon ab, wie genau die Annahme zutrifft, dass sich das betrachtete OMP in einer definierten Höhe befindet, zum anderen nimmt die Genauigkeit mit kleiner werdendem Neigungswinkel φ der Kamera ab. Eine Ungenauigkeit in der Angabe von $\vec{t}$ oder in der tatsächlichen Höhe des OMP wirkt sich auf die gemessene absolute Position aus. Sei Δh die Ungenauigkeit in der angenommenen Höhe und Δd die daraus resultierende Abweichung im Abstand d zur Kamera. Dann gilt:

$$\Delta d = \Delta h \frac{\cos\varphi}{\sin\varphi} \approx \Delta h \frac{1}{\varphi} \tag{5.2}$$

Die Approximation gilt für kleine Winkel φ, d.h. kleine Neigungswinkel der Kamera bzw. des Sichtstrahls gegenüber der *xy*-Ebene. Die Genauigkeit der gemessenen absoluten Position ist also proportional zur Ungenauigkeit in der Höhe des Objektmodellteils. Eine Ungenauigkeit wirkt sich umso stärker aus, je kleiner φ ist, bis bei $\varphi = 0$ schließlich keine Aussage über den Abstand des OMP zur Kamera mehr möglich ist und die Mono-Schätzung nicht mehr durchgeführt werden kann.

Bild 5.4 zeigt Bilder aus einer monokularen Sequenz, bei der die Objektdetektion links mit dem Merkmal *PDM* und rechts mit dem Merkmal *Hautfarbe* durchgeführt wurde.

Bild 5.4: ***Monokulare Detektion****. Links: Detektion mit flexiblem Konturmodell (PDM). Rechts: Detektion mit Merkmal "Hautfarbe" (skin). Die gefundene Silhouette bzw. der Bereich, der durch Farbklassifikation als ‚hautfarben' klassifiziert wurde, sind farbig markiert.*

Aus der Mono-Schätzung kann die 3D-Trajektorie für den Kopf ermittelt werden. Bild 5.5 zeigt den Verlauf der x- und y-Komponente der Trajektorie. Die z-Komponente hat bei der Schätzung immer denselben, konstanten Wert, der sich aus der Modellbeschreibung ergibt, und ist hier nicht visualisiert.

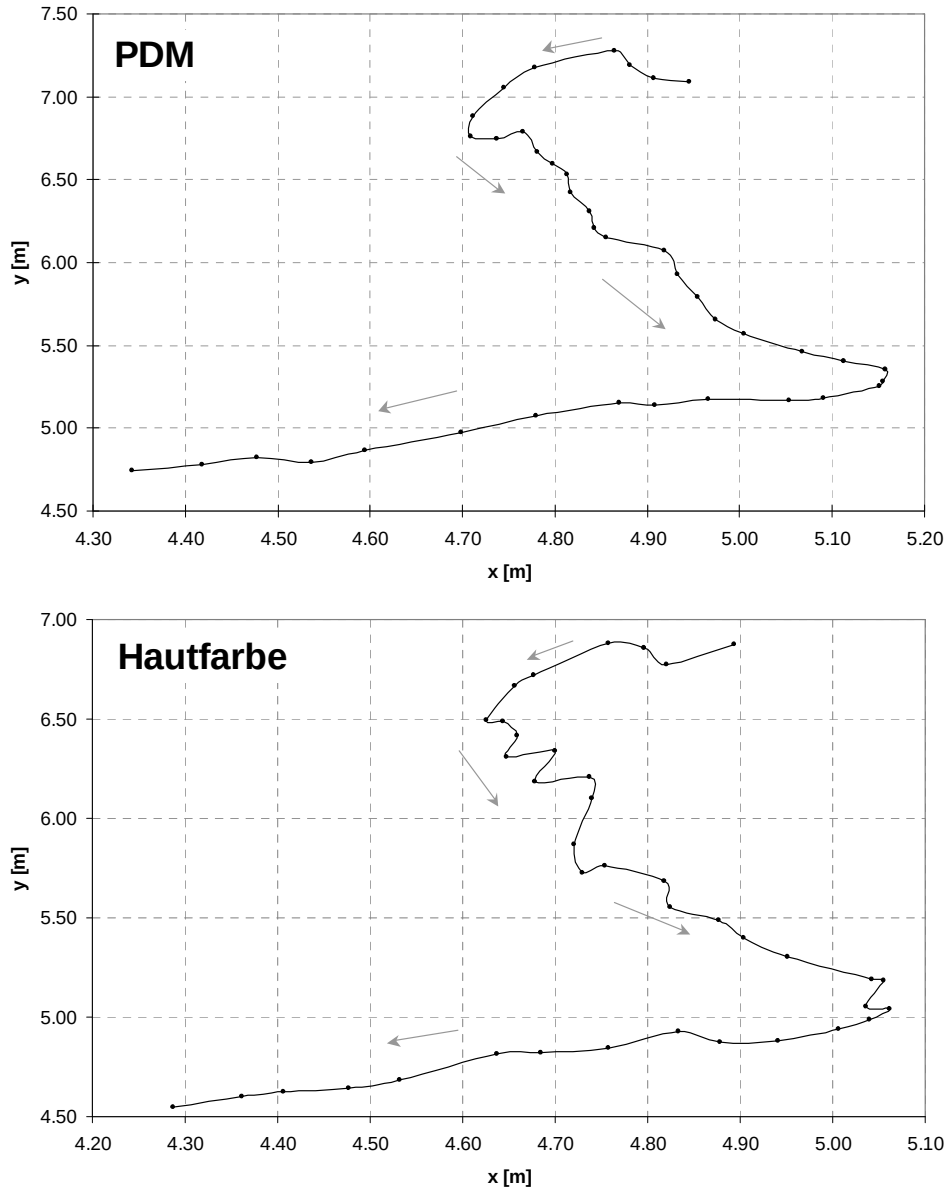


Bild 5.5: ***Trajektorie (x- und y-Komponente) der Kopfposition im zweidimensionalen Grundriss bei monokularer Detektion****. Oben: Auswertung mit Merkmal PDM. Unten: Auswertung mit Merkmal Hautfarbe.*

Der Trajektorienverlauf ist für beide Merkmale ähnlich, jedoch liefert die Messung mit dem Merkmal Hautfarbe systematisch kleinere absolute Werte für die *x*- und *y*-Komponente. Diese systematische Abweichung hat zwei Ursachen. Zum einen wirkt sich eine Ungenauigkeit in der Größe des Translationsvektors $\vec{t}$ nach Gleichung (5.2) auch in der gemessenen Absolutposition im Weltkoordinatensystem aus. Dieser Effekt ist umso größer, je kleiner φ ist. Dies ist insbesondere dann der Fall, wenn sich die detektierte Person wie hier weiter weg von der Kamera befindet. In Bild 5.5 zeigt sich dieser Effekt z.B. in der absoluten Differenz der *y*-Komponente. Für größere Abstände der Person von der Kamera nimmt auch die Differenz der *y*-Werte zu. Dies verdeutlicht

auch Bild 5.6, in dem die absolute Differenz der gemessenen *y*-Position gegen den mittleren Abstand aus beiden Messungen aufgetragen ist.

Ein weiterer systematische Fehler ergibt sich aus der Modellierung der Verschiebung von lokalem Koordinatensystem des OMP und dem Koordinatensystem für das Merkmal. Die gemessene Position beim Merkmal Hautfarbe liegt immer auf der Oberfläche des Kopfes der Person, müsste also um einen gewissen Betrag von der Kamera weg zur Kopfmitte hin verschoben werden. Die Korrektur mit $\vec{t} = (0,0,t)$ berücksichtigt jedoch nur eine Korrektur in *z*-Richtung, so dass sich bei der Schätzung für den Abstand systematisch zu kleine Werte ergeben. Das Merkmal PDM ist für die Mono-Schätzung also dem Merkmal Hautfarbe vorzuziehen, wenn absolute Messungen gemacht werden sollen.

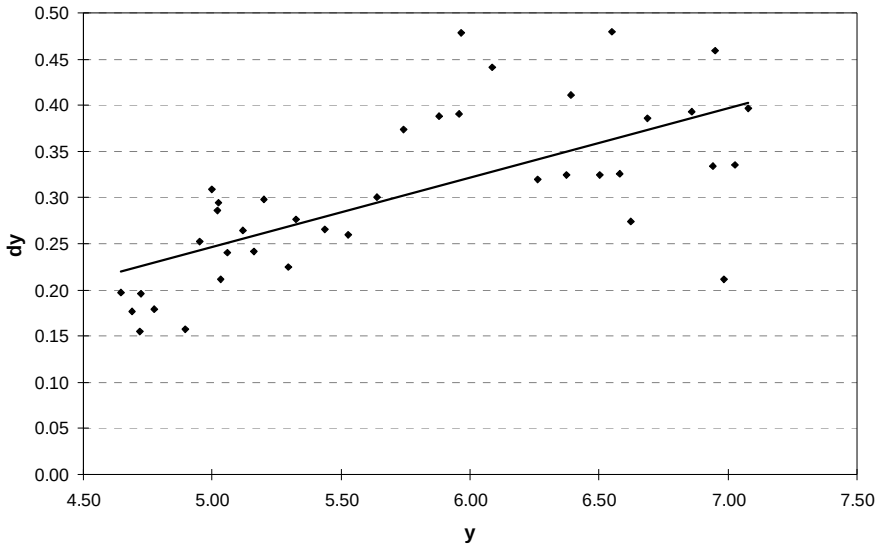


Bild 5.6: ***Positionsdifferenz in y-Richtung zwischen Detektion mit Merkmal PDM und Merkmal Farbe****. Ungenauigkeiten im Translationsvektor wirken sich mit zunehmender Entfernung und damit kleinerem Winkel φ stärker aus.*

Insbesondere bei der Trajektorie, die sich aus der Messung mit dem Merkmal *Hautfarbe* ergibt, fällt im oberen Teil ein zickzackförmiger Verlauf auf. Dieser ist unter anderem auf das Auf- und Abbewegen des Kopfes während des Gehens zurückzuführen. Hierdurch kommt es zu einer Oszillation der Höhe des Kopfes um die sich aus der Modellierung ergebende Soll-Höhe. Dies führt zu Messfehlern in der *x*- und *y*-Komponente. Die Oszillation der Kopfhöhe mit der Schrittfrequenz kann auch bei der Stereodetektion in Abschnitt 5.1.3 beobachtet werden, jedoch macht sie sich in diesem Fall als Oszillation in der (exakt vermessenen) *z*-Komponente bemerkbar, während die *xy*-Trajektorie einen wesentlich glatteren Verlauf hat (vgl. Bild 5.9).

5.1.3 Stereodetektion

Bei der im vorigen Abschnitt beschriebenen Auswertung von monokularen Bildfolgen kann unter Verwendung von Wissen über das 3D-Objektmodell die Position des lokalen Koordinatensystems für ein Objektmodellteil geschätzt werden. Durch die Annahme, dass sich das OMP in einer Ebene mit fest vorgegebener Höhe bewegt, ist dabei jedoch lediglich eine Aussage über die *x*- und *y*-Komponente der Position möglich. Die *z*-Komponente des geschätzten 3D-Punktes hat stets den konstanten Wert, der sich aus der Modellbeschreibung ergibt.

Eine unabhängige Bestimmung aller drei Koordinaten wird erst durch einen Stereo-Ansatz möglich, bei dem die Szene mit *n (n>1)* Kameras aufgenommen und die Bildauswertung simultan auf mehreren Bildern durchgeführt wird. Die gewonnenen dreidimensionalen Trajektorien beschreiben die tatsächliche Bewegung des OMP im Raum.

Bei der Stereo-Auswertung wird eine Segmentierung der Merkmale für alle *n* Kameras durchgeführt, in denen der Suchraum des betreffenden Objektmodellteils sichtbar ist, und von denen infolge dessen ein Bild eingezogen wurde. Für jedes Bild *i* wird der Sichtstrahl $\vec{S}_i$ des Schwerpunktes der Referenzpunkte berechnet und anschließend der Schnittpunkt aller Sichtstrahlen bestimmt. Bild 5.7 zeigt den binokularen Fall (*n=2*).

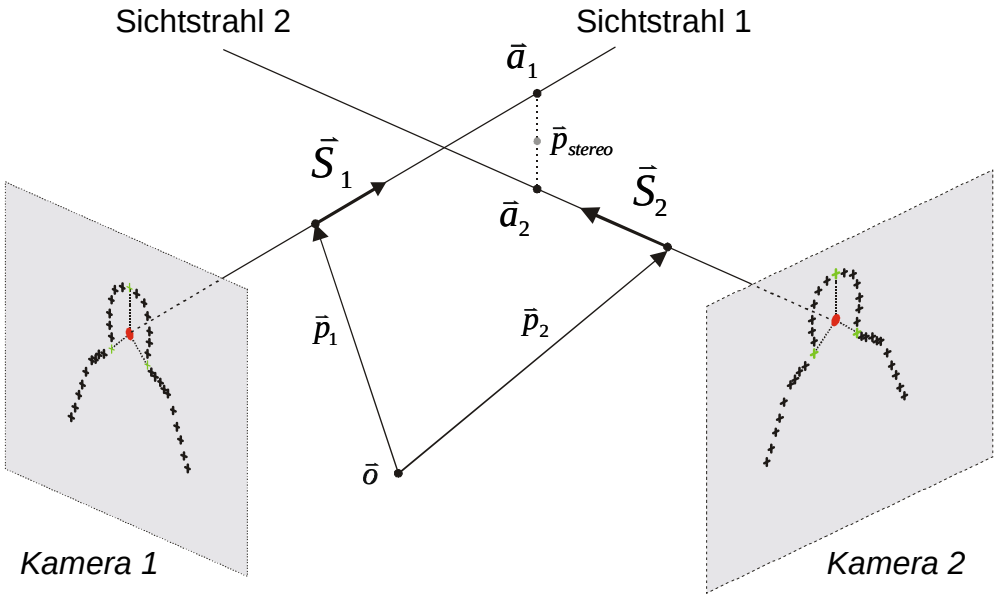


Bild 5.7: ***Binokulare Positionsbestimmung*** *mit n=2 Kameras zur Berechnung des 3D-Schnittpunktes aus den beiden Sichtstrahlen.*

Die Geradengleichungen der Sichtstrahlen sind durch Punkte $\vec{p}_i$ und Sichtstrahlvektoren $\vec{S}_i$ bestimmt, die sich aus den Kalibrierdaten der Kameras ergeben (siehe Bild 5.7). Die Minimierung von $\|(\vec{p}_1+\lambda\vec{S}_1)-(\vec{p}_2+\mu\vec{S}_2)\|$ liefert die beiden Geradenpunkte $\vec{a}_1$ und $\vec{a}_2$, an denen sich die Sichtstrahlen $\vec{S}_1$ und $\vec{S}_2$ am nächsten kommen. Bei Verwen-

dung von mehr als zwei Kameras werden die entsprechenden Punkte analog paarweise für die Sichtstrahlen $\vec{S}_i$, $\vec{S}_j$ mit $i \neq j$ berechnet. Der so definierte Schnittpunkt

$$\vec{p}_{stereo} = \frac{1}{2n} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \vec{a}_{ij} \tag{5.3}$$

mit $\vec{a}_{ij} = \vec{a}_i + \vec{a}_j$ beschreibt schließlich die 3D-Position des Merkmals im Weltkoordinatensystem. Zur Berechnung des Ursprungs des lokalen Koordinatensystems des OMP muss $\vec{p}_{stereo}$ noch um den Translationsvektor $\vec{t}$ verschoben werden, der den Versatz zwischen lokalem Koordinatensystem und *feature*-Ursprung beschreibt.

Bild 5.8 zeigt auf der linken Seite beispielhaft einige Bilder aus einer Sequenz, in der die für Sicherheitsanwendungen typische Überwachung eines Raumes durch zwei Kameras zu sehen ist. Die in den Raum eindringende Person wird mit dem Merkmal PDM detektiert und anschließend über die gesamte Bildfolge durch Stereodetektion verfolgt. Das Ergebnis der Messung und die Modellierung der Umgebung werden auf der rechten Seite dreidimensional in einem OpenGL-Monitor dargestellt und darüber hinaus in einem zweidimensionalen Grundriss des überwachten Raumes visualisiert. Die Modellierung der Umgebung wurde aus Gründen der Übersichtlichkeit auf die für den Interpretationsprozess relevanten Wände und Inventargegenstände beschränkt.

Um fehlerhafte Korrespondenzen zu vermeiden, kann in STABIL++ der maximal zulässige Abstand der Sichtstrahlen beschränkt werden. Für den Fall, dass der tatsächlich gemessene Abstand größer als der zulässige Maximalwert ist, kann eine Mono-Schätzung durchgeführt werden. Die Beschränkung wurde für die gezeigte Sequenz aufgehoben, um so eine durchgehende Stereo-Messung zu ermöglichen und die *z*-Komponente der Trajektorie für den Kopf zu bestimmen. Die Genauigkeit hängt dabei von mehreren Faktoren ab. Zum einen besteht zwischen den Aufnahmezeitpunkten der beiden korrespondierenden Kamerabilder der gezeigten Sequenz eine zeitliche Differenz von ca. 60 *ms* (siehe Abschnitt 5.1.1). Dadurch unterscheiden sich die Positionen der Person im rechten und linken Bild leicht voneinander. Darüber hinaus kann es bei der Anpassung des Silhouettenmodells an die Bildstrukturen zu Ungenauigkeiten kommen. Schließlich trägt auch die Güte der Kamerakalibrierung zur Genauigkeit bei.

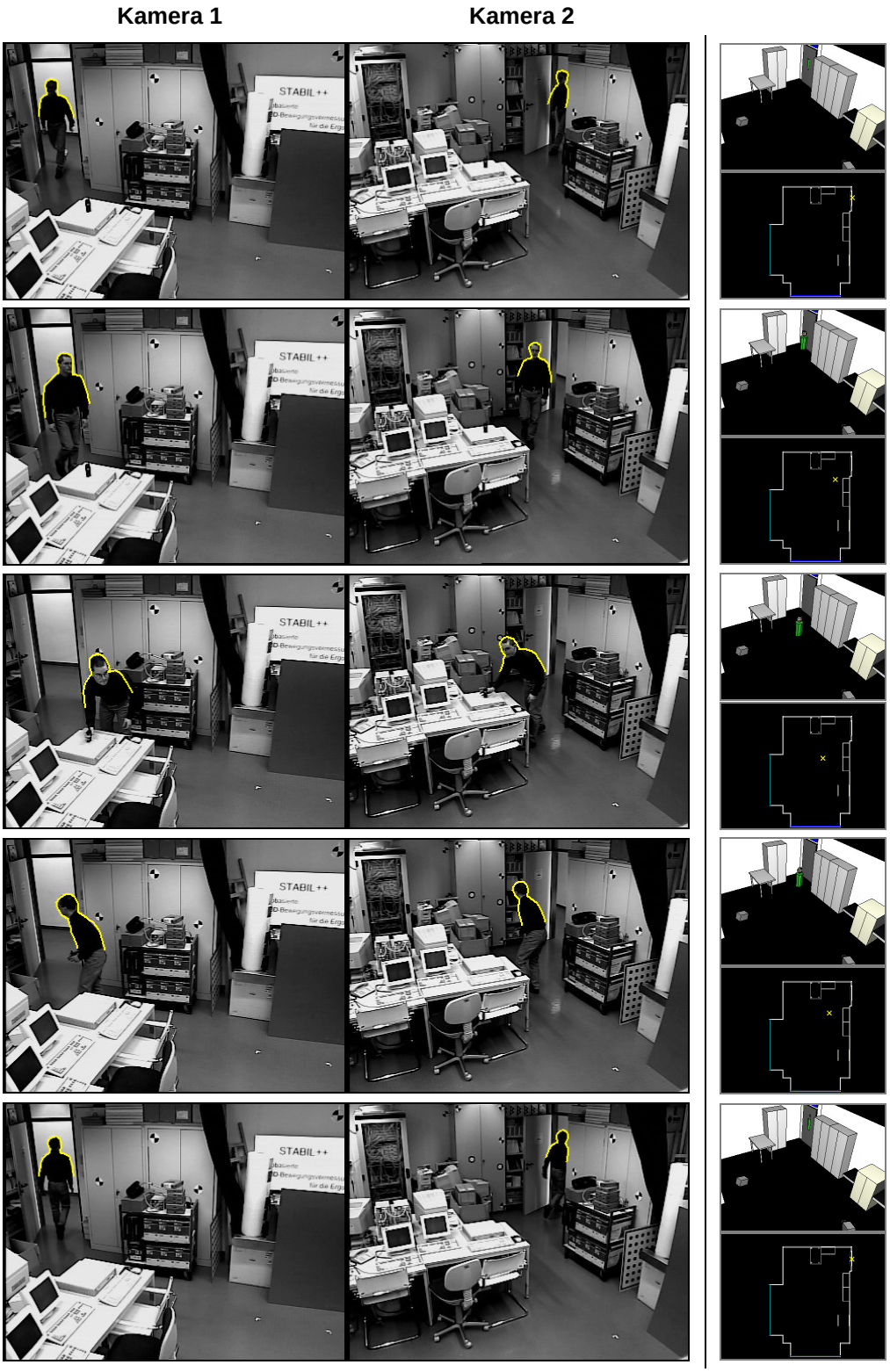


Bild 5.8: ***Stereodetektion****. Links: Überwachung eines Raumes mit zwei Kameras. Die Person wird mit Hilfe des PDM-Modells für die menschliche Silhouette detektiert und über die Sequenz verfolgt. Die gefundene Silhouette ist gelb eingezeichnet. Rechts: 3D-Visualisierung der Umgebung in einem OpenGL-Monitor und Angabe der Position in einem 2D-Grundriss.*

Bild 5.9 und Bild 5.10 zeigen den Verlauf der Trajektorie für den oberste Kopfpunkt. Durch die Verwendung mehrerer Kameras kann eine unabhängige Bestimmung der drei Koordinaten erreicht werden, auf Modellwissen über das 3D-Objektmodell muss hierzu nicht zurückgegriffen werden. Im *xy*-Diagramm in Bild 5.9 fällt auf, dass der Verlauf wesentlich glatter ist als bei der Mono-Schätzung (siehe Bild 5.5). Die Auf- und Abbewegung des Kopfes beim Gehen macht sich nicht mehr als Messfehler bemerkbar, sondern wird in der *z*-Komponente exakt erfasst.

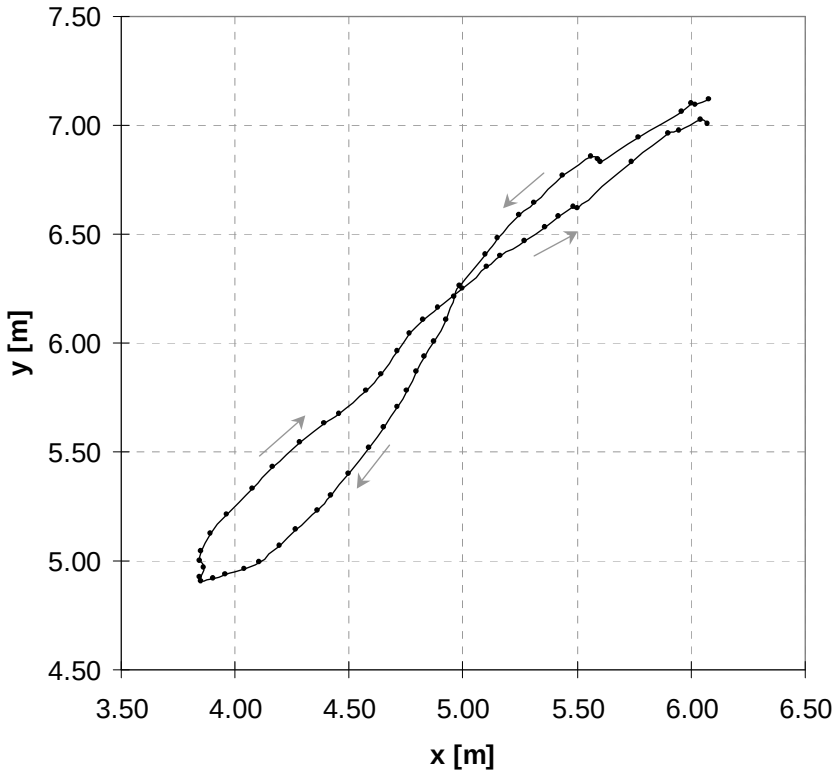


Bild 5.9: ***Trajektorie (x- und y-Komponente) der Kopfposition im zweidimensionalen Grundriss bei Stereodetektion****. Durch die Verwendung mehrerer Sichtstrahlen wird eine unabhängige Bestimmung der drei Koordinaten erreicht. Der Verlauf der Trajektorie wird dadurch wesentlich glatter, da Ungenauigkeiten durch Auf- und Abbewegung des Kopfes vermieden werden.*

Die in Bild 5.10 dargestellte *z*-Komponente der Trajektorie zeigt zum einen das Absenken des Kopfes beim Ergreifen des Gegenstandes, außerdem ist der Bewegung die Auf- und Abwärtsbewegung des Kopfes beim Gehen mit der Schrittfrequenz überlagert. Die zeitliche Differenz zwischen der Aufnahme zweier aufeinanderfolgender Bilder/Frames betrug etwa 120 ms, was einer Framerate von ca. 8-9 Bildern pro Sekunde entspricht.

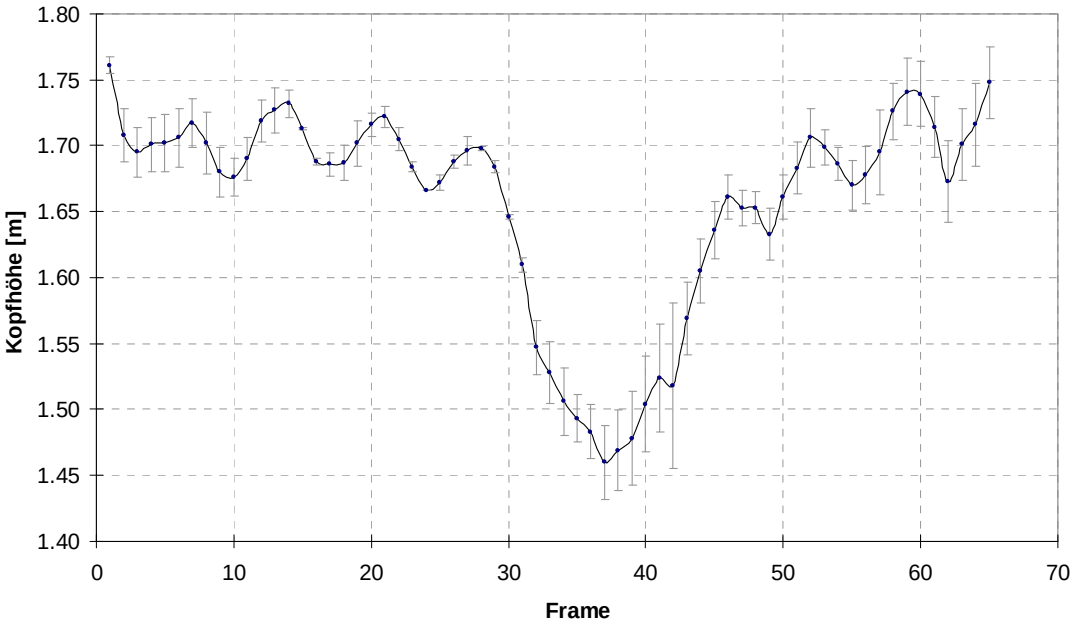


Bild 5.10: ***z-Komponente der Kopfposition bei Stereodetektion****. Der im mittleren Bild in Bild 5.8 zu sehenden Abwärtsbewegung des Kopfes ist dessen Auf- und Abbewegen beim Gehen mit der Schrittfrequenz überlagert. Die Fehlerbalken geben den Abstand der Sichtstrahlen bei der Stereo-Zuordnung an.*

5.1.4 Übergabe zwischen zwei Kameras

In modernen Videomanagementsystemen werden mit einer Vielzahl von Kameras häufig nicht nur einzelne Räume, sondern ganze Gebäudekomplexe beobachtet. Bei Verwendung einer entsprechend großen Anzahl von Kameras kann eine lückenlose Überwachung erreicht werden, und es wird möglich, eine Person über mehrere Räume hinweg zu verfolgen. Der zu überwachende Bereich, beispielsweise mehrere getrennte Räume mit den dazwischenliegenden Fluren, muss dabei von den Sichtbereichen der verschiedenen Kameras erfasst werden. Die Sichtbereiche müssen sich daher teilweise überlappen oder zumindest unmittelbar aneinander grenzen. So wird erreicht, dass eine einmal detektierte Objektmodellinstanz von einer Kamera zur nächsten übergeben werden kann.

Bild 5.11 zeigt beispielhaft die Überwachung der linken und rechten Raumhälfte durch jeweils eine Kamera, bei der sich im Übergangsbereich die Sichtbereiche der Kameras teilweise überlappen. Durch die Kombination von monokularer Schätzung (siehe Abschnitt 5.1.2) mit der im vorhergehenden Abschnitt dargestellten Stereodetektion wird es möglich, die in den Sichtbereich eintretende Person zu detektieren und lückenlos zu verfolgen.

Bild 5.11: ***Übergabe zwischen zwei Kameras mit Merkmal PDM****. Oben: Mono-Schätzung im linken Bild. Mitte: binokulare Positionsbestimmung im gemeinsamen Sichtbereich der beiden Kameras. Unten: monokulare Schätzung im rechten Bild.*

In der linken Raumhälfte ist die Person zunächst nur in einer Kamera sichtbar. Durch Verwendung von Modellwissen wird eine monokulare Schätzung durchgeführt, um von der im Bild detektierten Kopfposition auf einen 3D-Punkt in Weltkoordinaten zu schließen. Sobald sich die Person im Schnittbereich der Sichtbereiche beider Kameras befindet und die Merkmalsextraktion in beiden Kamerabildern durchgeführt werden kann, wird eine binokulare Positionsbestimmung möglich. Die durch Stereodetektion gewonnenen Punkte für die Position des Kopfes sind in dem in Bild 5.12 gezeigten *xy*-Diagramm durch offene Kreise dargestellt, die Beschriftung der einzelnen Datenpunkte gibt bei Stereo-Messung den Abstand der Sichtstrahlen in Zentimetern an.

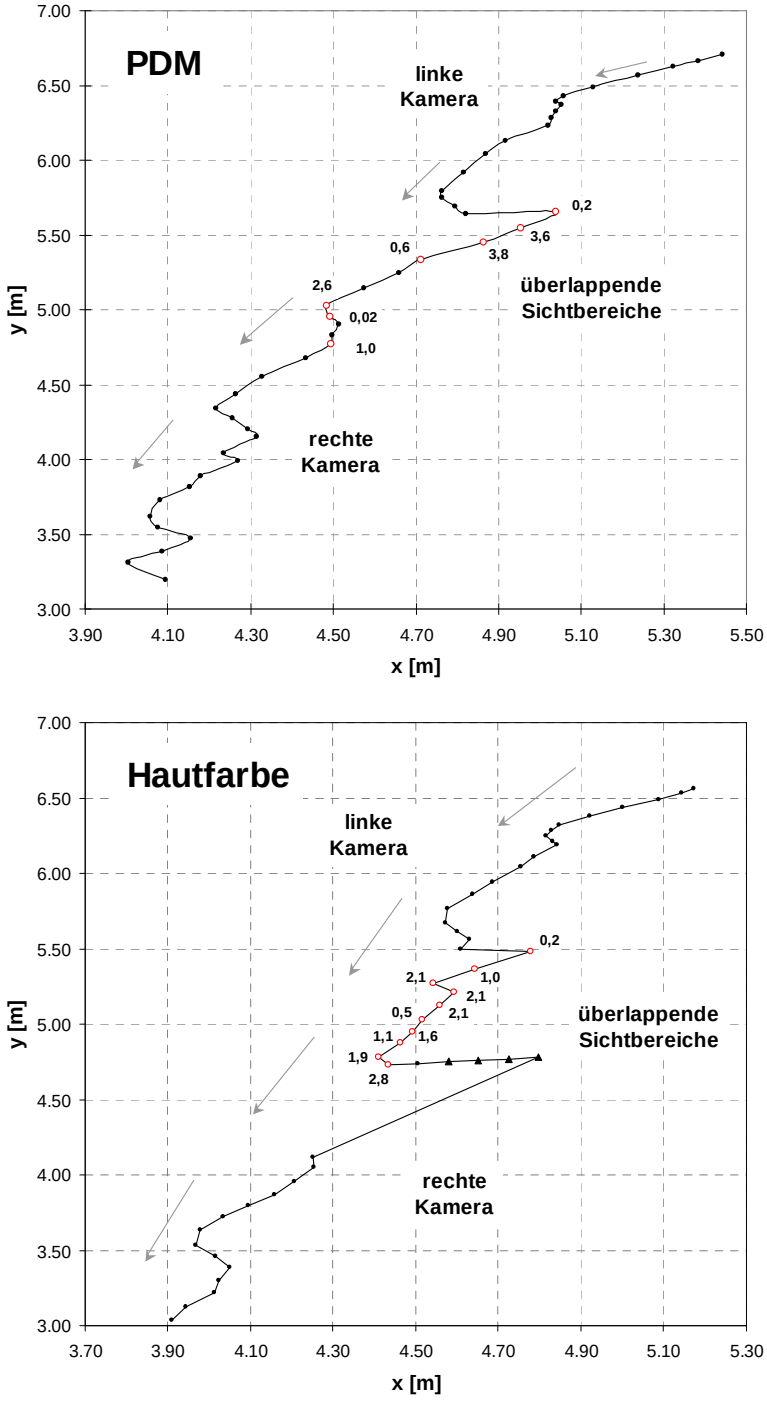


Bild 5.12: ***Trajektorie (x- und y-Komponente) der Kopfposition im zweidimensionalen Grundriss bei der Übergabe zwischen zwei Kameras****. Im Überlappungsbereich des Sichtbereiches beider Kameras erfolgt eine binokulare Positionsbestimmung (rote Kreise). Bei der Auswertung von nur einem Bild wird eine Mono-Schätzung (schwarze Punkte) durchgeführt. Die Datenpunkte bei der Stereodetektion sind mit dem Abstand der Sichtstrahlen (Angabe in [cm]) beschriftet. Die mit Dreiecken markierten Punkten wurde prädiziert.*

Im Bereich der Mono-Schätzung fällt wieder der zickzackförmige Verlauf der Trajektorie auf, der durch die Oszillation der tatsächlichen Kopfposition um die Soll-Position

verursacht wird, die sich aus der Modellbeschreibung ergibt. Weiterhin sind auch hier, genau wie bei der monokularen Detektion in Abschnitt 5.1.2, die Werte für das Merkmal *Hautfarbe* im Vergleich zum Merkmal *PDM* systematisch kleiner (vgl. auch Bild 5.6). Für die Positionsvorhersage für das OMP *Kopf* wurde ein einfacher Filter verwendet, der die Position des Kopfes aus den beiden letzten *history*-Werten linear prädiziert. Für den Fall, dass im Interpretationsprozess keine gültige Modellhypothese erstellt werden kann, wird die Position der Person geschätzt. Die geschätzten Werte sind in Bild 5.12 als Dreiecke dargestellt.

Die beiden Kamerabilder wurden wie in Abschnitt 5.1.1 beschrieben sequentiell aufgenommen, wodurch sich eine zeitliche Differenz der Aufnahmezeitpunkte von etwa 60 ms ergibt. Dies zusammen mit Ungenauigkeiten der Anpassung des Modells an die Bildstrukturen und Ungenauigkeiten bei der Kamerakalibrierung führt zu einer Messungenauigkeit von maximal einigen wenigen Zentimetern.

5.2 Gesichtsdetektion

Die Lokalisation des menschlichen Gesichtes in Einzelbildern und dessen Verfolgung in Videobildfolgen bilden die Grundlage für eine Vielzahl von Anwendungen. Bei einer Schnittstelle zur Mensch-Maschine Kommunikation ist beispielsweise die Steuerung der Maschine über Bewegungen der Arme, des Kopfes oder der Augen denkbar. Aber auch in anderen Anwendungen, wie z.B. bei Zutrittskontrollsystemen, ist eine Lokalisation der Person und das Auffinden des Gesichtes im Bild notwendig, um beispielsweise eine Identifikation der Person zu ermöglichen.

In diesem Abschnitt wird das in Abschnitt 4.4.7 vorgestellte Gesichtsmodell zusammen mit der in Abschnitt 4.5 erläuterten PDM-Suche für die Auswertung von Einzelbildern und Bildfolgen eingesetzt. Die verwendeten Bilder wurden mit einer CCD-Kamera vom Typ *CV-S 3200* der Firma *JAI* aufgenommen, deren Videosignal über einen low cost 32-Bit Framegrabber *FALCON* der Firma *IDS* digitalisiert wurde. Um Interlace-Effekte durch Bewegungen zu vermeiden, wurde jeweils nur ein Halbbild (*field*) gegrabbt. Die Auflösung in *x*-Richtung wurde entsprechend halbiert, so dass die Bilder das Format 288*384 Pixel haben.

5.2.1 Kantenextraktion

Bei der Personenverfolgung mit dem Konturmodell für den menschlichen Oberkörper genügte in Abschnitt 5.1 nach der Subtraktion eines Hintergrundbildes die Anwendung des einfachen Sobelfilters zur Kantenextraktion. Bei der Gesichtserkennung wird kein Hintergrundbild verwendet, sondern die Kantenextraktion wird unmittelbar auf den mit einem Gaußfilter geglätteten Originalbildern durchgeführt. Rauschen in den Grauwerten

der Bilder und nicht modellierte Objektdetails, wie z.B. Hautunreinheiten oder Falten, führen bei alleiniger Anwendung des Sobelfilters auf das Originalbild zu verrauschten Kantenbildern. Daher ist insbesondere bei verrauschtem Kamerasignal eine aufwändigere Bildvorverarbeitung notwendig.

Die für die PDM-Suche implementierten Klassen und Algorithmen erlauben die Verwendung verschiedener Methoden zur Kantenextraktion. Gute Resultate ergeben sich durch zusätzliche Anwendung eines Hysterese-Schwellwertes auf das mit einem Sobelfilter erzeugte Kantenbild (siehe Bild 5.13). Hierbei werden alle Pixel *p* des Sobelbildes, deren Grauwert $g_{edge}(p)$ über einem Grenzwert max_{hyst} liegt, als Kantenpunkt akzeptiert (*secure points*). Alle Pixel, deren Grauwert unter einer minimalen Schwelle min_{hyst} liegt, werden zurückgewiesen. Pixel, deren Grauwert zwischen min_{hyst} und max_{hyst} liegt (*potential points*), werden nur dann als Kantenpunkte akzeptiert, wenn sie über einen Pfad von potentiellen Punkten mit einer maximalen Länge von $length_{hyst}$ mit einem sicheren Punkt verbunden sind. Für die Klassifizierung eines Pixels p im Kantenbild als Kantenpunkt gilt also:

$$(p \text{ ist Kantenpunkt}) \Leftrightarrow (g_{edge}(p) >= max_{hyst}) \text{ ODER } (p \text{ ist über einen Pfad von potentiellen Punkten mit } length(path) <= length_{hyst} \text{ mit einem sicheren Punkt verbunden}); \text{ für die potentiellen Punkte gilt: } g_{edge}(p) >= min_{hyst}$$

In dieser Arbeit wurden für min_{hyst} und max_{hyst} die Werte 30 und 60 gewählt.

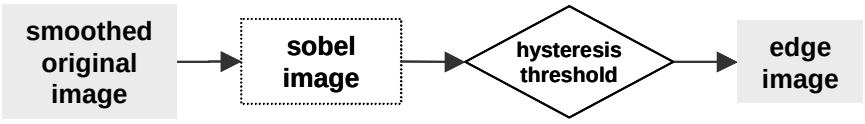


Bild 5.13: ***Kantenextraktion für die Gesichtserkennung bei verrauschtem Kamerasignal****.*

5.2.2 Gesichtslokalisation im Bild

Bild 5.14 zeigt beispielhaft die Anpassung des Gesichtsmodells an die Bildstrukturen bei der iterativen PDM-Suche.

Die initiale Gesichtsform entspricht der mittleren Modellkontur $\overline{S}$. Obwohl Größe und Form bei der initialen Positionierung nur schlecht mit dem tatsächlichen Gesicht übereinstimmen, wird bei der PDM-Suche schon nach wenigen Iterationen eine gute Grobanpassung des Modells an die Bildstrukturen erreicht. Nach 10 Iterationen befinden sich sämtliche Konturteile in etwa an der richtigen Position, lediglich die Feinstrukturen wie

beispielsweise der genaue Verlauf der Lippen oder der Augenpartie stimmen noch nicht ganz mit dem tatsächlichen Verlauf überein. In den folgenden Iterationen erfolgt die Feinanpassung, und nach etwa 30 Iterationen ist ein stationärer Zustand erreicht, bei dem die Suche abgebrochen werden kann. Zusätzliche Iterationen bringen keine weitere Verbesserung der Anpassung mehr.

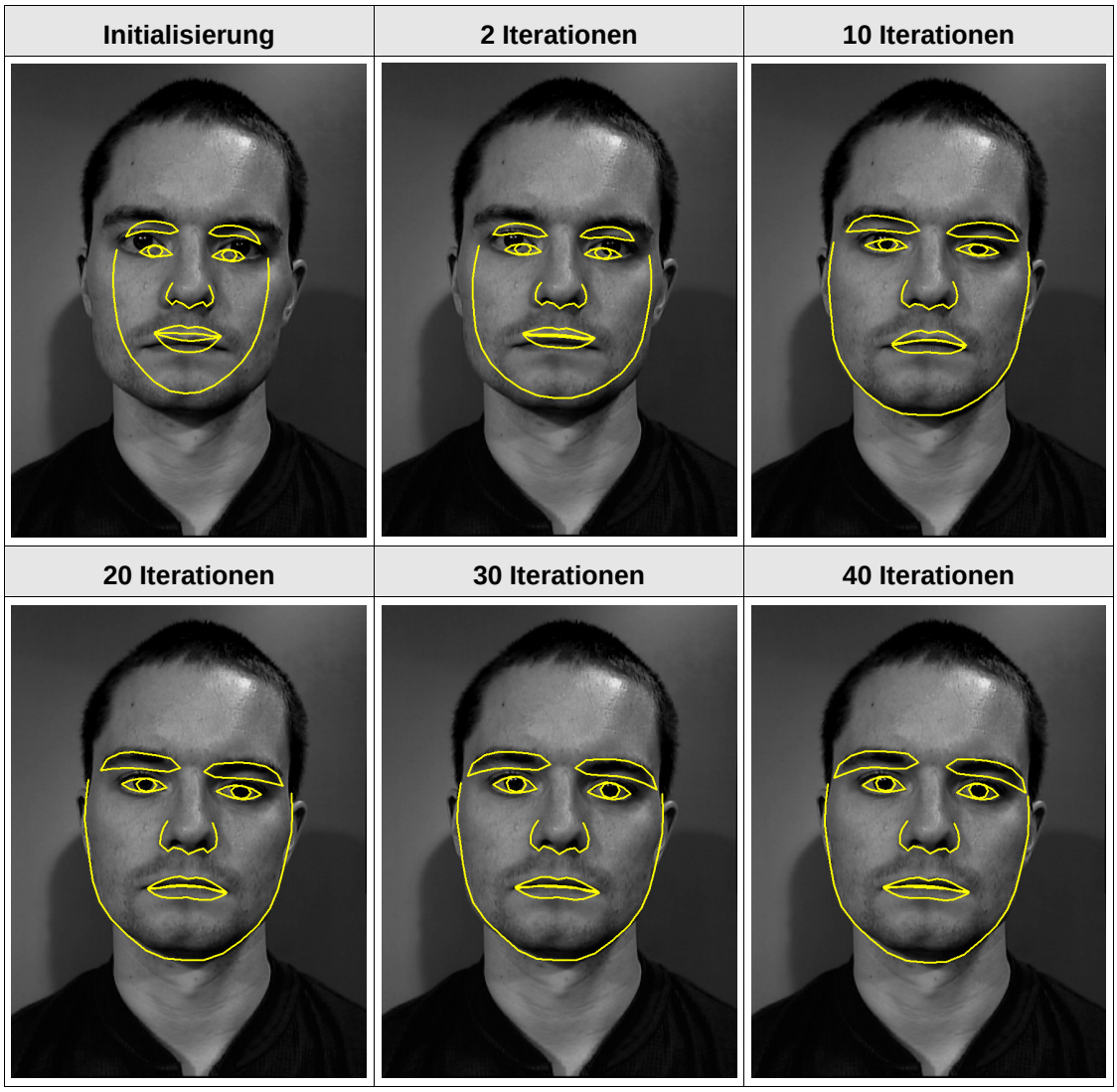


Bild 5.14: ***Anpassung des Gesichtsmodells an die Bildstrukturen.***

5.2.3 Berücksichtigung von Grauwertinformationen

Die Augenbrauen oder auch die Lippen sind im Kantenbild oft nicht scharf begrenzt, sondern bilden einen mehr oder weniger diffusen Bereich. Bei der Suche in Bild 5.14 führt dies dazu, dass die Augenbrauen außen nach links bzw. rechts zur Schläfe hingezogen werden, da diese mit dem Haaransatz eine klar definierte Kante ausbildet. Die

alleinige Berücksichtigung der Kantenstärke bei der Suche senkrecht zum Konturverlauf führt weiterhin dazu, dass einzelne Konturteile manchmal nicht exakt lokalisiert werden und die Modellkanten an falsche Bildkanten approximiert werden.

Um die Suche robuster zu machen und um eine genauere Anpassung des Modells an die Bildstrukturen zu erreichen, ist die zusätzliche Berücksichtigung von Grauwertinformationen sinnvoll. Denkbar wäre beispielsweise, bei der Modellerstellung für jeden Modellpunkt einen mittleren Grauwertverlauf entlang der Senkrechten zum Konturverlauf zu lernen. Bei der Suche im Bild könnte dann dasjenige Pixel als neue Kante ermittelt werden, bei dem die Korrelation zwischen mittlerem Grauwertprofil des Modells und dem tatsächlichen Verlauf maximal wird. Das Aussehen und damit auch der Verlauf der Grauwertprofile variiert jedoch von Person zu Person und auch bei verschiedenen Beleuchtungsbedingungen stark. Dies legt nahe, statt der rechenintensiven Korrelationsberechnung eine einfachere Heuristik zu verwenden: Für jeden Modellpunkt wird der qualitative Verlauf des Grauwertprofils definieret, und das Punktverteilungsmodell wird um dieses zusätzliche a priori Wissen erweitert.

Bei den meisten Europäern sind beispielsweise die Augenbrauen wesentlich dunkler als die umgebende Hautpartie und die Pupille erscheint im Vergleich zum Augapfel als schwarzer Kreis. Entlang der Suchrichtung senkrecht zum Konturverlauf hat der Gradient des Grauwertverlaufs am Kantenpunkt daher ein definiertes Vorzeichen.

Das PDM wird um diese Informationen erweitert, indem für jeden Modellpunkt ein Sollwert für das Vorzeichen des Gradienten festgelegt wird. Bei der Kantensuche wird für jeden Punkt p auf der Suchgeraden nach Gleichung (4.31) die Güte der Kante bewertet. Bei der Qualitätsberechnung wird das Vorzeichen des Gradienten gemäß

$$q_{edge}(p) = \begin{cases} q_{edge}(p) + sf * (1 - q_{edge}(p)) & , \quad \mathrm{sgn}_{\mathrm{mod}} = \mathrm{sgn}_{\mathrm{img}}(p) \\ q_{edge}(p) & , \quad \mathrm{sgn}_{\mathrm{mod}} = 0 \\ q_{edge}(p) * (1 - sf) & , \quad \mathrm{sgn}_{\mathrm{mod}} \neq \mathrm{sgn}_{\mathrm{img}}(p) \end{cases} \tag{5.4}$$

mit $0 \leq sf \leq 1$ berücksichtigt. Der Faktor sf (*sign factor*) gibt die Größe der Qualitätserhöhung bzw. -erniedrigung an, sgn_{mod} ist der im Modell hinterlegte Sollwert für das Vorzeichen des Gradienten, und $sgn_{img}(p)$ bezeichnet das Vorzeichen des Gradienten im Bild entlang der Suchrichtung. Die Werte sgn_{mod} und $sgn_{img}(p)$ für das Vorzeichen des Gradienten können die Werte –1 (Grauwert nimmt in Suchrichtung ab, d.h. Übergang hell-dunkel an der Kante), +1 (Grauwert nimmt in Suchrichtung zu, d.h. Übergang dunkel-hell an der Kante) und 0 (Gradient wird nicht berücksichtigt) annehmen.

Durch Gleichung (5.4) wird erreicht, dass die Güte einer Kante höher bewertet wird, wenn das aus dem Bild ermittelte Vorzeichen des Gradienten mit dem Sollwert aus dem Modell übereinstimmt. Bei entgegengesetzten Vorzeichen wird die Qualität entspre-

chend reduziert. Ein Punkt, der vom Ausgangspunkt der Suche weiter entfernt ist, aber das korrekte Vorzeichen für den Gradienten hat, kann somit eine bessere Qualität erreichen als ein Punkt, der zwar näher am Ausgangspunkt liegt, aber bei dem der Gradient nicht mit dem Modell übereinstimmt.

Bild 5.15 zeigt links am Beispiel der rechten Augenbraue das Ergebnis der PDM-Suche, wenn ausschließlich Kantenstärke und Abstand zum Ausgangspunkt für die Qualitätsberechnung verwendet werden. Durch die zusätzliche Berücksichtigung des Grauwertverlaufes an der Kante wird der tatsächliche Verlauf wesentlich besser approximiert, und die Außenkante wird nicht mehr nach rechts gezogen (siehe Bild 5.15 rechts). Die Kante, die sich aus dem Übergang vom Haaransatz zur Schläfe ergibt, ist zwar deutlich ausgeprägt, hat jedoch im Vergleich zum Übergang Schläfe-Augenbraue das falsche Vorzeichen. Der *sign factor* betrug in diesem Beispiel *sf=0.4*.

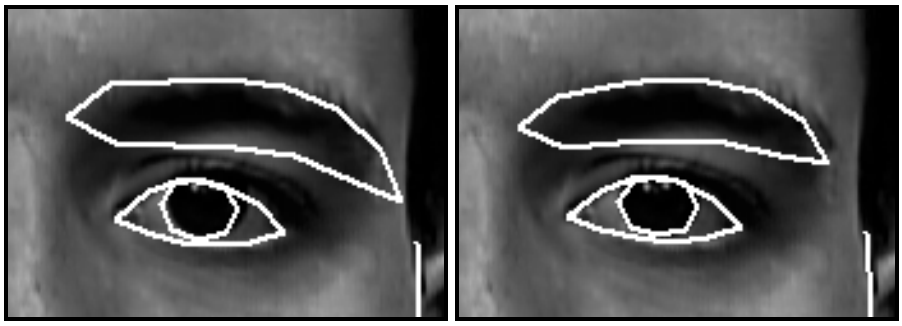


Bild 5.15: ***Verbesserung der Anpassung an die Bildstrukturen durch Verwendung von Grauwertinformationen****. Links: fehlerhafte Anpassung ohne zusätzliche Grauwertinformationen. Rechts: zusätzliche Berücksichtigung des Vorzeichens für den Gradienten in Suchrichtung bei der Kantensuche.*

5.2.4 Parametervektor für verschiedene Personen

Die mittlere Kontur $\overline{S}$ des PDM-Modells (siehe Bild 4.13) beschreibt das mittlere Gesicht, das sich aus dem Trainingsdatensatz ergibt und durch den Parametervektor $\vec{b} = \vec{o}$ charakterisiert ist. Die Gesichtsform realer Personen weicht von dieser Form stets ab, d.h. selbst für den neutralen Gesichtsausdruck werden sich Parameterwerte b_i mit $b_i \neq 0$ ergeben. Tabelle 5.1 zeigt die Größe der Parameterwerte bei neutralem Ausdruck für vier verschiedene Testpersonen.

Tabelle 5.1: ***Parameterwerte b_i bei neutralem Gesichtsausdruck für 4 verschiedene Testpersonen****.*

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}
0.06	0.37	0.76	0.81	0.54	-0.59	0.60	-1.34	-0.47	-1.00	0.00	0.04	-0.99	0.99	1.00	-0.85	-1.51
0.18	0.45	0.04	0.25	0.47	0.02	1.30	0.44	0.08	0.62	0.00	0.52	-0.11	-2.00	0.99	0.25	-1.71
-0.06	-0.22	-0.17	-0.11	0.96	0.39	0.24	0.63	-0.07	-0.56	-1.67	0.56	-0.37	1.00	-0.04	-0.40	1.00
0.04	-0.80	-0.16	-0.95	0.26	0.11	-0.40	-1.25	-0.33	0.88	-0.43	-0.66	-0.54	-0.97	-1.18	1.00	-0.04

Die Parameterwerte sind in normierter Form zusammen mit dem Bild der jeweiligen Testperson grafisch in Bild 5.16 dargestellt.

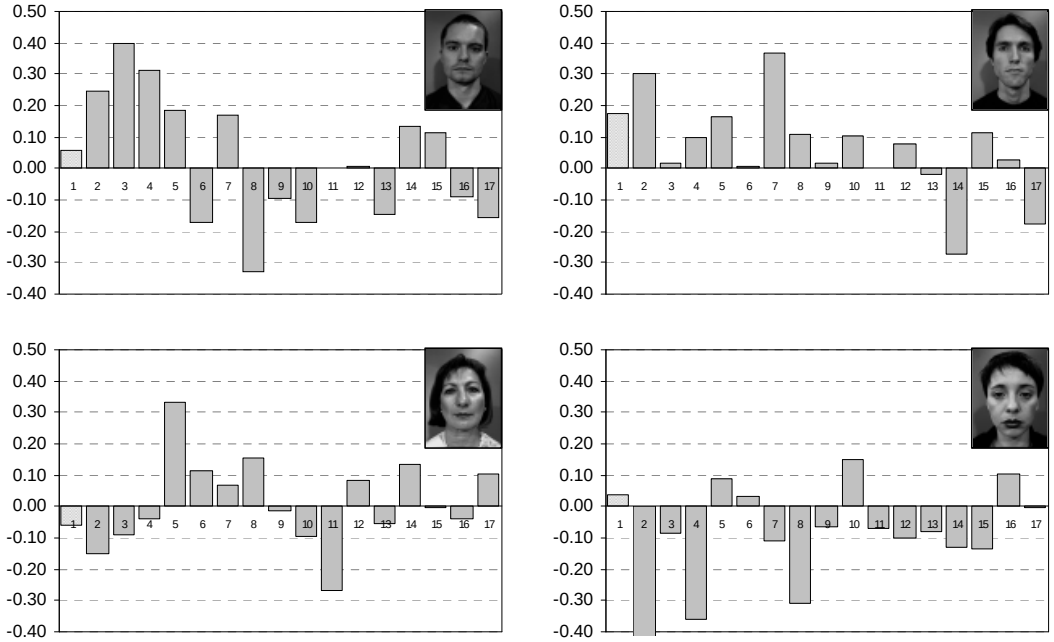


Bild 5.16: ***Normierte Parameterwerte beim neutralen Gesichtsausdruck für 4 verschiedene Testpersonen****. Bei der Normierung wurden die Parameterwerte entsprechend der Größe des zugehörigen Eigenwertes skaliert.*

Zur Normierung wurden die Parameterwerte unter Berücksichtigung der Größe der Eigenwerte λ_i skaliert. Die normierten Werte b_i^* wurden dabei gemäß

$$b_i^* = \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_1}} b_i \quad , \quad i = 1,...,17 \tag{5.5}$$

berechnet. Durch den Faktor $\sqrt{\lambda_i}$ wird eine Gewichtung im Verhältnis zur Größe der Standardabweichung des zugehörigen Eigenvektors erreicht, der Faktor $1/\sqrt{\lambda_1}$ bewirkt eine Skalierung der Werte.

Der Parameter b_1 beschreibt hauptsächlich Formvariationen, die sich aus der Drehung des Kopfes ergeben. Für die Frontalansicht liegt der Wert daher stets nahe bei 0. Die eigentliche Form des Gesichtes, d.h. die Form der einzelnen Gesichtsteile sowie deren relative Lage zueinander, wird dagegen insbesondere durch die höheren Eigenvektoren modelliert. Der Vektor

$$\vec{b}_{id} = (b_2^*, ..., b_{17}^*) \tag{5.6}$$

kodiert daher das Aussehen einer bestimmten Person und kann beispielsweise dazu verwendet werden, um einen Klassifikator zu trainieren, der eine Personen*identifikation* anhand der Lage von $\vec{b}_{id}$ im $\mathbb{R}^{16}$ durchführt. Die grafische Darstellung in Bild 5.16 zeigt das Muster für ein einzelnes Bild einer Person. Um zu einem für die Person charakteristischen mittleren Muster zu kommen und um die Verteilung der Parameterwerte durch eine geeignete Verteilung beschreiben zu können, sollte beim Training eine größere Anzahl an Bildern ausgewertet werden. Die Identifikation der Person kann dann mit einem geeigneten Klassifikator durchgeführt werden.

5.2.5 Charakterisierung von Mimiken

Jeder Eigenvektor des Gesichtsmodells beschreibt eine charakteristische Veränderung der Gesichtsform. Umgekehrt ist auch eine bestimmte Mimik durch eine charakteristische Veränderung der Form oder Position einzelner Gesichtsteile charakterisiert. So ist beispielsweise ein freudiger Gesichtsausdruck häufig gekennzeichnet durch einen breiten, leicht geöffneten Mund und leicht zusammengekniffene Augen. Ist eine Person erstaunt, bewegen sich häufig die Augenbrauen nach oben, und die Augen sind leicht geöffnet. Die Parameterwerte b_i können also dazu verwendet werden, um einen Klassifikator zu trainieren, der die Unterscheidung verschiedener Mimiken ermöglicht.

Die automatisierte Erkennung einer Mimik des Gemütszustandes einer Person kann sowohl auf Einzelbildern als auch durch die Analyse von Bildsequenzen durchgeführt werden. Bei der Analyse von Einzelbildern wird die Mimik durch die Lage des Parametervektors $\vec{b}$ im $\mathbb{R}^{17}$ beschrieben, während sich aus einer Bildsequenz eine Folge von Vektoren ergibt, deren Trajektorie im $\mathbb{R}^{17}$ einen charakteristischen Verlauf zeigt. Die Beurteilung des Gemütszustandes oder der Mimik einer Person ist insbesondere bei Einzelbildern selbst für den menschlichen Betrachter nicht immer eindeutig möglich. Auch kann der Gesichtsausdruck bei verschiedenen Personen selbst bei gleicher Mimik unterschiedlich sein. Daher ist zu erwarten, dass die besten Klassifikationsergebnisse erzielt werden, wenn ein Klassifikator für jede Person separat trainiert wird. Der Parametervektor sollte dabei relativ zum neutralen Gesichtsausdruck der betreffenden Person gemessen werden.

Bild 5.17 zeigt beispielhaft einige Bilder aus einer Sequenz, in der die Mimik *Erstaunen* zu sehen ist. Die gezeigte Mimik ist in erster Linie charakterisiert durch die Variation von Position und Form der Augenbrauen und der Lippen sowie die Änderung der Kinnpartie durch das Öffnen des Mundes. Für die Modellparameter der Eigenvektoren, durch die diese Veränderungen hervorgerufen werden, ergibt sich eine charakteristische Trajektorie. Durch diese wird die Mimik im Parameterraum beschrieben. Bild 5.18 zeigt

den zeitlichen Verlauf der entsprechenden Parameterwerte mit gleitendem Durchschnitt aus fünf Datenpunkten. Zusätzlich ist der Verlauf des Parameters b_1 gezeigt, durch den hauptsächlich das Drehen des Kopfes nach links und rechts modelliert wird.

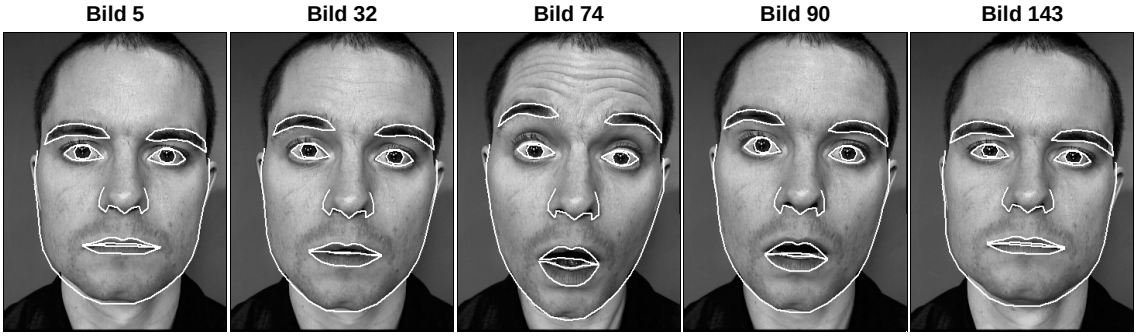


Bild 5.17: ***Bildfolge für Mimik Erstaunen****.*

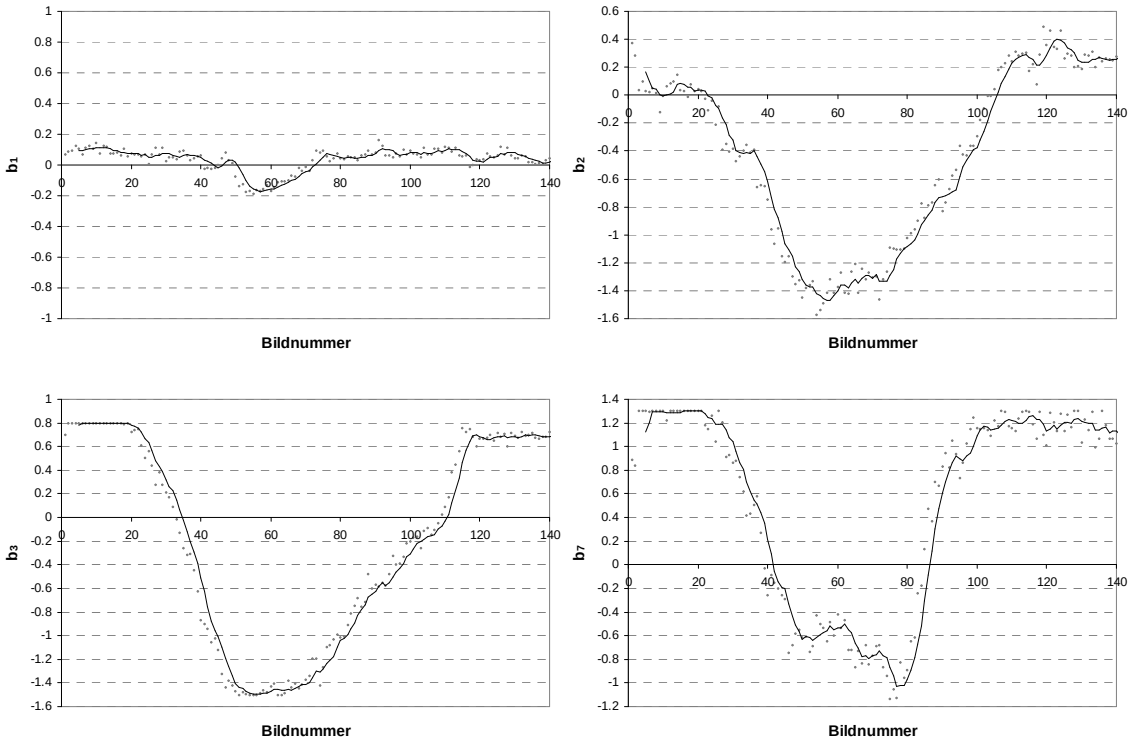


Bild 5.18: ***Zeitlicher Verlauf der Parameterwerte b_1 , b_2 , b_3 und b_7 für Mimik Erstaunen****. Durchgezogene Linie: gleitender Durchschnitt aus fünf Datenpunkten.*

Die Parameterwerte bleiben zunächst konstant und beschreiben den neutralen Gesichtsausdruck der gezeigten Testperson. Die Mimik beginnt daraufhin etwa bei Bild 20

und erreicht ihren stärksten Ausdruck ungefähr bei Bild 60. Schließlich kehren die Parameter ca. bei Bild 110 wieder zu ihrem Ausgangswert zurück.

Da die Person über die gesamte Sequenz frontal in die Kamera blickt, bleibt b_1 annähernd konstant und zeigt nur eine geringe Abweichung vom Ausgangswert. Im Gegensatz dazu weichen die übrigen Parameter im Extremum um bis zu 2 Standardabweichungen von ihrem Ausgangswert ab und entsprechen damit den charakteristischen Änderungen der Form und Position von Kinn, Mund und Augenbrauen (siehe auch Bild 4.15 auf Seite 81):

b_2: Gesicht wird schmaler

b_3: Mund öffnen und zu ‚o' formen, Augenbrauen heben

b_7: Augenbrauen heben, Mund öffnen

Zusätzlich zur Charakterisierung einer Mimik anhand des zeitlichen Verlaufs der b_i kann die Stärke eines Gesichtsausdrucks mit der Größe der Parameterwerte im Extremum quantifiziert werden. Darüber hinaus gestattet die Lokalisation der Gesichtsteile im Bild die automatisierte Vermessung zusätzlicher Merkmale und die Bestimmung weiterer Kenngrößen zur Charakterisierung eines Gesichtsausdruckes wie beispielsweise Augenabstand oder Mundwinkel.

5.3 Performance

Um den Zeitbedarf der einzelnen Schritte bei der Bildinterpretation abschätzen zu können, wurden Laufzeitmessungen mit dem Modell für den menschlichen Oberkörper und mit dem Gesichtsmodell durchgeführt. Die Messungen erfolgten auf einem PC mit Pentium III Prozessor und einer Taktfrequenz von 800 MHz unter dem Betriebssystem Windows NT 4.0. Die Bilder hatten eine Größe von 384*288 Pixel (Personendetektion) bzw. 288*384 Pixel (Gesichtsdetektion).

Tabelle 5.2 zeigt den Zeitbedarf der einzelnen Schritte bei der Kontursuche im Bild. Die angegebenen Zeiten sind Mittelwerte, die sich aus der Auswertung mehrerer Bilder für typische Suchparameter ergeben. Der Zeitbedarf setzt sich aus einem für jedes Bild festen Anteil und dem Zeitbedarf für die eigentliche PDM-Suche zusammen. Die Bilder wurden für die Messungen von Festplatte eingelesen, woraus ein für jedes Bild konstanter Anteil von 12 ms resultiert. Für jedes Bild konstant ist auch der Zeitbedarf für die sich anschließende Bildvorverarbeitung, bestehend aus Bildglättung mit einem Gaußfilter geeigneter Größe und der Berechnung des Kantenbildes. Bei der Detektion der Silhouette (vgl. Bild 5.4 und Bild 5.8) wird zusätzlich zunächst ein Hintergrundbild vom

aktuellen Kamerabild subtrahiert. Die eigentliche PDM-Suche setzt sich aus der Vorsuche und den Iterationsschritten (Algorithmus S. 84) zusammen, wobei der Zeitbedarf linear mit der Anzahl der Iterationen zunimmt. Der vergleichsweise hohe Zeitbedarf für die Vorsuche beim Silhouettenmodell wird durch die Suche an den 9 Positionen, die sich aus der initialen Positionierung ergeben, verursacht (vgl. Bild 4.19). An jeder Position wird ein kompletter Iterationsschritt –bestehend aus *Kantensuche, shape alignment* und *Anpassung der Formparameter*– durchgeführt. Beim Gesichtsmodell dagegen werden die Positionen der einzelnen Konturteile lediglich durch Kantensuche angepasst (siehe Abschnitt 4.5.6).

Tabelle 5.2: ***Zeitbedarf für die Kontursuche im Bild****. Dargestellt sind mittlere Werte für das Modell des menschlichen Oberkörpers (‚Silhouette') und das Gesichtsmodell (‚Gesicht').*

Operation	**Zeitbedarf [ms]**	
	Silhouette	*Gesicht*
Bildeinzug (von Festplatte)	12.0	12.0
Subtraktion des Hintergrundbildes	8.8	-
Bildglättung	7.5	12.3
Berechnung des Kantenbildes	6.2	7.2
Vorsuche	24.5	13.8
Iterationsschritt	2.0	5.0
Sonstiges	9.0	8.0
typisches Beispiel		
Anzahl der Iterationsschritte	10	5
Gesamtzeit Suche im Bild [ms]	**88.0**	**78.3**

Beim Einsatz des Silhouettenmodells in STABIL++ werden bei der Suche typischerweise 10 Iterationsschritte durchgeführt, woraus sich eine Gesamtsuchzeit von etwa *90 ms* pro Bild ergibt. Bei der Gesichtsverfolgung genügen nach der initialen Detektion in den darauffolgenden Bildern wenige Iterationen. Bei 5 Iterationen ergibt sich so eine Gesamtsuchzeit von ca. *80 ms*, was einer Frequenz von *12,5 Hz* oder halber Framerate (PAL Norm) entspricht.

Bei der Personenverfolgung mit STABIL++ werden im Interpretationsprozess zusätzlich zur Bildsegmentierung die extrahierten 3D-Szenenmerkmale in einem Modellmatch den 3D-Modellmerkmalen zugeordnet. Darüber hinaus wird vor dem Bildeinzug die Sichtbarkeit der Suchräume in den einzelnen Kameras überprüft, und das System muss die gefundenen Objektinstanzen verwalten. Der genaue Zeitbedarf für einen Interpreta-

tionszyklus –bestehend aus Bildeinzug, Segmentierung und Modellmatch– hängt außerdem von weiteren Faktoren wie z.B. der Suchraumgröße oder der Hintergrundschätzung ab. Der Zeitbedarf ist für die Merkmale *PDM* und *Hautfarbe* vergleichbar und beträgt bei monokularer Detektion etwa 200 ms, bei Stereodetektion ca. 300 ms. Da weder die implementierten Suchalgorithmen noch das System STABIL++, das sowohl für Anwendungen in der Sicherheitstechnik als auch in der Ergonomie verwendet werden kann, im Hinblick auf die Performance optimiert wurden, ist eine Erhöhung der Framerate leicht möglich (vgl. auch Abschnitt 6.2).

5.4 Grenzen des Verfahrens

Die vorangegangenen Abschnitte haben die Leistungsfähigkeit der in dieser Arbeit entwickelten Modelle und Methoden an einer Reihe von Testsequenzen demonstriert. Dennoch unterliegt die Anwendbarkeit der kantenbasierten Merkmalsextraktion wie auch die Auswertung durch Farbklassifikation gewissen Einschränkungen, da jede Segmentiermethode ihre spezifischen Nachteile hat. Auf die Grenzen der Farbklassifikation wurde bereits in Abschnitt 3.5.2 detailliert eingegangen. Erwähnt seien hier insbesondere noch einmal die eingeschränkte Robustheit gegenüber Beleuchtungsschwankungen und die fehlende Möglichkeit der Auswertung von einkanaligen Grauwertbildern.

Vermeidung entarteter Formen

Bei der Detektion mit Punktverteilungsmodellen muss generell darauf geachtet werden, dass das verwendete PDM sämtliche Formen erzeugen kann, die die im Bild zu detektierenden Objekte annehmen können. Hierzu sind beim Trainingsdatensatz eine hinreichend große Anzahl an repräsentativen Bildern auszuwerten. Für Objektklassen, die sich stark voneinander unterscheiden, müssen darüber hinaus mehrere verschiedene PDMs verwendet werden.

Die Variationsmöglichkeiten lassen sich weiterhin über die Anzahl t der verwendeten Eigenvektoren sowie die obere und untere Schranke ($\bar{b}_i$ und $\underline{b}_i$) für die einzelnen Parameterwerte beeinflussen. Eine große Anzahl an Eigenvektoren und betragsmäßig große Werte für die Schranken führen zu einer größeren Variationsbreite, es kommt jedoch zunehmend auch zu entarteten Formen, was dazu führt, dass die Suche weniger robust wird. Wenn im Extremfall sämtliche $2n$ Eigenvektoren verwendet werden und die Beschränkung komplett aufgehoben wird, spannen die Eigenvektoren den ganzen $\mathbb{R}^{2n}$ auf (d.h. $Span\,(\bar{p}_i,...,\bar{p}_{2n}) = \mathbb{R}^{2n}$), so dass *jede* Kontur mit $2n$ Punkten erzeugt werden kann. In Bild 5.19 wurden die ersten 15 Eigenvektoren des Silhouettenmodells verwendet und die Parameterwerte b_i, $i=1 \ldots 15$ schrittweise um eine Standardabweichung erhöht.

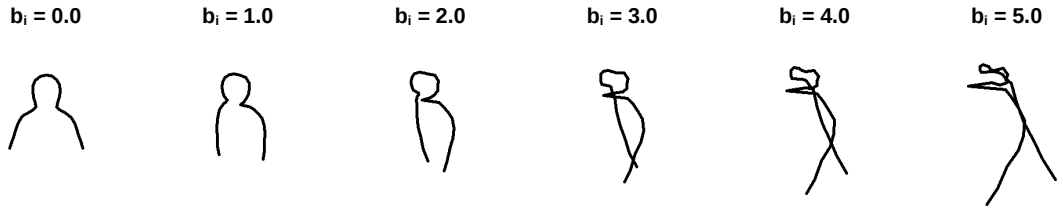


Bild 5.19: ***Erzeugung entarteter Konturen****. Zur Erzeugung der Silhouetten wurden die ersten 15 Eigenvektoren des Silhouettenmodells verwendet und die Parameterwerte b_i auf die angegebene Größe gesetzt.*

Positionierung der initialen Kontur

Die iterative PDM-Suche im Bild ist ein lokales Verfahren, bei dem die Position der einzelnen Konturteile nur begrenzt verändert werden kann. Die initiale Kontur muss daher so positioniert werden, dass der tatsächliche Kantenverlauf möglichst gut approximiert wird und die Suche lediglich eine Anpassung der Formparameter vornehmen muss. Bild 5.20 zeigt links die Positionierung der initialen Kontur mit den zu den einzelnen Modellpunkten gehörenden Suchradien. Die Konturpunkte sind in diesem Beispiel zu weit vom tatsächlichen Kantenverlauf entfernt. Die Suche kommt zwar nach einigen Iterationen in einen stationären Zustand, die Bildstrukturen werden aber trotz der Anpassung der Positionen für die einzelnen Konturteile nur unzureichend approximiert.

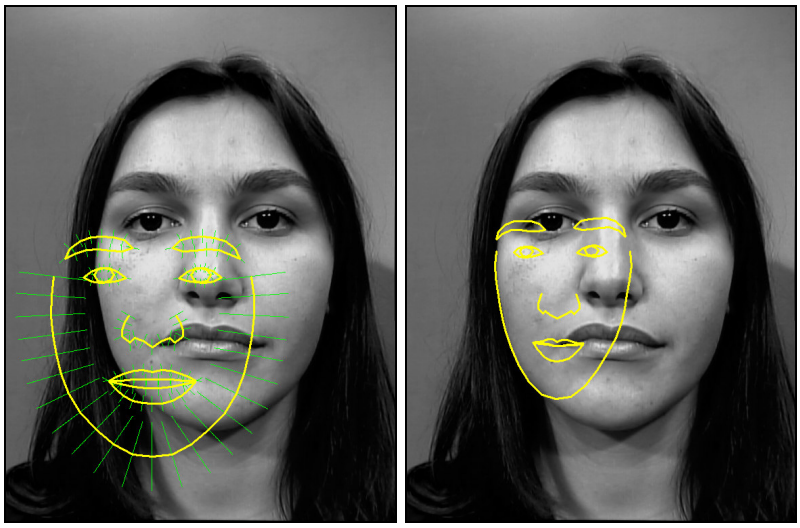


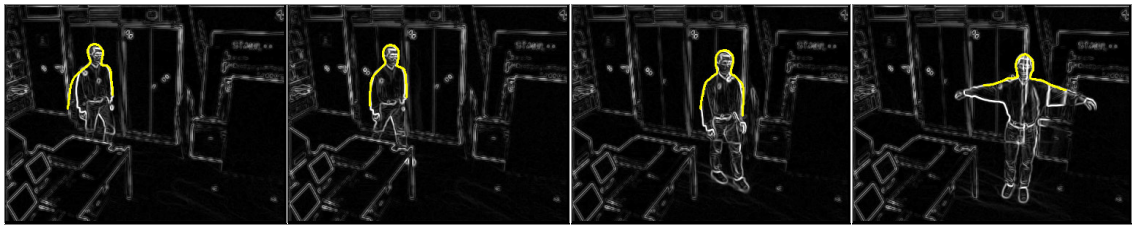
Bild 5.20: ***Fehlerhafte Anpassung bei schlechter Positionierung der initialen Kontur****. Links: Initialisierung und Suchradius. Rechts: stationärer Zustand nach 50 Iterationen.*

Fehlerhafte Anpassung bei schlechter Hintergrundschätzung

Zu fehlerhaften Anpassungen kommt es auch, wenn im Kantenbild Strukturen auftreten, die nicht von der Kontur des Objektes herrühren.

Bei Subtraktion eines Hintergrundbildes kommt es häufig zu Kantenstrukturen im Inneren, wenn sich das Objekt vor stark strukturiertem Hintergrund bewegt (vgl. Bild 4.17). Die Anpassung an diese inneren Kanten kann durch *blowing* vermieden werden (siehe Bild 4.21). Zu erhöhten Grauwerten im Kantenbild kommt es auch, wenn der Hintergrund schlecht geschätzt wurde. Bild 5.21 zeigt ein Beispiel für fehlerhafte Anpassungen, die sich in diesem Fall bei der Suche ergeben können. Im ersten Bild von Kamera 1 approximiert der linke Arm des Modells zunächst eine falsche Kante im Bild, kann sich jedoch im Verlauf der Sequenz wieder an die richtige Struktur anpassen. Bei Kamera 2 dagegen ist keine sinnvolle Anpassung an die Bildstrukturen mehr möglich. Der Verlauf des linken Armes kann im Kantenbild nicht von den übrigen Strukturen unterschieden werden und wird im weiteren Verlauf der Sequenz auch nicht mehr korrekt wiedergefunden.

Kamera 1



Kamera 2

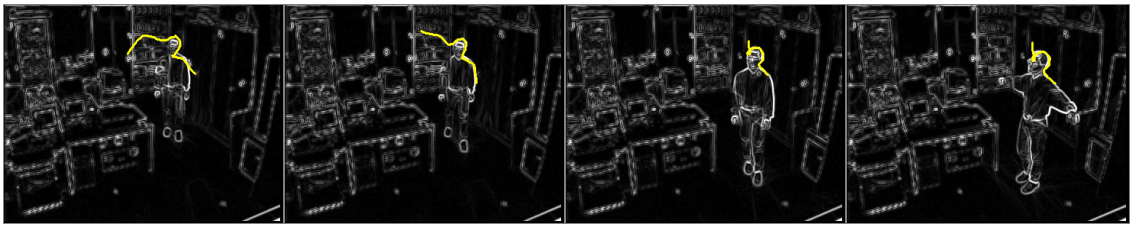


Bild 5.21: ***Fehlerhafte Anpassung bei schlechter Hintergrundschätzung****.*

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Bei vielen System, in denen eine Interaktion zwischen Mensch und Maschine stattfindet, bei der Überwachung von Gebäuden in der Sicherheitstechnik, aber auch beispielsweise in medizinischen Anwendungen, spielt die automatisierte Detektion und Vermessung menschlicher Bewegungen in Videobildfolgen eine zunehmende Rolle. Unterschiede im Aussehen zwischen verschiedenen Personen müssen hierbei ebenso berücksichtigt werden wie Unterschiede, die sich aufgrund verschiedener Ansichten und Bewegungen oder durch sich ändernde Umgebungsbedingungen ergeben. Um eine robuste Interpretation einer aufgenommenen Szene zu erreichen und die spezifischen Schwächen einzelner Segmentiermethoden zu kompensieren, ist es sinnvoll, verschiedene Verfahren zur Bildsegmentierung einzusetzen und unterschiedliche Merkmale aus den Bilddaten zu extrahieren.

In dieser Arbeit wird ein modellbasiertes System vorgestellt, mit dem formvariable Objekte in Videobildfolgen anhand verschiedener Merkmale verfolgt werden können. Zusätzlich zur Bildsegmentierung durch Farbklassifikation wird insbesondere auch auf die kantenbasierte Merkmalsextraktion mit flexiblen Konturmodellen eingegangen. Die Leistungsfähigkeit der entwickelten Modelle und Algorithmen wird durch den Einsatz zur Vermessung von Gestik und Mimik bei der Beobachtung menschlicher Bewegungen demonstriert.

Das dem System zugrunde liegende generische Objektmodell (OM) erlaubt die Modellierung beliebig aufgebauter artikularer 3D-Objekte aus mehreren Objektmodellteilen (OMPs). Zur näheren Charakterisierung können die einzelnen OMPs durch Merkmale beschrieben werden. Hierdurch werden auch die zur Bildsegmentierung verwendeten Methoden festgelegt.

Zur Realisierung kantenbasierter Merkmale werden statistische Punktverteilungsmodelle (*point distribution models*, PDMs) verwendet. Bei diesen wird eine flexible Kontur durch eine Menge von Punktkoordinaten beschrieben. Durch eine Hauptachsentransformation (*principal component analysis*, PCA) ergibt sich aus einem repräsentativen Trainingsdatensatz eine mittlere Kontur $\overline{S}$ sowie eine Reihe von Eigenvektoren, mit denen charakteristische Formvariationen beschrieben werden können. Die Variationsmöglichkeiten des Modells ergeben sich so in natürlicher Weise aus den Trainingsdaten. Durch Linearkombination der Eigenvektoren können neue Konturen erzeugt werden.

Die Größe der Parameterwerte b_i bestimmt dabei die Gewichtung der einzelnen Eigenvektoren und somit die Form der Kontur.

Im Rahmen der vorliegenden Arbeit werden Punktverteilungsmodelle für das Gesicht und die für Personen charakteristische Kopf-Schulter Partie der menschlichen Silhouette aus einer Vielzahl von Trainingsbildern erstellt. Die Definition der Trainingskonturen erfolgt halbautomatisch mit einem interaktiven Tool, das die Definition beliebiger –auch mehrteiliger– Konturen zulässt. Die Extraktion der Bildkanten erfolgt automatisiert, so dass eine schnelle Generierung des Trainingsdatensatzes möglich ist.

Die Modelle werden zur kantenbasierten Merkmalsextraktion eingesetzt. Für die Modellsuche im Bild wird ein Verfahren verwendet, bei dem die Form einer Startkontur iterativ an die Kantenstrukturen im Bild angepasst wird. Die Form der Kontur wird dabei stets konsistent mit dem zugrunde liegenden PDM-Modell gehalten. Die implementierten Klassen und Algorithmen sind generisch und für beliebige Punktverteilungsmodelle einsetzbar.

Personenverfolgung

Zur Verfolgung von Personen, z.B. bei der Überwachung sensibler Gebäudebereiche, bietet sich die charakteristische Kopf-Schulter Partie an. Das entsprechende Konturmodell für den menschlichen Oberkörper beschreibt die Silhouette durch 35 Punkte. Bei der Modellerstellung wurden Bilder verschiedener Personen ebenso berücksichtigt wie unterschiedliche Haltungen und Ansichten aus mehreren Kameraperspektiven. Die ersten 6 Eigenvektoren des PDM beschreiben *95%* der Gesamtvariation des Trainingsdatensatzes, die zugehörigen Parameterwerte werden im Bereich von *–2.0* bis *+2.0* Standardabweichungen variiert.

Das Silhouettenmodell wird eingesetzt, um das Objektmodellteil *Kopf* des artikularen Objektmodells mit einem kantenbasierten Merkmal zu beschreiben. Mit einem kalibrierten Stereoaufbau können so Personen im Raum verfolgt und die 3D-Trajektorie der Bewegung bestimmt werden. Ist die Person nur in einer Kamera sichtbar, kann unter Verwendung von Modellwissen eine monokulare Schätzung durchgeführt und so ebenfalls auf die 3D-Bewegung geschlossen werden. Die Segmentierung mit dem Merkmal *PDM* wird der Segmentierung mit dem Merkmal *Hautfarbe* zur Detektion des Gesichtes mit einem Farbklassifikator gegenüber gestellt. Die durch den Abstand der Sichtstrahlen beim Stereo-Match definierte Messgenauigkeit beträgt in beiden Fällen einige wenige Zentimeter. Bei teilweise überlappenden Sichtbereichen ist es weiterhin möglich, gefundene Objektmodellinstanzen über mehrere Kameras hinweg zu verfolgen. Hierdurch können z.B. mehrere Räume in einem größeren Gebäudekomplex durch die Verwendung einer entsprechend großen Anzahl an Kameras lückenlos überwacht werden.

Gegenüber der Bildsegmentierung durch Farbklassifikation hat die Verwendung des kantenbasierten Merkmals PDM mehrere Vorteile. Bei sich ändernden Beleuchtungs-

verhältnissen kommt es bei der Extraktion von Farbinformationen aus dem Bild leicht zu Fehlklassifikationen (siehe Bild 3.12). Durch die Verwendung des Merkmals PDM wird die Segmentierung weitestgehend unabhängig von den Umgebungsbedingungen. Die Verfolgung von Personen ist weiterhin beispielsweise auch dann möglich, wenn die Person sich umdreht und das Gesicht nicht mehr sichtbar ist (siehe Sequenz auf Seite 116). Die kantenbasierte Merkmalsextraktion kann darüber hinaus auch auf den weniger speicherintensiven Grauwertbildern erfolgen.

Gesichtsdetektion

Zur genaueren Lokalisation des Kopfes oder bei der Interpretation von Bildern verschiedener Mimiken müssen die Gesichtszüge genauer vermessen werden.

Im Rahmen der Arbeit wurde eine Datenbank mit über 1000 Bildern erstellt, die neben Aufnahmen verschiedener Individuen auch Bilder unterschiedlicher Mimiken enthält. Dabei wurden sowohl Bewegungen des Kopfes als Ganzes berücksichtigt als auch Variationen, die sich aus der Formvariation einzelner Gesichtsteile und deren relative Bewegung zueinander ergeben. Das aus 212 repräsentativen Trainingsbildern erstellte zehnteilige Modell besteht aus insgesamt 134 Punkten. Um auch hier 95% der Gesamtvariation des Trainingsdatensatzes zu berücksichtigen, werden bei der Suche im Bild Linearkombinationen der ersten 17 Eigenvektoren verwendet. Die Variation der zugehörigen Parameterwerte erfolgt für jeden Eigenvektor separat.

Anhand von Beispielen wird gezeigt, wie mit dem Gesichtsmodell und den implementierten Suchalgorithmen Gesichter in Einzelbildern lokalisiert und in Videobildfolgen kontinuierlich verfolgt werden können. Durch die zusätzliche Berücksichtigung von Grauwertinformationen in einem heuristischen Grauwertmodell wird hierbei eine wesentlich bessere Anpassung des Modells an die Bildstrukturen erreicht als dies mit dem reinen Kantenmodell der Fall ist.

Bei der Auswertung von Bildfolgen verschiedener Mimiken ergeben sich für die Parameterwerte b_i im Parameterraum charakteristische Trajektorien, die zur Charakterisierung der untersuchten Mimik verwendet werden können. Darüber hinaus ist es durch die genaue Lokalisation der einzelnen Gesichtsteile im Bild leicht möglich, zusätzliche Merkmale, wie z.B. Augenabstand, Mundwinkel usw., zu bestimmen. Durch Auswertung von Einzelbildern und Videobildfolgen kann so die Merkmalsextraktion zum Aufbau eines geeigneten Klassifikators für Mimiken aus Trainingssequenzen weitgehend automatisiert werden.

Fazit

Der vorgestellte Ansatz zur Verfolgung formvariabler Objekte in Videobildfolgen kombiniert farb- und kantenbasierte Segmentierverfahren mit Methoden der 3D-

Objektrekonstruktion. Er eignet sich für den Einsatz in multimodalen Systemen zum Erfassen und Verstehen menschlicher Verhaltensweisen ebenso wie zum Einsatz im Bereich der Sicherheitstechnik oder für ergonomische Studien zur Gewinnung anthropometrischer Daten.

6.2 Ausblick

Das vorgestellte System erlaubt die Erfassung menschlicher Bewegungen bei der Beobachtung von Gestik und Mimik. Dies ist die Grundlage für eine Vielzahl von Anwendungen beispielsweise in der Sicherheitstechnik oder im Bereich der Mensch-Maschine Interaktion. Die bestehenden Modelle und Algorithmen bilden den Ausgangspunkt für eine Reihe weiterführender Arbeiten.

Entwurf geeigneter Klassifikatoren für Gestik und Mimik

Neben der reinen Vermessung menschlicher Bewegungen gewinnt zunehmend auch die automatisierte Analyse der Messdaten an Bedeutung. Eine natürlicher nächster Schritt ist die Erweiterung des Systems um Module zur automatisierten Interpretation von Gestik und Mimik. Die in dieser Arbeit entwickelten Modelle und Algorithmen können dazu verwendet werden, um aus Bildfolgen geeignete Merkmale zu extrahieren, die verschiedene Gesten und Mimiken charakterisieren und mit denen geeignete Klassifikatoren trainiert werden können. Die Auswertung kann dabei weitestgehend automatisiert werden.

Aufbau statistischer Modelle zur Objektbeschreibung

Das System erlaubt die automatisierte Merkmalsextraktion aus Videobildfolgen. Die extrahierten Objekteigenschaften können zum Aufbau statistischer Modelle verwendet werden, die z.B. das Aussehen einer Person oder die Art, wie sie sich bewegt, beschreiben. Für das Aussehen bieten sich Merkmale wie z.B. die Größe der Person, deren Haarfarbe oder auch texturelle Eigenschaften beispielsweise zur Beschreibung der Kleidung an. Die Art der Bewegung ergibt sich aus der Vermessung von Trajektorien der Extremitäten und der Gesichtszüge. Die Modelle könnten das Wiedererkennen einer Person auch nach einem längeren Zeitraum ermöglichen.

Weitere Merkmale und simultane Auswertung

Im Bereich der Sicherheitstechnik ist eine zuverlässige Verfolgung von Personen auch bei stark variierenden Umgebungsbedingungen notwendig. Damit das System auch außerhalb der Laborumgebung eingesetzt werden kann, bietet sich eine Erweiterung der

bestehenden Modelle und Algorithmen um zusätzliche Verfahren zur Bildsegmentierung an, um die Robustheit zu erhöhen. Die bestehende Systemarchitektur sollte dahingehend erweitert werden, dass die *gleichzeitige* Verwendung mehrerer Merkmale möglich wird. Je nach Vertrauenswürdigkeit müssen die unterschiedlichen Informationen geeignet kombiniert werden.

Bei schlechter Schätzung des Hintergrundes führt die Subtraktion eines Hintergrundbildes vom aktuellen Kamerabild zum verstärkten Auftreten von Kanten. Hierdurch kommt es vermehrt zu fehlerhaften Anpassungen des Konturmodells an die Bildstrukturen. Hier könnte die zusätzliche Berücksichtigung von Informationen aus dem Differenzbild zweier aufeinanderfolgender Frames bei der Kantenberechnung helfen, um so unabhängiger vom Hintergrund zu werden. Bild 6.1 zeigt verschiedene Kantenbilder bei der Anwendung des Sobelfilters. Durch Multiplikation des Kantenbildes, das sich aus dem Differenzbild zweier aufeinanderfolgender Frames ergibt (2. Bild von rechts), mit dem Kantenbild des Originalbildes (links) lassen sich sowohl die im Kantenbild des Differenzbildes auftretende Verdopplung bewegter Objekte als auch die Kanten im Hintergrund weitestgehend vermeiden (rechts). Hierzu sollten entsprechende Tests im Außenbereich bei wechselnder Beleuchtung und vor beweglichem Hintergrund (z.B. Äste von Bäumen) durchgeführt werden.

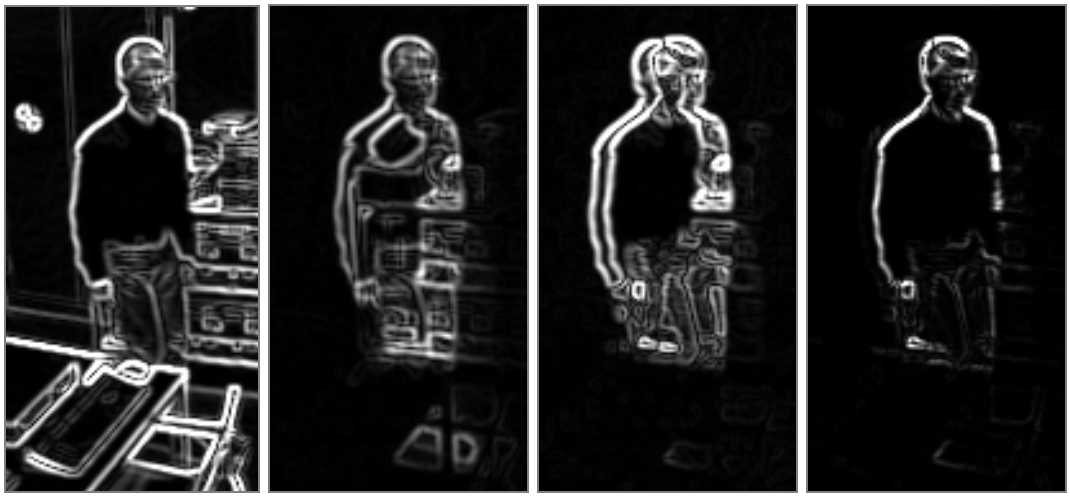


Bild 6.1: ***Kantenextraktion durch den Sobelfilter****. Links: angewendet auf Originalbild. 2. v. links: nach Subtraktion eines Hintergrundbildes. 2. v. rechts: nach Subtraktion zweier aufeinanderfolgender Frames. Rechts: Multiplikation von Sobelfilter angewendet auf Originalbild und Differenzbild zweier aufeinander folgender Frames.*

Bei der ausschließlichen Verwendung von Kanteninformationen kann es zur fehlerhaften Anpassung des Modells an die Bildstrukturen kommen (vgl. Bild 5.15 und Bild 5.21). Die Erweiterung des Konturmodells um ein heuristisches Grauwertmodell führt bei der Gesichtsdetektion zu einer verbesserten Anpassung. Die zusätzliche Auswertung von Textureigenschaften könnte zu einer weiteren Verbesserung der Segmentierung

beitragen. Für die Umgebung jedes Konturpunktes könnte hierzu ein Template der Grauwerte im ersten Bild gelernt werden, das in den darauffolgenden Frames kontinuierlich aktualisiert wird und in Kombination mit einem Kalman-Filter auch als Schätzung verwendet werden könnte.

Performancesteigerung und Messgenauigkeit

Weder das System STABIL++ noch die entwickelten Suchalgorithmen wurden bislang im Hinblick auf Ausführungsgeschwindigkeit und Messgenauigkeit optimiert. Zum Erreichen des Echtzeitbetriebes mit voller Framerate und einer erhöhten Messgenauigkeit bei der Stereodetektion sind eine Reihe von Maßnahmen möglich, von denen hier einige beispielhaft genannt seien.

Der Zeitbedarf für den Bildeinzug von Festplatte beträgt bei den in Abschnitt 5.3 gemachten Messungen etwa *12 ms*. Im Live-Betrieb hängt der genaue Wert unter anderem davon ab, mit welcher Auflösung das Bild eingezogen wird, welches Halbbild verwendet wird und zu welchem Zeitpunkt in Bezug auf das Sync-Signal der Kamera der Bildeinzug gestartet wird. Der Bildeinzug kann jedoch zeitlich parallel zur Auswertung des vorhergehenden Frames stattfinden. Bei der offline-Auswertung bietet sich hierfür z.B. ein separater Thread an. Im Live-Betrieb kann das Bild durch so genannten asynchronen Bildeinzug mittels *direct memory access* (DMA) direkt in den Hauptspeicher transferiert werden, so dass auch hier Bildaufnahme und Bildsegmentierung parallelisierbar sind.

Einen großen Teil des Zeitbedarfs bei der Kontursuche im Bild wird durch die Bildvorverarbeitung –bestehend aus Subtraktion des Hintergrundbildes, Bildglättung und Berechnung des Kantenbildes– verursacht (siehe Tabelle 5.2), da die Berechnung bislang auf dem kompletten Bild durchgeführt wird. Eine Beschränkung auf die Größe der Suchintervalle bei der Kantensuche (vgl. Abschnitt 4.5.7) würde hier deutliche Geschwindigkeitsvorteile bringen. Auch der Zeitbedarf für die Iterationsschritte selbst lässt sich durch die Verwendung geeigneter Pointer-Strukturen noch deutlich reduzieren.

Die Messgenauigkeit, mit der die Position einer sich bewegenden Person durch Stereomessung ermittelt werden kann, wird unter anderem durch den sequentiellen Bildeinzug begrenzt. Der Aufnahmezeitpunkt von Bildern verschiedener Kameras ist leicht unterschiedlich (Abschnitt 5.1.1 und 5.1.3). Verbesserungen lassen sich hier leicht z.B. durch Synchronisation der Kameras erreichen oder auch durch die Verwendung hochwertiger Digitalisierungskarten, die einen *Parallel*einzug mehrerer Bilder gestatten, so dass es keine zeitliche Differenz zwischen den einzelnen Bildern gibt.

Anhang

A Homogene Koordinaten

Das Koordinatensystem der Kamera (KKS) fällt im Allgemeinen nicht mit dem Weltkoordinatensystem (WKS) zusammen, sondern ist im Vergleich zu diesem rotiert und verschoben. Die Beschreibung der Lage der beiden Systeme zueinander geschieht durch die Angabe einer Rotationsmatrix ${}^{c}R_{w}$ und eines Translationsvektors $\vec{T} = (t_x, t_y, t_z)$. Ein Weltpunkt $\vec{p}_w = (x_w, y_w, z_w)^T$ kann durch

$$\vec{p}_c = (x_c, y_c, z_c)^T = {}^{c}R_w^{-1}(\vec{p}_w - \vec{T}) \qquad \text{(A.1)}$$

ins Koordinatensystem der Kamera transformiert werden (siehe Bild A.1).

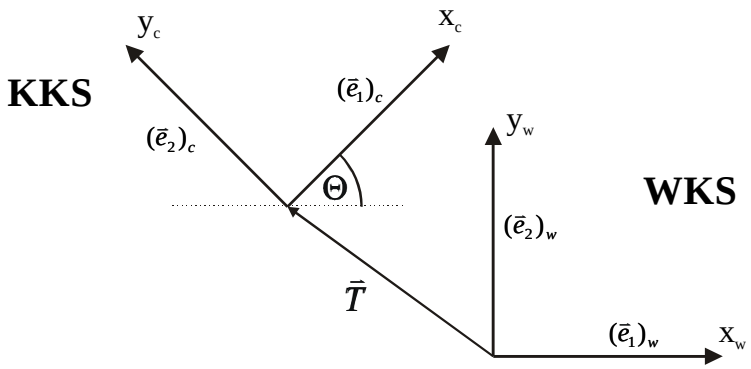


Bild A.1: ***Transformation zwischen Kamerakoordinatensystem (KKS) und Weltkoordinatensystem (WKS)***.

Rotationsmatrix und Translationsvektor beschreiben in dieser Darstellung die Lage des Kamera*koordinatensystems* im Weltkoordinatensystem. Die Spaltenvektoren von ${}^{c}R_w \in \mathbb{R}^{3x3}$ enthalten also die Koordinaten der Basisvektoren $(\vec{e}_i)_c$ des Kamerakoordinatensystems, ausgedrückt in der Basis $(\vec{e}_i)_w$ des Weltkoordinatensystems, für einen Translationsvektor $\vec{T} = \vec{o}$.

Die durch Gleichung (A.1) beschriebene Umrechnung der Koordinaten kann durch die Einführung so genannter homogener Koordinaten vereinfacht dargestellt werden. Hierbei werden die Koordinatenvektoren gemäß

$$\vec{p}_w \rightarrow \begin{pmatrix} \vec{p}_w \\ 1 \end{pmatrix} \quad , \quad \vec{p}_c \rightarrow \begin{pmatrix} \vec{p}_c \\ 1 \end{pmatrix} \quad , \quad \vec{T} \rightarrow \begin{pmatrix} \vec{T} \\ 1 \end{pmatrix} \tag{A.2}$$

um eine Dimension erweitert.

Translation und Rotation werden in einer Transformationsmatrix

$${}^{w}T_c = \begin{pmatrix} & & & t_x \\ & {}^{c}R_w & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.3}$$

zusammengefasst, mit der sich die Umrechnung in Weltkoordinaten durch

$$\vec{p}_w = {}^{w}T_c \; \vec{p}_c \tag{A.4}$$

ausdrücken lässt. Analog kann die Umrechnung aus (A.1) als

$$\vec{p}_c = {}^{c}T_w \; \vec{p}_w \tag{A.5}$$

geschrieben werden mit

$${}^{c}T_w = {}^{w}T_c^{-1} = \begin{pmatrix} & & & -t_x \\ & {}^{c}R_w^{-1} & & -t_y \\ & & & -t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{21} & r_{31} & -t_x \\ r_{12} & r_{22} & r_{32} & -t_y \\ r_{13} & r_{23} & r_{33} & -t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.6}$$

B Shape Alignment

Zur Ausrichtung zweier Konturen S_1 und S_2 bei der Modellsuche im Bild muss eine Transformation $T(s,\Theta,t_x,t_y)$ bestimmt werden, die S_1 durch Skalierung s, Translation (t_x,t_y) und Rotation Θ so transformiert, dass die Summe der Abstandsquadrate korrespondierender Punkte minimiert wird (vgl. Abschnitt 4.5.8). Die unterschiedlich starke Berücksichtigung einzelner Punkte durch einen Gewichtsvektor $\vec{w} = (w_1 \ldots w_n)^T$ führt zusammen mit den Definitionen

$$a = s \cdot \cos\Theta \tag{B.1}$$

$$b = s \cdot \sin\Theta \tag{B.2}$$

auf den zu minimierenden Ausdruck

$$\begin{aligned} D &= \sum_{i=1}^{n} w_i (dx_i^{\,2} + dy_i^{\,2}) \\ &= \sum_{i=1}^{n} w_i (ax_{1i} - by_{1i} + t_x - x_{2i})^2 + \sum_{i=1}^{n} w_i (bx_{1i} + ay_{1i} + t_y - y_{2i})^2 \end{aligned} \tag{B.3}$$

Die Methode der kleinsten Quadrate führt bei der Minimierung von D durch Bildung der partiellen Ableitungen $\partial_a, \partial_b, \partial_{t_x}, \partial_{t_y}$ zu folgenden Gleichungen:

$$\frac{\partial D}{\partial a} = 2\sum_{i=1}^{n} w_i \left\{ a(x_{1i}^{\,2} + y_{1i}^{\,2}) + t_x x_{1i} + t_y y_{1i} - x_{1i}x_{2i} - y_{1i}y_{2i} \right\} \stackrel{!}{=} 0 \tag{B.4}$$

$$\frac{\partial D}{\partial b} = 2\sum_{i=1}^{n} w_i \left\{ b(x_{1i}^{\,2} - y_{1i}^{\,2}) - t_x y_{1i} + t_y x_{1i} + x_{2i}y_{1i} - x_{1i}y_{2i} \right\} \stackrel{!}{=} 0 \tag{B.5}$$

$$\frac{\partial D}{\partial t_x} = 2\sum_{i=1}^{n} w_i (ax_{1i} - by_{1i} + t_x - x_{2i}) \stackrel{!}{=} 0 \tag{B.6}$$

$$\frac{\partial D}{\partial t_y} = 2\sum_{i=1}^{n} w_i (ay_{1i} + bx_{1i} + t_y - y_{2i}) \stackrel{!}{=} 0 \tag{B.7}$$

Mit den Definitionen

$$P_x = \sum_{i=1}^{n} w_i x_{1i} \quad , \quad Q_x = \sum_{i=1}^{n} w_i x_{2i} \tag{B.8}$$

$$P_y = \sum_{i=1}^{n} w_i y_{1i} \quad , \quad Q_y = \sum_{i=1}^{n} w_i y_{2i} \tag{B.9}$$

$$C_1 = \sum_{i=1}^{n} w_i (x_{1i} x_{2i} + y_{1i} y_{2i}) \quad , \quad C_2 = \sum_{i=1}^{n} w_i (x_{1i} y_{2i} - y_{1i} x_{2i}) \tag{B.10}$$

$$Z = \sum_{i=1}^{n} w_i ({x_{1i}}^2 + {y_{1i}}^2) \quad , \quad W = \sum_{i=1}^{n} w_i \tag{B.11}$$

ergibt sich für die gesuchten Transformationsparameter folgendes Gleichungssystem:

$$M \cdot \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} Z & 0 & P_x & P_y \\ 0 & Z & -P_y & P_x \\ P_x & -P_y & W & 0 \\ P_y & P_x & 0 & W \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ Q_x \\ Q_y \end{pmatrix} \tag{B.12}$$

Die Bildung der Inversen der Koeffizientenmatrix *M* führt auf

$$M^{-1} = \frac{1}{WZ - ({P_x}^2 + {P_y}^2)} \begin{pmatrix} W & 0 & -P_x & -P_y \\ 0 & W & P_y & -P_x \\ -P_x & P_y & Z & 0 \\ -P_y & -P_x & 0 & Z \end{pmatrix} \tag{B.13}$$

und so zu Ausdrücken für die Parameter der Transformation:

$$a = \frac{C_1 W - P_x Q_x - P_y Q_y}{WZ - ({P_x}^2 + {P_y}^2)} \tag{B.14}$$

$$b = \frac{C_2 W + P_y Q_x - P_x Q_y}{WZ - ({P_x}^2 + {P_y}^2)} \tag{B.15}$$

$$t_x = \frac{-C_1 P_x + C_2 P_y + ZQ_x}{WZ - (P_x^{\,2} + P_y^{\,2})} \tag{B.16}$$

$$t_y = \frac{-C_1 P_y - C_2 P_x + ZQ_y}{WZ - (P_x^{\,2} + P_y^{\,2})} \tag{B.17}$$

Aus *a* und *b* ergibt sich der Rotationswinkel zu

$$\Theta = \begin{cases} \arctan(b/a) & , \quad a \neq 0 \\ \pi/2 & , \quad a = 0 \end{cases} \tag{B.18}$$

und der Skalierungsfaktor *s* gemäß

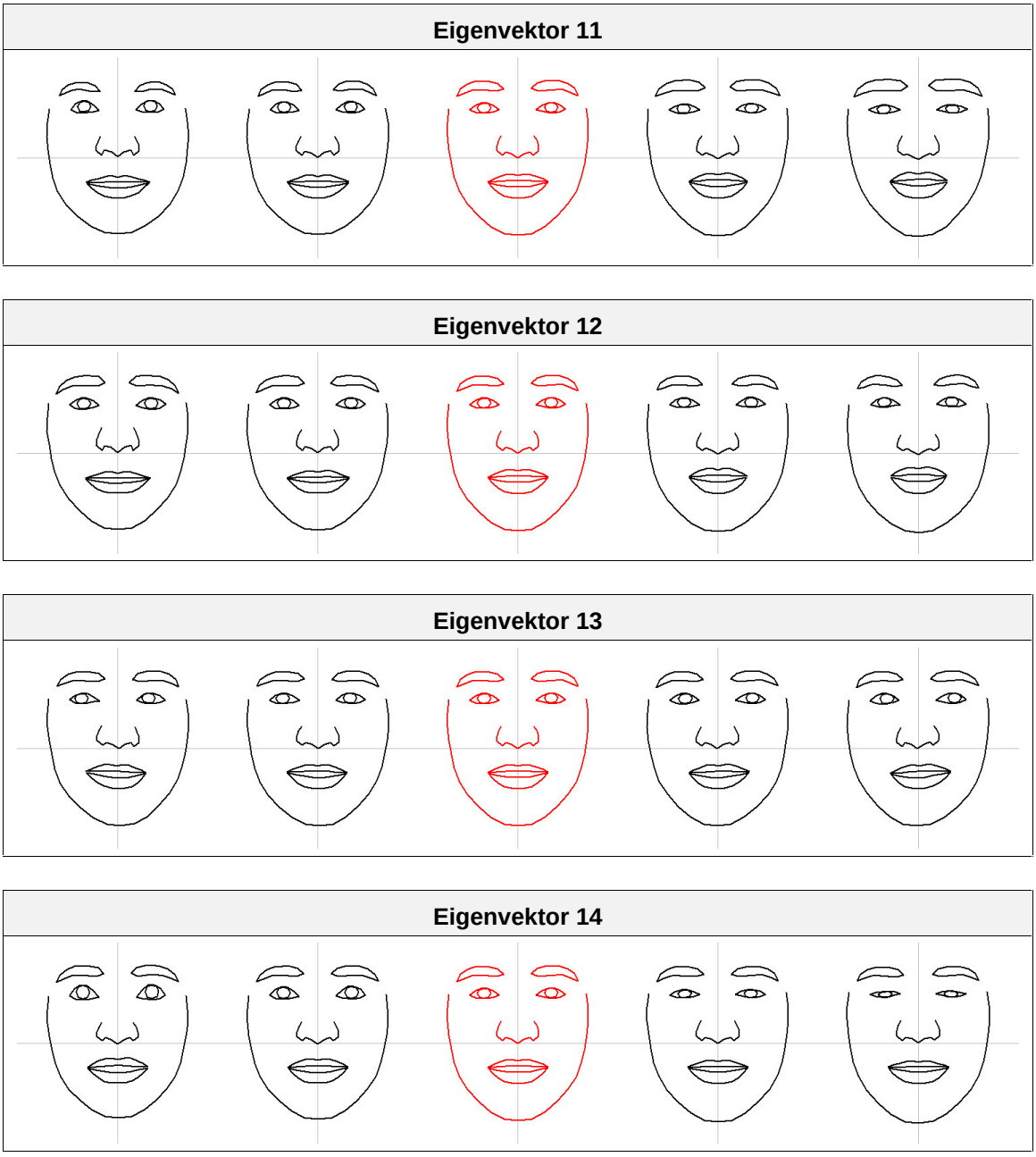
$$s = \sqrt{a^2 + b^2} \tag{B.19}$$

Ausrichtung mehrerer Konturen

Die Ausrichtung einer Menge von *N* Trainingskonturen S_i erfolgt nach dem in [CTC95] beschriebenen Verfahren. Hierzu werden zunächst sämtliche Trainingskonturen durch *shape alignment* an der ersten Kontur des Trainingsdatensatzes ausgerichtet. Aus den ausgerichteten Konturen wird eine mittlere Kontur $\bar{S}$ bestimmt, die durch Rotation, Skalierung und Translation geeignet normiert wird. Die Normierung kann z.B. so gewählt werden, dass der Schwerpunkt von $\bar{S}$ mit dem Koordinatenursprung zusammenfällt. An dieser normierten Kontur werden wiederum alle Konturen ausgerichtet. Das Verfahren wird so lange wiederholt, bis die mittlere Transformation bei der wiederholten Ausrichtung der Konturen an der normierten Kontur die Identitätsabbildung ist.

C Variation des Gesichtsmodells durch die Eigenvektoren 11-17

Bei dem in Abschnitt 4.4.7 vorgestellten Gesichtsmodell beschreiben die ersten 17 Eigenvektoren 95% der Gesamtvariation. In Bild 4.15 wurde aus Gründen der Übersichtlichkeit nur die Änderung der Gesichtsform für die ersten 10 Eigenvektoren gezeigt. Da bei der PDM-Suche jedoch alle 17 Eigenvektoren verwendet werden, demonstriert Bild C.1 ergänzend die Änderung der Gesichtsform bei Variation der Eigenvektoren 11-17 im Bereich –2.0 ... +2.0 Standardabweichungen.



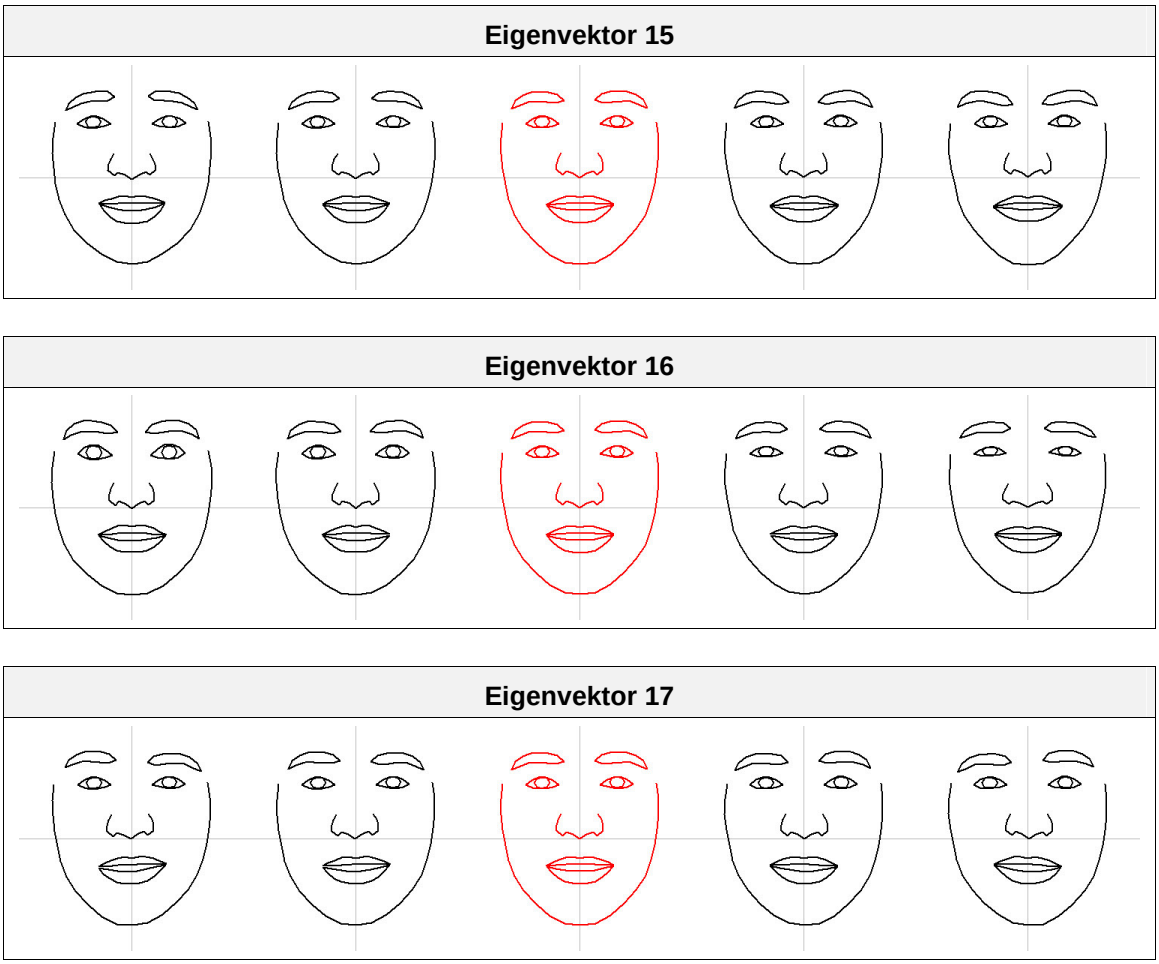


Bild C.1: ***Veränderung der Gesichtsform bei Variation der Eigenvektoren 11-17*** *im Bereich –2.0 ... +2.0 Standardabweichungen. Jeder Eigenvektor beschreibt charakteristische Formvariationen des mittleren Gesichts (Mitte).*

D Geräteübersicht

Kameras

JAI-2040 high resolution color CCD camera, ½´´ (Sequenzen in Bild 5.4 und Bild 5.11)

JAI CV-S3200 DSP surveillance color CCD camera, ½´´ (Sequenzen in Bild 5.8 und Bild 5.17)

Framegrabber

FALCON 32 bit low cost PCI Framegrabber

Rechner

Intel Pentium III Prozessor, 800 MHz Taktfrequenz, 256 MB Hauptspeicher, Betriebssystem: Microsoft Windows NT 4.0, Service Pack 5

Symbol- und Abkürzungsverzeichnis

β	Rückweisungsparameter der Farbklassifikation; $\beta \in [0,1]$
κ	radialer Verzerrungskoeffizient [$\frac{1}{mm^2}$]
λ_k	zum Eigenvektor p_k gehörender Eigenwert
$\vec{\mu}_k$	Mittelwert der Farbklasse Ω_k
Θ	Rotationswinkel
Ω_0	Rückweisungsklasse des Farbklassifikators
Ω_k	Farbklasse k; $k=1 \ldots n$
$\underline{b}_i$, $\overline{b}_i$	untere bzw. obere Grenze für b_i in Standardabweichungen
$\vec{b}$	Parametervektor – Eigenwerte in Vielfachen der Standardabweichung
b	Kamerakonstante
B	Grauwert des Blaukanals eines Farbbildes
b_i	zum Eigenvektor i gehörender Parameterwert
b_i^*	normierte Parameterwerte
B^n	B-Spline n-ter Ordnung
C	Kovarianzmatrix
CAD	Computer Aided Design
$\vec{c}$	m-dimensionaler Merkmalsvektor zur Beschreibung der Farbe
$\vec{c}_j$	Schwerpunkt der Kontur S_j
c_{dir}	Kostenfunktion der Gradientenrichtung
c_{mag}	Kostenfunktion der Gradientenamplitude
$c_{total}(\cdot)$	Gesamtkosten beim Intelligent Scissors Algorithmus
C_x, C_y	Koordinaten des Kamera-Hauptpunktes (Lot des Abbildungszentrums auf die Bildebene)
c_{ZC}	Kostenfunktion der Nulldurchgänge des Laplace-Operators

DP	dynamische Programmierung
E_i	Kante bei der inneren Modellstruktur
f(x)	Impulsantwort eines Kantendetektors
f_i	primäres Modellmerkmal für omp_i
$g_{mag}^{\max}$	maximaler Grauwert des Amplitude des Sobel-Bildes
$g_{edge}^{\max}$	maximaler Grauwert des Kantenbildes
g(p)	Grauwert des Pixels *p*
g(x)	Grauwertverlauf einer Kante
$g_0(x)$	Grauwertverlauf einer verrauschten Kante nach Glättung des Bildes
$g_{edge}(p)$	Grauwert des Pixels *p* im Kantenbild
G	Grauwert des Grünkanals eines Farbbildes
h(n)	Höhe des Konturteils *n*
HMM	Hidden Markov Model
i_1i_2	i_1i_2-Farbraum
img_{ZC}	Bild mit den Nulldurchgängen des Laplace-Operators für Bild *img*
$I_1I_2I_3$	$I_1I_2I_3$-Farbraum
KKS	Kamerakoordinatensystem
MMI	Mensch-Maschine Interaktion
MMK	Mensch-Maschine Kommunikation
o.b.d.A.	ohne Beschränkung der Allgemeinheit
OMP	Object Model Part (Objektmodellteil)
p_i	a priori Wahrscheinlichkeit für die Farbklasse Ω_k
$\vec{p}_k$	k-ter Eigenvektor eines Punktverteilungsmodells
$\vec{p}_w$	Vektor im Weltkoordinatensystem WKS
$\vec{p}_c$	Vektor im Kamerakoordinatensystem KKS
$\vec{p}_r$	Vektor in Pixelkoordinaten / Rechnerkoordinaten
p(·)	Wahrscheinlichkeitsdichte
P	Eigenvektormatrix

PCA	Principal Component Analysis
PDM	Point Distribution Model
$\mathbb{R}$	Menge der reellen Zahlen
R	Grauwert des Rotkanals eines Farbbildes
ROI	Region Of Interest
$r_{search}(n)$	Suchradius bei der Kantensuche für Konturteil n
s	Skalierungsfaktor für eine Kontur
s_i	3D-Szenenmerkmal *i*
s_x, s_y	horizontaler bzw. vertikaler Abstand der Pixel auf dem CCD-Chip
s(n)	Größe des Konturteils *n* (= $\sqrt{w(n) \cdot h(n)}$)
sf	*sign factor* für die Berücksichtigung des Gradientenvorzeichens bei der PDM Suche, sf$\in$ [0,1]
$sgn_{img}(p)$	Vorzeichen des Gradienten der Grauwerte im Bild in Suchrichtung, $sgn_{img}(p) \in \{-1,0,1\}$
sgn_{mod}	Soll-Vorzeichen des Gradienten der Grauwerte in Suchrichtung; mögliche Werte, $sgn_{mod} \in \{-1,0,1\}$
$\bar{S}$	mittlere Kontur eines PDM
S_i	einzelne Kontur
S_{edge}	Kontur, die sich aus den gefundenen Kantenpunkten im Bild ergibt (i.A. nicht konsistent mit dem PDM-Modell!)
S_{ini}	initiale Kontur bei der PDM-Suche
S_{res}	Ergebniskontur der PDM-Suche
S_{PDM}	Kontur im lokalen Modellkoordinatensystem
$\vec{t}$	Translationsvektor zwischen Position des Merkmals und Ursprung des lokalen Koordinatensystems eines Objektmodellteils
${}^{j-1}T_j$	homogene Transformationsmatrix zur Umrechnung vom lokalen Koordinatensystem des Objektmodellteils omp_j ins lokale Koordinatensystem von omp_{j-1}
$\vec{T}$	Translationsvektor zwischen KKS und WKS
V_i	Knoten bei der inneren Modellstruktur
w(n)	Breite des Konturteils *n*

w_{dir}	Gewichtung für c_{dir}
w_{mag}	Gewichtung für c_{mag}
w_{ZC}	Gewichtung für c_{ZC}
W	Gewichtsmatrix bei der Anpassung der Formparameter
WKS	Weltkoordinatensystem
$\overline{x_i x_{i+1}}$	Strecke zwischen den Punkten x_i und x_{i-1}
x_i, y_i	Koordinaten der Kontur
$x_{min}(n)$	minimale x-Koordinate des Konturteils *n*
$x_{max}(n)$	maximale x-Koordinate des Konturteils *n*
$y_{min}(n)$	minimale y-Koordinate des Konturteils *n*
$y_{max}(n)$	maximale y-Koordinate des Konturteils *n*

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Literaturverzeichnis

[AM79] B. D. O. Anderson, J. B. Moore: *Optimal Filtering*. In: Thomas Kailath (editor), „Information and System Science Series“, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA, 1-61, 1979.

[Asc02] http://www.ascension-tech.com/

[BH94] Baumberg, A., Hogg, D.: *Learning Flexible Models from Image Sequences*, Proc. ECCV (1), Lecture Notes in Computer Science 801, ed. J. O. Eklundh, Springer-Verlag, 299-308, 1994.

[BHG93] N. Badler, M. J. Hollick, J. P. Granieri: *Real-Time Control of a Virtual Human Using Minimal Sensors*. Presence: Teleoperators and Virtual Environments, 1(1), 82-86, 1993.

[BHU00] P. Brigger, J. Hoeg, M. Unser: *B-Spline Snakes: A Flexible Tool for Parametric Contour Detection*. IEEE Trans. Image Processing, 9(9), 1484-1496, 2000.

[BI98] A. Blake, M. Isard: *Active Contours*, Springer-Verlag, Berlin Heidelberg New York, 1998.

[BK00] M. Brand, V. Kettnaker: Discovery and Segmentation of Activities in Video. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), 844-851, 2000.

[BL97] J. Boyd, J. Little: *Global versus Structured Interpretation of Motion: Moving Light Displays*. In Proceedings of IEEE Nonrigid and Articulated Motion Workshop, 18-25, San Juan, Puerto Rico, Juni 1997.

[BM96] W. A. Barrett, E. N. Mortensen: *Interactive live-wire boundary extraction*. Medical Image Analysis 1(4), 331-341, 1996.

[BS96] I. N. Bronstein, K. A. Semendjajew: *Teubner-Taschenbuch der Mathematik*. Hrsg. E. Zeidler, B. G. Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1996.

[Can86] J. Canny: *A computational approach to edge detection*. IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6), 679-698, 1986.

[Cas96] K. R. Castleman: *Digital Image Processing*. PRENTICE HALL, New Jersey, New Jersey, 1996.

[CET99] T. F. Cootes, G. Edwards, C. J. Taylor: *Comparing Active Shape Models with Active Appearance Models*, Proc. British Machine Vision Conference (ed. T. Pridmore, D. Elliman), 1, 173-182, 1999.

[CG99] J. Cai, A. Goshtasby: *Detecting human faces in color images*. Image and Vision Computing 18, 63-75, 1999.

[CH00] I.-C. Chang, C.-L. Huang*: The model-based human body motion analysis system*. Image and Vision Computing, 18, 1067-1083, 2000.

[CHT94] T. F. Cootes, A. Hill, C. J. Taylor, J. Haslam: *The Use of Active Shape Models For Locating Structures in Medical Images*, Image and Vision Computing, 12(6), 355-366, 1994.

[CT93] T. F. Cootes, C. J. Taylor: *Active Shape Model Search using Local Grey-Level Models: A Quantitative Evaluation*, Proc. British Machine Vision Conference, (Ed. J. Illingworth), BMVA Press, 639-648, 1993.

[CTC92] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham: *Training Models of Shape from Sets of Examples*, Proc. British Machine Vision Conference, 9-18, Springer-Verlag, 1992.

[CTC95] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham*: Active Shape Models – Their Training and Application*. Computer Vision and Image Understanding, 61(1), 38-59, 1995.

[DGH98] Darrell, T., G. Gordon, M. Harville and J. Woodfill: *Integrated person tracking using stereo, color, and pattern detection*. Proc. IEEE Computer Vision and Pattern Recognition, Santa Barbara, CA, 601-609, 1998.

[DJD01] N. Duta, A. K. Jain, M.-P. Dubuisson-Jolly: *Automatic Construction of 2D Shape Models*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6), 433-446, 2001.

[EF78] P. Ekman and W.V. Friesen, *Facial Action Coding System (FACS): Manual*. Palo Alto: Consulting Psychologists Press, 1978.

[Fau93] O. Faugeras: *Three-Dimensional Computer Vision - A Geometric Viewpoint*. The MIT Press, Cambridge, 1993.

[Gav99] D. M. Gavrila: *The Visual Analysis of Human Movement: A Survey*, Computer Vision and Image Understanding, 73(1), 82-98, 1999.

[GL97] A. Gruen, H. Li: *Linear Feature Extraction with 3D LSB-Snakes*. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images, 287-298. Birkhäuser Verlag, Basel, 1995.

[GLC01] G. Guo, S. Z. Li, K. L. Chan: *Support vector machines for face recognition*. Image and Vision Computing, 19, 631-638, 2001.

[Haf99] W. Hafner: *Segmentierung von Video-Bildfolgen durch Adaptive Farbklassifikation*. Dissertation, Technische Universität München, 1998.

[Hal02] http://www.mvtec.com/halcon/

[HFP00] L. Herda, P. Fua, R. Plänkers, R. Boulic, D. Thalmann: *Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion,* Proc. of Computer Animation 2000, Philadelphia, IEEE Press.

[HFP01] L. Herda, P. Fua, R. Plänkers, R. Boulic, D. Thalmann: *Using Skeleton-Based Tracking to Increase the Reliability of Optical Motion Capture*, Human Movement Science Journal. In Press.

[HHD00] I. Haritaoglu, D. Harwood, L. S. Davis: *W^4: Real-Time Surveillance of People and Their Activities*. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), 809-830, 2000.

[HM96] W. Hafner, O. Munkelt: *Using Color for Detecting Persons in Image Sequences*. Pattern Recognition and Image Analysis, 7(1), 47-52, 1997.

[Hot33] H. Hotelling: *Analysis of a Complex of Statistical Variables into Principal Components*. J. Educ. Psychol., 24, 417-441,498-520, 1933.

[HWR01] F. Hülsken, F. Wallhoff, G. Rigoll: *Facial Expression Recognition with Pseudo-3D Hidden Markov Models*. 23rd DAGM Symposium, München, pp. 291-297, 2001.

[IT96] T. McInerney, D. Terzopoulos: *Deformable models in medical image analysis: a survey*. Medical image analysis, 1(2), 91-108, 1996.

[KP88] M. Katsikitis, I. Pilowsky: *A study of facial expression in Parkinson´s disease using a novel microcomputer-based method*. Journal of Neurology, Neurosurgery, and Psychiatry, 51, 362-366, 1988.

[KWT87] M. Kass, A. Witkin, D. Terzopoulos: *Snakes: active contour models*. International Journal of Computer Vision, 1(4), 321-331, 1987.

[Lan97] S. Lanser*: Modellbasierte Lokalisation gestützt auf monokulare Videobilder*. Dissertation, Technische Universität München, 1997.

[Len87] R. Lenz: *Linsenfehlerkorrigierte Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D-Messungen in Echtzeit*. In E. Paulus, Hrsg., Mustererkennung, Informatik-Fachberichte 149, 212-216. Deutsche Arbeitsgemeinschaft für Mustererkennung, Springer-Verlag, 1987.

[LK99] H.-K. Lee, J. H. Kim: *An HMM-Based Threshold Model Approach for Gesture Recognition*. IEEE Trans. Pattern Analysis and Machine Intelligence, 21(10), 961-973, 1999.

[LML00] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, A.

Baumgartner: *Automatic Extraction of Roads from Aerial Images Based on Scale Space and Snakes*. Machine Vision and Applications, 12(1), 22-31, 2000.

[LTC97] A. L. Lanitis, C. J. Taylor, T. F. Cootes: *Automatic Interpretation and Coding of Face Images Using Flexible Models*, IEEE Trans. Pattern Analysis and Machine Intelligence, 19(7), 743-756, 1997.

[LZB95] S. Lanser, Ch. Zierl, R. Beutlhauser: *Multibildkalibrierung einer CCD-Kamera*. In G. Sagerer, S. Posch and F. Kummert, editors, *Mustererkennung*, Informatik aktuell, 481-491. Deutsche Arbeitsgemeinschaft für Mustererkennung, Springer-Verlag, 1995.

[MB95] E. N. Mortensen, W. A. Barrett: *Intelligent Scissors for Image Composition*. ACM SIGGRAPH'95, LosAngeles, CA, 1995.

[MCB02] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, R. Harvey: *Extraction of Visual Features for Lipreading*. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(2), 198-213, 2002.

[MDK64] M. P. Murray, A. B. Drought, R. C. Kory: *Walking Patterns of Normal Men*. The Journal of Bone and Joint Surgery 46-A, 336-360, 1964.

[MG00] T. B. Moeslund, E. Granum: *Multiple Cues used in Model-Based Human Motion Capture*. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, Frankreich, März 2000.

[MG01] T. B. Moeslund, E. Granum: *A Survey of Computer Vision-Based Human Motion Capture*. Computer Vision and Image Understanding, 81, 231-268, 2001.

[MHB97] T. Molet, Z. Huang, R. Boulic, D. Thalmann: *An Animation Interface Designed for Motion Capture*. Proc. of Computer Animation, Genua, Mai 1997, ISBN 0-8186-7984-0, 77 – 85, IEEE Press.

[MN95] H. Murase, S. K. Nayar: *Visual Learning and Recognition of 3-D Objects from Appearance*. International Journal of Computer Vision, 14, 5-24, 1995.

[Moe99] T. B. Moeslund: *Summaries of 107 Computer Vision-Based Human Motion Capture Papers*. Technical Report, Laboratory of Image Analysis, Aalborg University, Dänemark, 1999.

[MSM90] S. Menet, P. Saint-Marc, G. Medioni : *"B-Snakes": Implementation and Application to Stereo*. Proc. Image Understanding Workshop, 720-726, 1990.

[Mun94] O. Munkelt: *Erkennung von Objekten in Einzelvideobildern mittels Aspektbäumen*. Dissertation, Technische Universität München, 1994.

[MZ99] Y. Matsumoto, A. Zelinsky: *Real-time Face Tracking System for Human-Robot Interaction*. Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics Conference (SMC'99), II-830-II-835, Tokyo, Japan, 12.-15. Oktober, 1999.

[NFI97] W. Neuenschwander, P. Fua, L. Iverson, G. Székely, O. Kübler: *Ziplock Snakes*. International Journal of Computer Vision, 25(3), 191-201, 1997.

[Nie83] H. Niemann: *Klassifikation von Mustern*. Springer-Verlag, 1983.

[OKS80] Y. Ohta, T. Kanade, T. Sakai: *Color Information for Region Segmentation*. Computer Graphics and Image Processing, 13(3), 222-241, 1980.

[PF01] R. Plänkers, P. Fua: *Tracking and Modeling People in Video Sequences*. Computer Vision and Image Understanding, 81(3), 285-302, 2001.

[PFA99] R. Plänkers, P. Fua, N. D´Apuzzo: *Automated Body Modeling from Video Sequences*. ICCV Workshop on Modeling People, Korfu, Griechenland, September 1999.

[PR00] M. Pantic, J. M. Rothkrantz: *Automatic Analysis of Facial Expressions: The State of the Art*. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(12), 1424-1445, 2000.

[PSH97] V. I. Pavlovic, R. Sharma, S. Huang: *Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review*. IEEE Trans. Pattern Analysis and Machine Intelligence, 19(7), 677-695, 1997.

[Rid99] C. Ridder: *Interpretation von Videobildfolgen zur Beobachtung artikularer Bewegung von Personen anhand eines generischen 3D Objektmodells*. Dissertation, Technische Universität München, 1999.

[RMK95] C. Ridder, O. Munkelt, H. Kirchner: *Adaptive Background Estimation and Foreground Detection using Kalman-Filtering*. In O. Kaynak, M. Özkan, N. Bekiroglu, I. Tunay, Hrsg., *Proceedings of International Conference on recent Advances in Mechatronics*, ICRAM'95, 193-199, Bogazici University, 80815 Bebek, Istanbul, Türkei, August 1995. UNESCO Chair on Mechatronics.

[RMR99] B. Radig, O. Munkelt, C. Ridder: *A Model-Driven Three-Dimensional Image-Interpretation System Applied to Person Detection in Video Images*. In B. Jähne, H. Haußäcker und P. Geißler, Hrsg. Handbook of computer vision and applications, Band 3: Systems and Applications, Kapitel 22. Academic Press, 1999.

[Roh93] K. Rohr: *Incremental Recognition of Pedestrians from Image Sequences*, Proc. Computer Vision and Pattern Recognition, 8-13, 1993.

[SC92] J. Shen, S. Castan: *An Optimal Linear Operator for Step Edge Detection*.

Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing, 54(2), 112-133, 1992.

[Sch94] R. Schuster: *Adaptive Modeling in Color Image Sequences*. DAGM Symposium, Wien, 161-169, 1994.

[Sch95] R. Schuster: *Objektverfolgung in Farbbildfolgen*. Dissertation, Technische Universität München, 1995.

[Sei94] A. Seidl: *Das Menschmodell RAMSIS – Analyse, Synthese und Simulation dreidimensionaler Körperhaltungen des Menschen*. Dissertation, Technische Universität München, 1994.

[SP95] T. Starner, A. Pentland: *Real-Time American Sign Language Recognition from Video Using Hidden Markov Models*. Technical Report TR-375, MIT Media Lab., 1995.

[Toy98] Toyama, K.: *Prolegomena for Robust Face Tracking*, Microsoft Research Technical Report MSR-TR-98-65, Vision Technology Group, Microsoft Research, Redmon, WA 98052, 1998.

[UM90] F. Ulupinar, G. Medioni: *Refining Edges Detected by a LoG Operator*. Computer Vision, Graphics, and Image Processing 51, 275-298, 1990.

[Uns99] Splines: *A Perfect Fit for Signal and Image Processing*, IEEE Signal Processing Magazine, 16(6), 22-38, 1999.

[Wac97] S. Wachter: *Verfolgung von Personen in monokularen Bildfolgen*. Vice Versa Verlag, Berlin, 1997.

[WAD96] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland: *Pfinder: Real-Time Tracking of the Human Body*. Technical Report No. 353, MIT Media Lab, 1996.

[XP98] C. Xu, J. L. Prince: *Generalized Gradient Vector Flow External Forces for Active Contours*. Signal Processing - An International Journal, 71(2), 131-139, December 1998.

[XPP00] C. Xu, D. L. Pham, J. L. Prince: *Image Segmentation Using Deformable Models*. SPIE Handbook on Medical Imaging - Volume III: Medical Image Analysis, edited by J.M. Fitzpatrick and M. Sonka, 2000.

[ZJD00] Y. Zhong, A. K. Jain, M.-P. Dubuisson-Jolly: *Object Tracking Using Deformable Templates*. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(5), 544-549, 2000.

Index

W

Z