Intuitive Interaktion durch videobasierte Gestenerkennung

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock

vorgelegt von

Cornelius Malerczyk, geb. am 09. August 1968 in Kiel

aus Hofheim am Taunus

Rostock, August 2009

urn:nbn:de:gbv:28-diss2010-0040-7

Gutachter:

Prof. Dr.-Ing. Bodo Urban	Universität Rostock
Prof. Dr. sc. techn. Oliver Staadt	Universität Rostock
Prof. Dr.-Ing. Dr. h.c. mult. José Luis Encarnação	TU Darmstadt

Datum der Verteidigung: 25. Januar 2010

Für meinen Vater †

Danksagung

Verschiedene Personen haben zum Gelingen dieser Arbeit beigetragen, und bei denen möchte ich mich an dieser Stelle ganz herzlich bedanken. Zuallererst gilt mein Dank Herrn Prof. Dr. Bodo Urban vom Fachgebiet Multimediale Kommunikation der Universität Rostock für die Möglichkeit, die Promotion bei ihm durchführen zu können.

Diese Arbeit wurde in der Abteilung Visual Computing am Zentrum für Graphische Datenverarbeitung e.V. in Darmstadt durchgeführt. Meinen Abteilungsleitern Herrn Michael Schnaider und Herrn Holger Graf bin ich zu großem Dank verpflichtet für die kontinuierliche Betreuung über die Jahre und die Schaffung einer Atmosphäre, in der es Spaß machte, an wissenschaftlichen Fragestellungen zu arbeiten. Zu diesem angenehmen Arbeitsumfeld trugen auch meine Arbeitskollegen bei, von denen ich mich speziell bei Dr. Bernd Schwald bedanken möchte. Des weiteren haben auch die von mir betreuten Diplomanden und studentischen Mitarbeiter ihren Anteil am Entstehen dieser Arbeit, wofür ich ihnen hiermit danken möchte.

Schließlich gilt mein ganz persönlicher Dank meiner Frau Martina und meinen Kindern Aaron und Tabea, die mich während der vielen Jahre unseres Zusammenlebens immer unterstützten und nicht nur dadurch ihren Beitrag zum erfolgreichen Abschluss dieser Arbeit geleistet haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Die Nutzung des Computers hat in den letzten Jahren und Jahrzehnten stetig zugenommen. Insbesondere im privaten Haushalt ist dieser Trend ungebrochen. Laut Statistischem Bundesamt nutzen heute beispielsweise über 80% der Deutschen einen Computer [Bun08]. Während noch im Jahre 2002 nur gut jeder zweite Haushalt (57%) über einen Personalcomputer verfügte, waren es 2007 bereits fast drei Viertel (73%) der Haushalte in Deutschland [Rad08]. Aber auch außerhalb der eigenen vier Wände begegnen einem Computersysteme immer häufiger. Oft sind jedoch die Systeme nicht sofort offensichtlich als Computer erkennbar, da die Verwendung eines Computers von den meisten Menschen direkt mit dessen Ein- und Ausgabegeräten assoziiert wird [Bux02], die klassischen Eingabegeräte wie Tastatur und Computermaus bei solchen Systemen aber oft nicht vorhanden sind. Gute Beispiele dafür sind die immer beliebter werdenden Navigationssysteme für das Auto oder auch Fahrscheinautomaten in den Bahnhöfen oder an den Bahnsteigen. Diese Systeme bedienen sich meist eines Tastbildschirms (engl.: *touch screen*) als gleichzeitiges Ein- und Ausgabegerät und ermöglichen so einer Vielzahl von Nutzern eine einfache Bedienung des Systems ausschließlich durch Drücken mit dem Finger auf entsprechende Schaltflächen auf dem Bildschirm. Interaktion mit Computersystemen, die an öffentlichen Plätzen verwendet werden sollen, müssen auch aus technischer Sicht jedermann zugänglich sein. Der Anwender muss entweder bekannte Technik vorfinden, die er deshalb auch sofort bedienen kann, oder aber die Eingabetechnik muss so einfach und intuitiv sein, dass fast jeder neue Nutzer das System ohne Bedienungsanleitung und ohne zeitaufwendige Trainingsphase verwenden kann. In den letzten Jahren hat sich deshalb die Technologie der Tastbildschirme als Eingabemedium für Computersysteme an öffentlich zugänglichen Orten weitestgehend durchgesetzt. Allerdings hat diese Technologie zwei entscheidende Nachteile: Zum einen ist die Bedienung des Bildschirms ausschließlich durch Drücken mit einem einzigen Finger möglich. Dadurch ist die Bandbreite der möglichen Anwendungen und deren Bedienung stark eingeschränkt. Aufwendige Menüstrukturen wie sie von klassischen Anwendungen auf dem Personalcomputer bekannt sind, sind nicht realisierbar. In der Regel beschränkt sich die Benutzerschnittstelle auf die Bedienung einer vorgegebenen Auswahl von Schaltflächen, die der Anwender durch

Abbildung 1.1: Klassische Eingabegeräte für die Interaktion mit VR-Welten: SpaceNavigator der Frima 3Dconnexion, CyberGlove II der Firma Immersion Corporation und Interaktiongerät für das Infrarot-Trackingsystem EOS [SM02].

Drücken mit dem Finger betätigen müssen, um in einer Baumstruktur von Menüpunkten zu navigieren. Zum anderen ist eine ergonomische Verwendung eines Tastbildschirms durch dessen Größe begrenzt. Der Anwender steht während der Interaktion in direktem Kontakt mit dem Bildschirm und hat damit auch einen sehr kleinen Abstand zum Ausgabegerät. Je größer die Ausgabefläche wird, desto größer muss aber auch der Abstand zum Bildschirm selber werden, damit der Anwender den Inhalt des Dargestellten noch vollständig erfassen kann. Immer preiswerter werdende Ausgabegeräte wie beispielsweise LCD- oder Plasma-Fernseher aber auch Projektionssysteme ermöglichen aber die einfache und kostengünstige Verwendung von großen Anzeigesystemen. Mit deren Verwendung werden deshalb neue Interaktionsformen und Eingabesysteme notwendig, die zum einen für große Anzeigesysteme geeignet sind und zum anderen auch den technisch unerfahrenen Benutzern die Möglichkeit bieten, schnell, einfach und intuitiv mit dem System zu interagieren.
Videobasierte Gestenerkennung kann hier einen entscheidenden Beitrag zur intuitiven Interaktion mit Computersystemen leisten. Zum einen bietet eine kamerabasierte Eingabe die Möglichkeit, dass der Anwender mit ausreichendem Abstand zum Ausgabegerät mit dem System interagieren kann, und so auch bei großen Anzeigesystemen den dargestellten Inhalt vollständig erfasst. Zum anderen ist der Nutzer nicht durch technische Hilfsmittel zur Eingabe beschränkt. Bei der Erkennung und Verfolgung von Handgesten muss weder der Umgang mit einem Eingabegerät wie beispielsweise einer Computermaus erlernt werden, noch ist der Anwender durch Kabel in seiner Bewegungsfreiheit eingeschränkt. Da Kameras außer Sicht des Anwenders positioniert werden können, erhöht sich insbesondere der immersive Charakter der Interaktion, da dann der Ausgabebildschirm das einzige technische Gerät ist, das für den Anwender sichtbar ist.

Eine weitere Anwendungsklasse ist die Interaktion mit virtuellen Welten. Für viele Computernutzer ist zwar der Umgang mit virtuellen dreidimensionalen Welten heute nicht mehr ungewöhnlich, die Interaktion selbst erfordert dennoch ein hohes Maß an Lernaufwand. Klassische Eingabegeräte für die Interaktion mit VR-Welten sind die Spacemouse, Datenhandschuhe und Infrarot-Trackingsysteme. Eine Spacemouse ist eine Erweiterung der klassischen zweidimensionalen Computermaus, bestehend aus einer Art Puck, der sich ziehen, drücken, kippen und drehen lässt und diese Bewegungen auf die Translation und Rotation in der 3D-

Abbildung 1.2: Klassische Eingabegeräte für Spielekonsolen: Gamepad für die Xbox 360, Nintendo Wii Remote mit Nunchuk-Erweiterung und Sony PlayStation Eye.

Welt abbildet (siehe Abbildung 1.1, links). Datenhandschuhe werden vom Anwender wie ein normaler Handschuh angelegt (siehe Abbildung 1.1, Mitte). Über beispielsweise faseroptische Kabel im Inneren des Handschuhs werden die Bewegungen der einzelnen Fingergelenke der Hand ermittelt, an den Rechner übertragen und zur Interaktion mit der virtuellen Welt verwendet. Infrarot-Trackingsysteme verwenden ein Stereo-Kamerasystem, um die Position und Orientierung von Infrarotlicht-reflektierenden Markern (siehe Abbildung 1.1, rechts) zu bestimmen [SM02].
Auch bei der Interaktion mit virtuellen Welten kann die videobasierte Gestenerkennung helfen, den Umgang mit dem Computer zu erleichtern, da die Verwendung von Handgesten für den Anwender natürlich ist und somit einfacher zu erlernen ist als herkömmliche Eingabemodalitäten. Um beispielsweise Gegenstände in einer dreidimensionalen virtuellen Szene zu bewegen, scheint es offensichtlich, den gewünschten Gegenstand im Virtuellen zu greifen, an eine andere Position zu bewegen und dort wieder abzusetzen. Für ein Gestenerkennungssystem bedeutet dies, dass es in der Lage sein muss, zwischen verschiedenen Gesten zu unterscheiden (beispielsweise die geschlossene und die offene Hand als Synonyme für das Greifen und Loslassen eines Objektes). Zusätzlich muss das System in der Lage sein, die Position der Hand in Echtzeit im dreidimensionalen Raum zu bestimmen und in die virtuelle Welt zu übertragen, damit der Anwender ohne Verzögerung die Reaktion des Systems erkennt.

Eine dritte Kategorie von Anwendungen mit einfachen und intuitiven Eingabegeräten bildet die Welt der Computerspiele und Spielekonsolen. Hier haben Interaktionsgeräte jenseits von Tastatur und Computermaus seit Jahren großen Erfolg. Während klassische Controller für Computerspiele die Interaktion über Steuertasten und Joysticks realisieren (siehe Abbildung 1.2, links), verwendet die 2006 erschienene Spielkonsole Nintendo Wii einen Controller, der mit einer Infrarotkamera und einem Beschleunigungssensor ausgestattet ist und so die Bewegungen und Drehungen des Controllers erfasst und für die Spielsteuerung nutzt (siehe Abbildung 1.2, Mitte). Eine Sonderstellung unter den Eingabegeräten für Spielekonsolen

stellt das 2003 erschienene EyeToy[1]-System für die Playstation der Firma Sony dar, da es das erste kommerziell erfolgreiche und als Massenprodukt vertriebene Eingabegerät ist, das videobasiert arbeitet und so die Bewegungen des Spieler zur Steuerung des Spiels verwendet. Das EyeToy-System ist eine USB-Kamera, die an die Spielekonsole angeschlossen wird (siehe Abbildung 1.2, rechts). Mittels bildverarbeitender Verfahren werden die Bewegungen des Spielers in Echtzeit analysiert und entsprechende Interaktionsereignisse für das Spielgeschehen generiert. Durch die Verwendung einer einzelnen Kamera, die direkt auf dem Fernseher als Ausgabegerät aufgestellt wird, ist die Interaktion jedoch auf rein zweidimensionale Interaktionsformen des ganzen Körpers des Spielers beschränkt.
Auch bei der Verwendung mit Computerspielen kann eine videobasierte Gestenerkennung helfen, die Interaktion mit dem Spiel zu vereinfachen oder zu verbessern. Durch die Verwendung eines Stereokamerasystems können beispielsweise dreidimensionale Ereignisse erzeugt werden und ermöglichen so nicht nur einen intuitiven Umgang mit den Anwendungen, sondern schaffen sogar die Möglichkeit, neue Spielideen zu entwickeln und umzusetzen.

1.2 Problemstellung und Zielsetzung

Videobasierte Gestenerkennung ist seit Langem ein intensives Forschungsthema im Bereich der Computergrafik. Dabei wird der Begriff Gestenerkennung für sehr unterschiedliche Teilbereiche des menschlichen Ausdrucks gesehen: Sie umfasst sowohl das Erkennen von Gesten des gesamten Körpers als auch insbesondere das Erkennen von Gesten der menschlichen Hand. Handgesten werden entweder als statisch (wie beispielsweise eine Zeigegeste) oder als dynamisch (wie beispielsweise Winken) klassifiziert. Eine Sonderstellung bei der Handgestenerkennung nimmt die Erkennung der Zeigegeste ein, da sie die vom Menschen wohl am häufigsten verwendete Handgeste ist und sie mit der Zeigerichtung außerdem kontextbezogene Zusatzinformation beinhaltet.
Während in den letzten Jahren eine Vielzahl von Verfahren zur Erkennung von Handgesten entwickelt und untersucht wurde, gibt es zu Verfahren, die für eine intuitive Interaktion zwischen Mensch und Computer (HCI) verwendet werden können, wenig Untersuchungen.
Wie das folgende Kapitel 2 zeigen wird, ist der aktuelle Stand der Forschung sehr stark geprägt durch die Entwicklung von Algorithmen und Verfahren, die es ermöglichen, Gesten der menschlichen Hand allgemein und vollständig zu erfassen. Ziel dieser Verfahren ist dabei insbesondere, Unterschiede der einzelnen Gelenkstellungen der Handknochen zu erkennen und so ein vollständiges virtuelles Abbild der realen Hand zu erzeugen. Da die menschliche Hand aber mit bis zu 27 Freiheitsgraden (vergleiche auch Abschnitt 2.3) ein hochkomplexes kinematische Modell ist, müssen in der meisten Verfahren starke Einschränkungen während der Erkennung in Kauf genommen werden. Da bei einer solch hohen Anzahl an Freiheitsgeraden meist sehr rechenzeitaufwendige Algorithmen eingesetzt werden müssen, akzeptieren viele veröffentlichte Verfahren beispielsweise, dass die Berechnungen zur Erkennung einer Geste nicht in Echtzeit zu erreichen ist. Damit wird aber eine Verwendung dieser Verfahren zur intuitiven Interaktion durch videobasierten Gestenerkennung verhindert, da nur durch eine sofortige Reaktion des Systems der immersive Charakter einer Anwendung gewahrt

[1] http://www.eyetoy.com/

wird. Andere oft getroffene Einschränkung beziehen sich auf den technischen Aufbau des Systems oder die verwendeten Hilfsmittel. Häufig werden spezielle Markierungen bis hin zu Handschuhen, die getragen werden müssen, gefordert, um eine robuste Erkennungsrate zu gewährleisten. Oder der Anwender des Gestenerkennungssystems muss aktiv auf die Position und Orientierung der Hand in Bezug auf die verwendete(n) Kamera(s) achten, damit zu jedem Zeitpunkt ein für das Verfahren gefordertes Erscheinungsbild der Hand im Kamerabild gewährleistet werden kann. Es ist offensichtlich, dass solche Einschränkungen eine intuitive Verwendbarkeit der Verfahren nicht gewährleisten können, da sich der Anwender nicht ausschließlich auf die Anwendung, mit der er interagieren möchte, konzentrieren kann, sondern immer wieder auch die korrekte Verwendung des Eingabemediums überprüfen muss.
Während einerseits viele in der wissenschaftlichen Literatur beschriebenen Verfahren durch die geforderten Einschränkungen eine intuitive Interaktion nicht oder nur erschwert zulassen, beschränken sich andererseits Verfahren, die den Anforderungen für eine einfache und intuitive Interaktion genügen, fast ausschließlich auf die Erkennung und Interpretation einer Zeigegeste. Zwar ist die Zeigegeste als Hilfsmittel für eine videobasierte Interaktion zwischen Mensch und Computer die wichtigste und am häufigsten verwendete Geste, dennoch sind die Anwendungsmöglichkeiten, die durch die ausschließliche Verwendung der Zeigegeste eröffnet werden, eingeschränkt. So ist es beispielsweise nicht möglich, intuitiv mit virtuellen Welten zu interagieren und Objekte im dreidimensionalen Raum zu greifen und an einer anderen Stelle wieder loszulassen.
Aus wissenschaftlicher Sicht bleibt somit die Frage nach einer intuitiven Interaktion zwischen Mensch und Computer durch videobasierte Gestenerkennung weitgehend unbeantwortet. Zum einen erfüllen die Verfahren nicht die Anforderungen für eine intuitive Interaktion aus technischer Sicht, weil beispielsweise die Forderung nach der Echtzeitfähigkeit des Systems nicht gewährleistet werden kann. Zum anderen erschweren die Verfahren eine Interaktion aus Anwendersicht, weil entweder Gesten, die der Anwender intuitiv verwenden möchte, nicht unterstützt werden oder weil der Anwender den Umgang mit dem Gestenerkennungssystem zunächst erlernen und später aktiv unterstützen muss.

Das Ziel dieser Arbeit besteht deshalb darin, Verfahren zu entwickeln, die es ermöglichen, durch die Verwendung von Handgesten intuitiv und einfach mit Computersystemen und deren Anwendungen zu interagieren. Dabei liegen die Anforderungen an die Verfahren für eine intuitive Verwendbarkeit auf der Hand:

- Die Verfahren dürfen keine direkten technischen Hilfsmittel für den Anwender fordern. Weder das Anlegen eines Handschuhs noch das Verwenden eines Gerätes, das der Anwender in die Hand nehmen muss, ermöglicht eine schnelle und intuitive Bedienung eines Systems. Die verwendeten Kameras sollten für den Anwender nicht offensichtlich erkennbar sein, um das immersive Erlebnis und damit die intuitive Interaktion nicht zu stören. Insbesondere sollte der Anwender sich nicht an die Gegebenheiten des Systems anpassen müssen und beispielsweise aktiv auf die Position und Ausrichtung der Kameras achten müssen. Vielmehr sollte das System in der Lage sein, auf unterschiedliche Anwender geeignet zu reagieren. Um den intuitive Charakter der Interaktion zu gewährleisten, muss zudem gefordert werden, dass der Anwender nicht

auf die Gestenerkennung selbst achten muss, sondern sich ausschließlich auf die Anwendung konzentrieren kann, mit der er interagieren möchte. Die Verwendung eines weiteren Bildschirms, auf dem die aufnehmenden Kamerabilder oder Anweisungen an den Benutzer eingeblendet werden, ist damit nicht möglich.

- Die Verfahren müssen mit minimalem Trainingsaufwand verwendbar sein. Von intuitiver Interaktion wird erwartet, dass ein neuer Nutzer eines Systems in der Lage ist, ganz ohne oder nur mit einer sehr kurzen Lernphase das System zu bedienen. Eventuelle Lernphase dürfen nur einmalig vor dem ersten Einsatz durchgeführt werden, um beispielsweise dem Verfahren zu ermöglichen, sich auf individuelle Unterschiede der Hand einzustellen. Dies kann ähnlich wie bei heutigen Spracherkennungssystemen realisiert werden, für deren Anwendung bei der ersten Verwendung zunächst ein kurzer Text vorgelesen werden muss. Für Verfahren, die an Orten verwendet werden, an denen viele unterschiedliche Menschen das System nur für eine kurze Zeit benutzen, darf keine Trainingsphase vorausgesetzt werden.

- Die Verfahren müssen echtzeitfähig sein. Das System muss in der Lage sein, Bildwiederholraten von 20 Bildern pro Sekunde oder mehr zu erreichen, um eine flüssig wirkende Interaktion zu ermöglichen. Außerdem muss die Verzögerung zwischen realem Ereignis der Hand und der daraus resultierenden Reaktion des Systems sehr kurz sein, da Reaktionszeiten von mehr als 200 Millisekunden von den meisten Menschen als unnatürlich empfunden werden [MN97, WWWR03].

Aufbauend auf diesen drei Systemanforderungen kann für die vorliegende Arbeit folgende Arbeitshypothese aufgestellt werden:

Hypothese 1 (Intuitive Interaktion) *Ausgehend vom derzeitigen Stand der Forschung und Technik ist es möglich, Verfahren zu entwickeln, die eine intuitive Interaktion zwischen Mensch und Computer ausschließlich durch die Verwendung von Handgesten ermöglichen und dabei die Forderungen nach geräteloser Bedienung, Reduktion des Trainingsaufwandes und Echtzeitfähigkeit erfüllen.*

Da in den letzten Jahren große Ausgabegeräte wie Fernseher und Projektoren für die Darstellung von Computeranwendungen immer beliebter geworden sind, liegt der Schwerpunkt dieser Arbeit nicht bei der intuitiven Steuerung von klassischen Desktop-Anwendungen, die sitzend vor einem normalen Monitor bedient werden, sondern bei Anwendungen, die auf einem großen Ausgabegerät dargestellt werden, vor dem der Anwender in der Regel mit einigem Abstand steht. Während der potentielle Raum für die Suche nach der menschlichen Hand bei der Verwendung eines Gestenerkennungssystems am Schreibtisch relativ klein ist, stellt sich hier allerdings die Frage nach einem schnellen und robusten Auffinden der interagierenden Hand in einem weitaus größeren Interaktionsraum. Aus technischer Sicht lassen

sich damit zwei Hauptprobleme identifizieren, die gelöst werden müssen, um eine intuitive Interaktion zu ermöglichen:

- Initialsuche der menschlichen Hand
  Die zu entwickelnden Verfahren müssen in der Lage sein, die Hand des Anwenders in dem Moment zu erkennen, wenn eine Interaktion startet. Trotz des großen Interaktionsraums muss dieser Schritt eines Verfahrens vollständig automatisch ablaufen, da dem Anwender keine direkte Kontrolle der Kamerabilder zur Verfügung steht, wenn der intuitive Charakter des Systems erhalten bleiben soll.

- Merkmalsextraktion und Klassifikation
  Die Verfahren müssen in der Lage sein, anhand der Bildeigenschaften die zur Interaktion notwendigen Informationen zu gewinnen. Im Falle einer Zeigegeste beispielsweise ist neben der Berechnung der Position der Hand auch die Bestimmung der Zeigerichtung im dreidimensionalen Raum zwingend erforderlich. Sollen mehrere Gesten verwendet werden dürfen, ist außerdem eine schnelle und robuste Unterscheidung der Gesten notwendig.

Für die Initialsuche der Hand sind in der Literatur im Wesentlichen zwei grundsätzlich unterschiedliche Ansätze beschrieben. Zum einem werden neben klassischen Verfahren der statistischen und syntaktischen Mustererkennung auch neuronale Netze und Optimierungsstrategien wie z.B. Genetische Algorithmen verwendet, um die Geste anhand von vorab bekannten Eigenschaften wie beispielsweise der Kontur zu finden. Allen Verfahren ist dabei gemein, dass sie sehr rechen- und zeitintensiv sind und sich deshalb nur bedingt für den Einsatz in einem Echtzeitsystem eignen. Zum anderen wird das einfache und robuste Verfahren der Hautfarbensegmentierung verwendet, um erste Schätzungen über die Position der Hand zu erlangen. Das häufig auftretende Problem, dass neben beiden Händen auch der Kopf des Anwenders als potentielle Gestenregion erkannt wird, muss dann in einem folgenden Klassifikationsschritt gelöst werden.
Für die Klassifikation unterschiedlicher Gesten und der für die Interaktion notwendigen Extraktion der entsprechenden Merkmale sind in der Literatur eine Vielzahl von unterschiedlicher Verfahren beschrieben. Neben Hidden Markov Modellen, Neuronalen Netzen und der Analyse von Merkmalsvektoren werden auch Aktive Konturen und klassische Verfahren der Mustererkennung eingesetzt. Die Entscheidung, welches Verfahren für das jeweiliges System zum Einsatz kommt, wird dabei in der Regel durch die jeweils angenommenen Einschränkungen wie Echtzeitfähigkeit oder Kameraaufbau getroffen.

Der Inhalt dieser Arbeit setzt sich mit neuen Verfahren zur videobasierten Handgestenerkennung auseinander, welche die oben genannten Bedingungen zur intuitiven Interaktion zwischen Mensch und Computer erfüllen und Handgesten in Echtzeit im dreidimensionalen Raum erfassen und verfolgen können. Die Grundlage für die Erkennung von Handgesten im 3D-Raum bildet dabei die Verwendung eines kalibrierten Stereokamerasystems, weshalb die dafür grundlegenden Themen wie Kalibrierung der Kameras und 3D-Rekonstruktion von korrespondierenden Bildpunkten dargestellt werden sollen. Neben der theoretischen Entwicklung

der Verfahren soll auch der direkte Bezug zu praktischen Anwendungen erläutert werden. Insbesondere die durch unterschiedliche Anwendungen vorgegebenen Anforderungen an die Verfahren sollen behandelt werden.

1.3 Organisation der Arbeit

Die vorliegende Arbeit unterteilt sich in drei übergeordnete Einheiten. In den einleitenden Kapiteln 1 bis 4 wird das Thema der videobasierten Gestenerkennung grundsätzlich behandelt. Es werden die zum Verständnis notwendigen Begriffe erklärt und der Stand der Technik erläutert. Die darauf folgenden Kapitel 5 bis 8 beschreiben detailliert die in dieser Arbeit entwickelten Verfahren zur intuitiven Interaktion durch videobasierte Gestenerkennung. Die Arbeit schließt in den Kapiteln 9 und 10 mit praktischen Anwendungen der entwickelten Verfahren und einem Ausblick auf weitere Arbeiten. Im Einzelnen haben die Kapitel den folgenden Inhalt:
Kapitel 2 führt in die videobasierte Gestenerkennung ein und gibt einen Überblick über existierende Ansätze, Verfahren und Systeme zur Erkennung und Verfolgung von Handgesten.
Kapitel 3 beschreibt die Kalibrierung eines Stereo-Kamerasystems und die dreidimensionale Rekonstruktion von korrespondierenden Bildpunkten und liefert damit die technischen Grundlagen für die in den folgenden Kapiteln 4 bis 7 entwickelten Verfahren zur Gestenerkennung.
Die Kapitel 4 und 5 beschreiben Verfahren zur Erkennung und Verfolgung einer Zeigegeste. Dabei verwendet das in Kapitel 4 erläuterte Verfahren *Aktive Formen*, um die Silhouette der Hand auf Bildebene anzupassen und in einem weiteren Schritt die Position der Hand und die Zeigerichtung im dreidimensionalen Raum zu bestimmen.
Kapitel 5 beschreibt ein Verfahren zur dreidimensionalen Bestimmung der Fingerspitze in einer Zeigegeste und deren Projektion auf die Ausgabefläche des Bildschirms zur Bestimmung der Zeigerichtung.
In Kapitel 6 wird ein Verfahren entwickelt, das den statistischen naiven Bayes-Klassifikator verwendet, um mittels Bildsegment-Informationen unterschiedliche Handgesten in Echtzeit zu unterscheiden und so neben der Position der Hand im dreidimensionalen Raum auch zusätzliche Informationen über die verwendetet Handgeste für die Interaktion bereit stellt.
Kapitel 7 beschreibt ein neues Verfahren zur Bestimmung von dynamischen Fingerbewegungen wie beispielsweise dem Pinzettengriff, also dem Zugreifen und Loslassen von Objekten mit Daumen und Zeigefinger anhand von Bildsegment-beschreibenden *Pseudo-Zernike-Momenten*. Zur Beschleunigung des Verfahrens wird der Suchraum durch eine Diskretisierung des dynamischen Prozesses der Geste in einzelne statische Handposen erheblich eingeschränkt. Die Suche nach den von den kalibrierten Kameras gesehenen Handposen wird dabei über einen Vergleich mit zuvor künstlich erzeugten Silhouetten eines virtuellen Handmodells und deren vorberechneten statistischen Momenten realisiert. Durch zugrunde liegende Skelettinformationen der vordefinierten Handposen ist eine standardisierte Beschreibung der Geste gewährleistet, die insbesondere für eine intuitive Interaktion mit virtuellen dreidimensionalen Welten von Vorteil ist.

Im Anschluss vergleicht Kapitel 8 die Anforderungen und Ergebnisse der zuvor vorgestellten Verfahren und bewertet diese in Bezug auf die Voraussetzungen zur intuitiven Interaktion und die Verwendbarkeit der Algorithmen in unterschiedlichen Anwendungsszenarien. In einer Usability-Studie wird die Akzeptanz des Verfahren der *Interaktion durch Punktprojektion* bei mehr als 80 Anwendern evaluiert.
Zum Abschluss des Inhalts dieser Arbeit folgt Kapitel 9, in dem ausgewählte Anwendungen beschrieben werden, in denen die in dieser Arbeit entwickelten Verfahren verwendet werden. Eine Zusammenfassung der Arbeit sowie ein Ausblick findet sich abschließend in Kapitel 10.

1.4 Zusammenfassung der wichtigsten Ergebnisse

Diese Arbeit befasst sich mit den in Abschnitt 1.2 aufgezeigten und bislang ungelösten Problemen einer intuitiven Interaktion durch videobasierte Gestenerkennung. Ziel ist es, Verfahren zu entwickeln, die die existierenden Probleme bei der automatischen Initialsuche nach einer Geste und bei der Merkmalsextraktion und Klassifikation von Handgesten unter den Bedingungen für einen einfachen und intuitiven Umgang mit einem Gestenerkennungssystem ermöglichen. Die Verfahren unterstützen die Forderungen nach intuitiver Interaktion, indem auf technische Hilfsmittel verzichtet wird, Interaktion ohne oder nur mit minimalem Trainingsaufwand stattfinden kann und indem die Echtzeitfähigkeit der Verfahren gewährleistet wird. Damit können wesentliche Probleme der videobasierten Erkennung von Handgesten gelöst werden, um eine intuitive Interaktion zwischen Mensch und Computer zu ermöglichen.
Im Rahmen dieser Arbeit werden vier unterschiedliche Verfahren entwickelt und untersucht, die eine intuitive Interaktion mit Computersystemen und deren Anwendungen durch videobasierte Gestenerkennung ermöglichen. Je nach Anwendungstyp und den daraus resultierenden Anforderungen an die Eingabemodalitäten des Systems können die unterschiedlichen Verfahren variabel eingesetzt werden und bestehende Verfahren oder vorhandene Eingabegeräte ersetzen. Die in dieser Arbeit vorgestellten Verfahren bedienen sich eines kalibrierten Stereo-Kamerasystems und ermöglichen so die Erkennung und Verfolgung von Handgesten im dreidimensionalen Raum. Die Verwendung von Kameras zur gestenbasierten Interaktion unterstützt dabei den intuitiven Charakter der Eingabe, da der Nutzer kein technisches Gerät bedienen muss, sondern ausschließlich mit seinen eigenen Händen agieren kann. Da die verwendeten Kameras, Kabel und auch der Rechner selbst außerhalb der direkten Sicht des Anwenders positioniert werden kann, ist damit in der Regel der Ausgabebildschirm das einzige technische Gerät, das für den Nutzer sichtbar ist. Schnelle Reaktionszeiten des Systems unter 200 Millisekunden, kurze Verarbeitungszeiten pro Bildpaar von weniger als 50 Millisekunden und eine permanente visuelle Rückantwort des Systems garantieren bei allen Verfahren die Möglichkeit eines Einsatzes für Echtzeit-Anwendungen.
Die ersten beiden in dieser Arbeit vorgestellten Verfahren sind in der Lage, insbesondere die Zeigegeste der menschlichen Hand als wichtigste statische Geste zur Interaktion zu erkennen und zu verfolgen. Beide Verfahren adressieren insbesondere das Problem einer automatischen Initialsuche nach der Handgeste, um sicherzustellen, dass der Anwender mit der

Interaktion ohne aktives Eingreifen beginnen kann. Die Verfahren der *3D-Rekonstruktion von Aktiven Formen* und der *3D-Punktprojektion* ermöglichen eine präzise Berechnung sowohl der Position der Hand im dreidimensionalen Raum als auch des Interaktionspunktes auf dem Ausgabebildschirms in Echtzeit. Beide Verfahren unterstützen so die intuitive Steuerung sowohl von Anwendungen, die klassische deiktische Eingabemodalitäten wie eine Computermaus verwenden als auch von multimodalen Anwendungen, welche die Kombination von beispielsweise Gesten- und Spracherkennung als Eingabemodalitäten unterstützen.
Mit den so gewonnenen Erkenntnissen über eine echtzeitfähige Initialsuche nach der interagierenden Handgeste wird in einem weiteren Verfahren das Problem der Verwendbarkeit von verschiedenen Gesten zur intuitiven Interaktion behandelt. Das Verfahren der *Merkmalbasierten Gesten-Klassifikation* ermöglicht neben der Berechnung der Position der Handgeste im dreidimensionalen Raum auch eine Unterscheidung verschiedener Handgesten in Echtzeit. Damit wird insbesondere die Interaktion mit virtuellen Welten erleichtert, da der Anwender beispielsweise durch die Unterscheidung einer geschlossenen und einer offenen Hand Gegenstände im Virtuellen aufnehmen und an einer anderen Stelle des 3D-Raums wieder ablegen kann.
Das letzte der vier hier vorgestellten Verfahren beschäftigt sich mit dem Problem einer echtzeitfähigen Merkmalsextraktion, um dynamische Prozesse einer Geste in Echtzeit bestimmen und für die Interaktion nutzbar machen zu können. Das Verfahren der *Momenten-Analyse* liefert als Rekonstruktionsergebnis neben der Position im 3D-Raum auch die Orientierung der Hand und alle den Handposen zugrunde liegenden Gelenkwinkel. Mit der Möglichkeit, dynamische Prozesse der Hand wie beispielsweise dem Zugreifen und Loslassen von Objekten mittels Daumen und Zeigefinger (Pinzettengriff) in feinen Abstufungen der durchgeführten Bewegungen zu erkennen, lassen sich insbesondere feinmotorische Prozesse der Hand rekonstruieren und für die intuitive Interaktion mit dreidimensionalen virtuellen Welten verwenden.

Die Umsetzung und die Verwendbarkeit der in dieser Arbeit entwickelten Verfahren wird in verschiedenen Anwendungsgebieten erfolgreich nachgewiesen. Im musealen Umfeld stehen beispielsweise Anwendungen zur Untersuchung von digitalisierten zwei- und dreidimensionalen Kunstwerken zur Verfügung [MS03a, SM03, SM04, MS04, Mal04, MDS05a, MDS05b]. Im häuslichen Umfeld können technische Geräte durch Gestenerkennung gesteuert werden [SMS01, Mal08a, Mal08b] und an öffentlichen Orten mit vielen unterschiedlichen Nutzern ist Gestenerkennung beispielsweise zur Steuerung eines Informations-Kiosksystems nutzbar [MSG03, Mal08b].

Kapitel 2

Videobasierte Gestenerkennung

Dieses Kapitel gibt einen Überblick über den aktuellen Stand der Forschung und Technik zu videobasierten Gestenerkennungssystemen. Da es zur Zeit noch keine universellen Verfahren zur videobasierten Interaktion mit dem Computer mittels Gestenerkennung gibt, werden neben einer Einteilung existierender Verfahren auch Schwierigkeiten und Herausforderungen bei der Erkennung von Handgesten untersucht. Für die wichtigsten Einzelschritte zur Gestenerkennung wie beispielsweise der Bestimmung der Position der Hand, der Merkmalsbestimmung und der Klassifikation unterschiedlicher Gesten werden die in der Literatur beschriebenen Verfahren und Methoden untersucht und vorgestellt. Aus dieser Untersuchung werden zwei neuartige Taxonomien entwickelt, die als Grundlage für die Untersuchungen und Entwicklungen in den weiteren Kapiteln dienen. Neben den wichtigsten in der Literatur beschriebenen Verfahren zur Handgestenerkennung werden abschließend einige bereits auf dem Markt existierende, kommerzielle Systeme vorgestellt, die eine einfache Interaktion zwischen Mensch und Computer ermöglichen.

2.1 Einleitung

Gestenerkennung ist seit Langem ein intensives Forschungsthema im Bereich der Computergrafik. Dabei wird der Begriff Gestenerkennung für sehr unterschiedliche Teilbereiche des menschlichen Ausdrucks gesehen: Sie umfasst sowohl das Erkennen von Gesten des gesamten Körpers [Pop07, MHK06, WHT03, Gav99] als auch insbesondere das Erkennen von Gesten der menschlichen Hand [EBN+07, MA07]. Handgesten werden entweder als statisch (wie beispielsweise eine Zeigegeste [STTC06, SMC01, CBV03]) oder als dynamisch [ELD01] (wie beispielsweise Winken) klassifiziert. Verfahren zur Erkennung dynamischer Bewegungen der Hand und deren Bedeutung spielen für eine direkte Interaktion zwischen Mensch und Computer nur eine untergeordnete Rolle und werden daher in dieser Arbeit nicht weiter untersucht. Eine Sonderstellungen bei der Handgestenerkennung nimmt die Erkennung der Zeigegeste ein, da sie als einfache und intuitive Schnittstelle zwischen Mensch und Computer verwendet werden kann, um sowohl mit virtuellen Welten zu interagieren als auch Anwenderprogramme zu steuern [CBV03, SLM+03]. In dieser Arbeit wird der Begriff *Gestenerkennung*

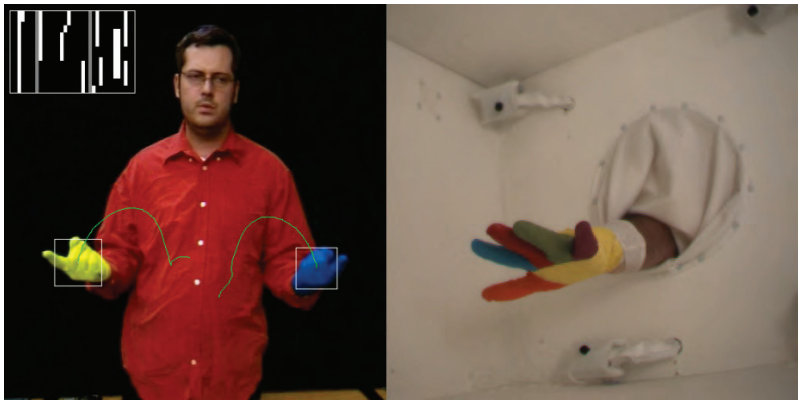

Abbildung 2.1: Verwendung von farblich markierten Handschuhen zur Handgestenerkennung. Unterscheidung zweier Hände [BWK+04], links und Trennung einzelner Finger durch Farbmarkierungen [UEB+06], rechts.

daher ausschließlich für die Erkennung und Verfolgung von Posen der menschlichen Hand verwendet.

Die ausschließliche Verwendung der Hand als Eingabemodalität ist eine attraktive Methode, um die Schnittstelle zwischen Mensch und Computer einfach und intuitiv zu gestalten. Um aber eine hohe Akzeptanz insbesondere bei technisch unversierten Nutzern zu erreichen, gelten die im vorigen Kapitel beschriebenen Anforderungen bezüglich Echtzeitfähigkeit, Trainingsaufwand und der Verwendung von technischen Hilfsmitteln. Die Bedingung der Echtzeitfähigkeit ist bei den in vielen Systemen verwendeten Datenhandschuhen zusammen mit einer hohen Genauigkeit der Handpose gegeben. Datenhandschuhe werden zum Erfassen von Handgesten verwendet, indem ein Handschuh mit meist elektromagnetischen Sensoren an den Gelenken der Hand bestückt ist, welche die Orientierung der einzelnen Gelenke erfassen und somit Orientierung und Pose der Hand an ein auswertendes System übertragen [FI02, SZ94]. Allerdings verhindern sowohl die Verkabelung des Handschuhs als auch lange und aufwendige Kalibrierungsprozeduren ein schnelles und einfaches Interagieren mit der eigentlichen Anwendung. Um zumindest die Kabel und damit die eingeschränkte Bewegungsfreiheit des Anwenders zu eliminieren, verwenden einige Ansätze farblich markierte Handschuhe ohne aktive Sensoren (siehe Abbildung 2.1). Solche Handschuhe werden sowohl für eine einfache Unterscheidung der beiden Hände des Anwenders bei dynamischen Gesten [BWK+04] (siehe Abbildung 2.1, links) als auch für eine detailierte Rekonstruktion von Handposen [UEB+06] (siehe Abbildung 2.1, rechts) verwendet.

Durch immer leistungsstärkere Computerhardware und immer preiswerter werdende Kamerasysteme sind die Möglichkeiten einer videobasierten und damit für den Anwender gerätelosen Interaktion mit dem Computer in den vergangenen Jahren immer einfacher realisierbar geworden. Dennoch fordern die meisten Gestenerkennungssysteme für eine korrekte Erkennung der Gesten Einschränkungen in Bezug auf die herrschenden Umgebungsbedingungen

wie z.B. die Verwendung von uniformen und zumindest unbewegten Hintergründen oder aber auch die Forderung nach konstanten Lichtverhältnissen. Eines der wichtigsten Kriterien zur tatsächlichen Anwendbarkeit eines Gestenerkennungssystems ist seine Echtzeitfähigkeit. Nur wenn das System in der Lage ist, Gesten unmittelbar zu erkennen und darauf zu reagieren, ist ein System als einfach und intuitiv anwendbares Interaktionsmedium zwischen Mensch und Computer geeignet.

2.2 Begriffsdefinitionen

Sowohl in der wissenschaftlichen Literatur als auch umgangssprachlich wird der Begriff der **Geste** oft unterschiedlich verwendet. Aus diesem Grund sollen an dieser Stelle zunächst die für diese Arbeit relevanten Begriffe definiert und erläutert werden. Grundsätzlich ist eine Geste immer als symbolische und nonverbale Kommunikationsform aufzufassen. Gesten können mit verschiedenen Körperteilen (Hand, Kopf oder Augen für beispielsweise Winkeln, Nicken oder Augenzwinkern) oder auch dem ganzen Körper (z.B. Verbeugung oder Kniefall) ausgeführt werden. Zunächst spielt es keine Rolle, ob eine Geste als Kommunikation zwischen zwei Menschen oder zwischen einem Mensch und einer Maschine wie beispielsweise einem Computer stattfindet. Unter **Gestenerkennung** versteht man im Allgemeinen eine automatische Erkennung einer vom einem Menschen ausgeführten Geste durch ein spezielles Computerprogramm. Da sich diese Arbeit ausschließlich mit der Kommunikation zwischen Mensch und Computer mittels **Handgesten** beschäftigt, werden die Begriffe *Geste* und *Handgeste* hier synonym verwendet. Der wichtigste Verfeinerungsschritt des Gestenbegriffs ist die Unterteilung in **statische** und **dynamische** Gesten.

- **Statische Handgesten** beschreiben eine starre Ausprägung des kinematischen Modells der menschlichen Hand. Dabei spielt die Position und die Orientierung der Hand im dreidimensionalen Raum keine Rolle. Entscheidend ist vielmehr, dass sich während einer statischen Geste die Gelenkwinkel innerhalb der Hand nicht oder zumindest nicht wesentlich ändern. Das wohl bekannteste Beispiel einer statischen Handgeste ist die **Zeigegeste**, für die es allerdings sehr unterschiedliche, individuelle Ausprägungen und Interpretationsarten gibt. Während ein Anwender lediglich den ausgestreckten Zeigefinger (mit oder ohne abgespreizten Daumen) verwendet, um auf eine Position zu deuten, verwendet ein anderer Anwender die vollständig geöffnete Hand zum Zeigen. Analog zu den umgangssprachlichen Begriffen der *Körperpose* und der *Positur*, also einer starren und betonten Körperhaltung, werden statische Handgesten demnach auch als **Handposen** bezeichnet und die beiden Begriffe bedeutungsgleich verwendet. Damit unterscheidet sich der Begriff der Pose in dieser Arbeit grundsätzlich von dem oft in der Technik verwendeten und durch einen Standard[1] definierten Begriff der Pose, die als Kombination von Position und Orientierung eines Objektes (z.B. eines Roboterarms) im dreidimensionalen Raum aufgefasst wird.

[1] DIN EN ISO 8373: Industrieroboter Wörterbuch

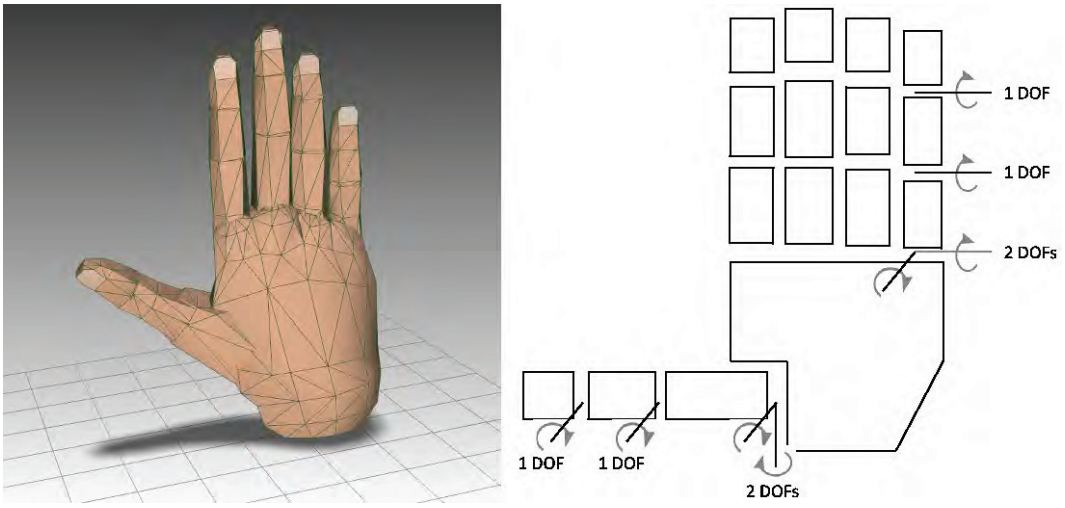

Abbildung 2.2: VRML-Handmodell mit knapp 1600 Polygonen (links) und Skizze der möglichen Gelenkrotationen mit internen zwanzig Freiheitsgraden.

- Der Begriff der **dynamischen Handgeste** wird für zwei unterschiedliche Arten von Gesten verwendet. Zum einen beschreibt eine dynamische Handgeste die Interpretation der Bewegung einer (starren) Handpose im Raum. Ein Beispiel hierfür ist die Geste des Winkens, wobei die geöffnete flache Hand vor dem Körper als Gruß aus der Distanz hin und her geschwenkt wird. Zum anderen kann eine definierte Bewegung der Fingergelenke (oft auch ungeachtet der Position und Orientierung der Hand im Raum) als dynamische Geste aufgefasst werden. Ein typisches Beispiel für eine solche dynamische Handgeste ist die sogenannte *Zitier-Geste*, bei der Zeige- und Mittelfinger beider Hände synchron gebeugt und gestreckt werden, um die bei Sprechen nicht sichtbaren Anführungszeichen eines Zitats zu imitieren. In dieser Arbeit werden dynamische Gesten mit einer Änderung der Finger- und Daumengelenkwinkel auch als **dynamischer Prozess einer Handgeste** bezeichnet.

Die in dieser Arbeit verwendeten Begriffe sollen nun noch einmal anhand eines einfachen Beispiels zusammengefasst werden: Eine geöffnete oder eine geschlossene Hand wird als **statische Handgeste** oder **Handpose** bezeichnet. Das Öffnen oder das Schließen der Hand durch Strecken und Krümmen der Finger und des Daumens wird als **dynamische Handgeste** oder als **dynamischer Prozess einer Handgeste** bezeichnet.

2.3 Handmodelle

Eine geeignete Beschreibung und Darstellung der menschlichen Hand spielt sowohl für die Erkennung und Verfolgung der Hand als auch für die Visualisierung von rekonstruierten Handposen eine entscheidende Rolle. Während für die videobasierte Erkennung eher auf

Merkmale der Oberfläche und das zugrunde liegende kinematische Modell geachtet werden muss, steht für eine realistische Darstellung der Rekonstruktionsergebnisse mehr das Oberflächenmodell mit seinen Deformationsmöglichkeiten im Vordergrund. Die menschliche Hand ist ein komplexes Objekt mit einer hohen Anzahl an beweglichen Gelenken und damit an zu berücksichtigenden Freiheitsgraden. Neben der Position und Orientierung im dreidimensionalen Raum (6 Freiheitsgrade, DOF) besteht die Hand aus insgesamt 15 für Handposen relevanten, verschiedenen Gelenken. Trotz biomechanischer Restriktionen der Gelenke [VCJT03] ergibt sich daraus eine hohe Anzahl von Freiheitsgraden des Handskeletts. Die tatsächlich verwendete Anzahl der Freiheitsgrade wird in der Literatur unterschiedlich behandelt. Um die volle Bewegungsfreiheit der Hand zu beschreiben, sind Modelle mit 20 [KH95] und mit 21 Freiheitsgraden [GT05] gebräuchlich. Unterschiedlich ist hier nur die Frage, ob das zweite Gelenk des Daumens mit einer einzigen oder mit zwei Rotationsachsen bewertet wird (siehe Abbildung 2.2, rechts). Zusammen mit der Position und Rotation des Handgelenks und damit der gesamten Hand im Raum ergibt sich somit eine maximale Anzahl von 27 Freiheitsgraden. Ausführliche Analysen der Bewegungsfreiheit des Handskeletts für die Rekonstruktion und die Animation der menschlichen Hand sind in [NHOD07] und [AHS03] zu finden. Ein bildbasiertes Verfahren zur Modellierung von individuellen und texturierten Handgeometrien ist in [RNL06] beschrieben.

2.4 Taxonomie für Gestenerkennungssysteme

Die in der Literatur beschriebenen Taxonomien für videobasierte Gestenerkennungssysteme sind fast ausschließlich technisch geprägt und teilen Systeme und Verfahren bezüglich der verwendeten Hardware, den verwendeten Algorithmen oder vorausgesetzten Einschränkungen ein. Da in dieser Arbeit die intuitive Interaktion für den Anwender im Vordergrund steht, soll neben der Auflistung von klassischen technischen Einteilungen im nächsten Abschnitt auch eine neue anwenderbezogene Taxonomie für Gestenerkennungssysteme entwickelt werden. Diese Einteilung geht zunächst vom Nutzer des Systems und der Anwendung, mit der er interagiert aus und leitet erst in einem zweiten Schritt die technischen Anforderungen an das jeweilige System ab.

2.4.1 Einteilung aus technischer Sicht

Typische technisch geprägte Einteilungen unterscheiden Gestenerkennungssysteme anhand einer Vielzahl von unterschiedlichen Parametern. Im Folgenden sollten die drei gebräuchlichsten Themengebiete, die für die Klassifikation verwendet werden, aufgelistet und näher erläutert werden.

- Verwendete Hardware

  Optische Systeme können nach der Anzahl der Kameras eingeteilt werden. Monokamerasysteme verwenden eine einzige Kamera für die Analyse des eingehenden Bildstroms. In der Regel ist damit nur eine Klassifikation unterschiedlicher statischer Po-

Abbildung 2.3: Erkennung von Buchstaben der Gebärdensprache *American Sign Language*, ASL. Das System verwendet ein Monokamerasystem und Farbinformation der Hand, um die Hand und deren Kontur vom Hintergrund zu trennen [BS02].

sen möglich. Für eine Schätzung der Position der Hand im Raum sind desweiteren Einschränkungen notwendig. Oft wird gefordert, dass die Kamera orthogonal zur Interaktionsfläche ausgerichtet ist und die Hand nur bedingt im Raum vor der Kamera rotiert wird. Ein typischer Fall für ein Monokamerasystem ist die Erkennung und Interpretation von Gebärdensprache wie beispielsweise der *American Sign Language*, ASL [BS02, SP95]. Stereokamerasysteme verwenden zwei oder mehr kalibrierte Kameras und ermöglichen eine dreidimensionale Rekonstruktion von korrespondierenden Bildpunkten. Ein weiteres Entscheidungskriterium ist die Art der Bildinformationen, die von den Kameras geliefert werden. Farbkameras ermöglichen eine einfache Segmentierung von Bildbereichen, in denen Hautfarbe zu erkennen ist [HSW08, VSA03]. Durch die Verwendung von Farbinformation ist hier allerdings das Datenaufkommen dreimal so hoch wie bei der Verwendung von Graustufenkameras, die Luminanzbilder liefern. Ohne die Möglichkeit, Farbinformationen zur Hautfarbensegmentierung zu verwenden, wird oft die Szene mit zusätzlichem Licht beleuchtet, um eine Trennung von Hand und Hintergrund zu erleichtern. Eine zusätzliche Beleuchtung im Nah-Infrarot-Bereich hat dabei zusätzlich den Vorteil, dass der Interaktionsraum abgedunkelt werden und so der Kontrast der Anwendung auf dem darstellenden Ausgabegerät erhöht werden kann [CBV03, Mal08b].

- Verwendete Algorithmen

  Eine Einteilung nach den hauptsächlich verwendeten Methoden und Algorithmen ist wohl die am häufigsten verwendetet Klassifikation von Gestenerkennungssystemen. Einen Überblick über Körper-, Gesichts- und Handgestenerkennungsmethoden gibt beispielsweise [MA07]. Die wichtigsten in der Literatur beschriebenen Verfahren zur Erkennung und Klassifizierung unterschiedlicher statischer und dynamischer Gesten sind dabei Hidden Markov Modelle, Neuronale Netze und Merkmalsvektoren. Diese drei wichtigen verfahren werden in den folgenden Abschnitten genauer untersucht. Daneben sind aber auch andere Methoden in der Literatur beschrieben, wie beispielsweise der *Condensation Algorithm* [BI98], der die Kontur einer Geste vor unruhigem Hintergrund verfolgt, die Stützvektormethode (engl.: *Support Vector Machine*, SVM

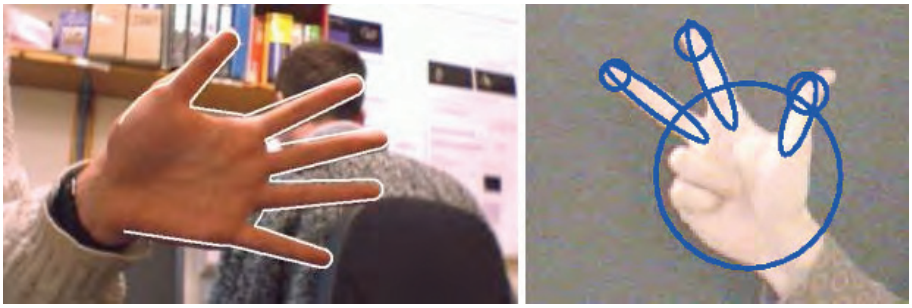

Abbildung 2.4: Unterschiedliche Hintergründe bei der Erkennung von Handgesten. Inhomogener und nicht statischer Hintergrund links (aus [STTC03]) und uniformer Hintergrund, rechts (aus [BLL02]). Beide Bilder zeigen die Ergebnisse der Erkennung durch Bildüberlagerungen.

[LGS08], ein Verfahren der Mustererkennung, das oft zur Klassifikation von Gesten und deren Merkmale verwendet wird oder endliche Automaten (engl.: *englisch Finite State Machine*, FSM) [LS04, HTH00], die oft zur Unterscheidung bekannter Gesten eingesetzt wird.

- Geforderte Einschränkung

  Da es zur Zeit noch keine allgemeingültige Methode zur Erkennung und Verfolgung von deformierbaren Objekten wie der menschlichen Hand gibt, fordern die meisten in der Literatur beschriebenen Verfahren und Systeme entweder Einschränkungen, unten denen das Verfahren effektiv arbeitet, oder fokussieren den Mehrwert eines neuen Algorithmus auf ein spezielles Problem, das für andere Anwendungen noch als Einschränkung gilt. Der wichtigste Punkt für eine intuitive Interaktion ist die Echtzeitfähigkeit. Viele Verfahren sind zwar in der Lage, die Pose der Hand genau zu erkennen [STTC06, BKMM+04], brauchen dafür aber deutlich mehr als die für eine Echtzeitanwendung geforderte $1/20$ Sekunde. Andere Systeme arbeiten zwar mit der notwendigen Geschwindigkeit von zwanzig Bildern pro Sekunde, reduzieren dabei aber die Genauigkeit der Erkennung oder die Anzahl vom System unterscheidbarer Gesten [SK07, BDC06]. Eine andere gebräuchliche Einschränkung ist die Wahl des Hintergrundes. Einige Systeme fordern einen uniformen und einfachen Hintergrund [TWYL05, HMLW03], während andere Verfahren besonderen Wert auf die Trennung der Hand vor einem unruhigen Hintergrund legen [FWCL07].

Ungeachtet der zuvor genannten Einteilungen soll in dieser Arbeit aus technischer Sicht eine neue und einfache Taxonomie verwendet werden, die nicht die benutzte Hardware oder die verwendeten Algorithmen zugrunde legt, sondern die Verfahren bereits im Hinblick auf die Aufgaben des Systems beschreibt. Dabei kann vorausgesetzt werden, dass ein Verfahren den Anforderungen für die intuitive Interaktion genügt und insbesondere die Bedingung der Echtzeitfähigkeit erfüllt. Da der Anwender nicht am Systemaufbau beteiligt ist, kann bereits im Vorfeld auf die erforderlichen Einschränkungen wie Beleuchtungssituation oder

Kameraaufbau eingegangen werden. Für viele Anwendungen, die mittels Gestenerkennung gesteuert werden sollen, ist es ausreichend, einzelne vordefinierte Handposen unterscheiden zu können und deren Position im Raum zu ermitteln. Oft sind bei einer Interaktion mit einer dreidimensionalen virtuellen Welt die Positionsbestimmung und die Unterscheidung einer Zeigegeste, der geschlossenen und der offenen Hand als Synonyme für *Zugreifen* und *Loslassen* von virtuellen Gegenständen bereits ausreichend. Dies ist beispielsweise der Fall, wenn es darum geht, Objekte aufzunehmen und an einer anderen Stelle des Raums wieder abzulegen. Andere Anwendungen erfordern allerdings mehr "Feingefühl" vom Nutzer. Beispiele dafür sind Anwendungen, bei denen ein virtuelles Objekt direkt manipuliert werden soll und damit einzelne Fingerstellungen relevant sind. Aufgrund dieser Überlegungen kann nun eine Einteilung getroffen werden, die ausschließlich die ermittelten Freiheitsgrade der interagierenden Hand betrachtet:

- Gestenerkennung mit 3+n Freiheitsgraden

  Neben der Position der Geste im Raum (3 DOF) ist das System in der Lage, n verschiedene statischer Handposen voneinander zu unterscheiden.

- Gestenerkennung mit 26 Freiheitsgrade

  Neben der Position und Orientierung der Handgeste im Raum (6 DOF) ist das System in der Lage, für alle 15 internen Gelenke der Hand die entsprechenden Rotationswinkel anzugeben.

Aus dieser einfachen Einteilung lassen sich direkt die technischen Anforderungen an ein Gestenerkennungssystem ableiten, wenn bereits die Anwendung, für die das System eingesetzt werden soll, bekannt ist. Soll beispielsweise eine Anwendung das Neupositionieren von Objekten in einer virtuellen Welt ermöglichen, ist ein Verfahren aus der Klasse "3+n Freiheitsgrade" ausreichend. Das Verfahren benötigt ein kalibriertes Stereokamerasystem und muss in der Lage sein, die interagierende Hand in den Kamerabildern vom Hintergrund der Szene zu trennen. Außerdem ist eine Methode notwendig, die entweder auf 2D-Bildbasis oder anhand von 3D-Informationen n unterschiedliche Gesten unterscheiden kann. Ein solches Verfahren wird in den Kapiteln 5 und 6 dieser Arbeit beschrieben. Sollen in einer Anwendung Änderungen an einem dreidimensionalen virtuellen Objekt vorgenommen werden oder Objekte beispielsweise mit einem Pinzettengriff[2] aufgenommen werden, ist ein Verfahren aus der Klasse "26 Freiheitsgrade" notwendig.

2.4.2 Einteilung aus Benutzersicht

Neben der zuvor entwickelten Taxonomie aus technischer Sicht soll in dieser Arbeit eine zweite Einteilung von Gestenerkennungssystemen vorgestellt werden. Die Einteilung verzichtet vollständig auf technische Randbedingungen und betrachtete die Verfahren ausschließlich aus der Sicht des Anwenders. Da der Nutzer eines Gestenerkennungssystems, das einfach und intuitiv bedienbar ist, sich nicht mit der verwendeten Technik auseinander setzen soll,

[2] Pinzettengriff ist das Greifen mit Daumen und Zeigefinger

steht die Bedienung der Anwendung selbst im Vordergrund. Damit lassen sich Gestenerkennungssysteme anhand der Aufgaben, für die das System verwendet werden soll, klassifizieren:

- Anwendungssteuerung mittels GUI-Elementen

  Für die Anwendungssteuerung können Gesten zur Bedienung einer grafischen Benutzeroberfläche (GUI) eingesetzt werden. Dabei werden Gesten zur Steuerung klassischer Standardelemente wie Schaltflächen, Menüs, Regler und Auswahlboxen eingesetzt. Das System muss hier in der Lage sein, einzelne Handgesten zu erkennen und in Echtzeit zu verfolgen. Durch die Tatsache, dass diese Art der Anwendungssteuerung das klassische Eingabemedium der Computermaus durch ein videobasiertes Gestenerkennungssystem ersetzt, bietet sich hier die Verwendung einer Zeigegeste als Eingabemodalität an, da auch die Computermaus mit ihrer Rückantwort in Form eines Mauszeigers (Cursor) auf dem Bildschirm einen deiktischen Charakter ausweist. Für die Anwendungssteuerung bietet sich die Möglichkeit für eine multimodale Erweiterung, zum Beispiel durch die Verknüpfung von Sprache und Zeigegeste.

- Anwendungssteuerung durch direkte Objektmanipulation

  In diesem Fall kann das Gestenerkennungssystem dazu verwendet werden, dass der Anwender direkt und ohne Umwege über zusätzliche Steuerelemente mit einem virtuellen Objekt interagiert. Insbesondere sind dafür Anwendungen prädestiniert, die eine Interaktion mit dreidimensionalen Objekten verlangen. Ein typisches Beispiel ist die Untersuchung eines 3D-Modells durch die nutzerbestimmte Rotation des Objektes. In vielen Anwendungsszenarien ist daher eine generelle und vollständige Erkennung der Handpose mit einer hohen räumlichen Auflösung, also der Bestimmung der einzelnen Gelenkrotationen notwendig. Auch für die Objektmanipulation ist für eine intuitive Verwendbarkeit eine permanente visuelle Rückantwort des Systems erforderlich, beispielsweise durch die Darstellung der erkannten Handpose als eigenständiges 3D-Modell. Objektmanipulation ist allerdings nicht ausschließlich auf 3D-Modelle beschränkt. Wie in Kapitel 9 gezeigt wird, kann eine Zeigegeste dazu verwendet werden, ein 2D-Objekt wie beispielsweise ein digitalisiertes Gemälde mittels einer virtuellen Lupenfunktion zu erkunden. Die Zeigegeste steuert dann Position die Lupe selbst und verändert damit direkt und ohne den Umweg von interaktiven Schaltflächen das Erscheinungsbild des Gemäldes.

Die Einteilung in Anwendungssteuerung mittels GUI-Elementen und durch direkte Objektmanipulation ist aus Sicht des Anwenders zwar der zuvor vorgestellten technischen Klassifikation anhand der verwendeten Freiheitsgrade sehr ähnlich, allerdings nicht identisch. Wie in Kapitel 9 gezeigt wird, ist es auch mit nur vier Freiheitsgraden (der Position im Raum (3 DOF) einer Zeigegeste (1 DOF)) möglich, dreidimensionale Objekte auf intuitive Weise zu rotieren und so zu erkunden. Dabei wird die Rotation des 3D-Objektes durch Pfeilsymbole auf dem Bildschirm realisiert, auf die der Anwender deuten kann.
In der Literatur sind viele Prototypen beschrieben, die klassische zweidimensionale Be-

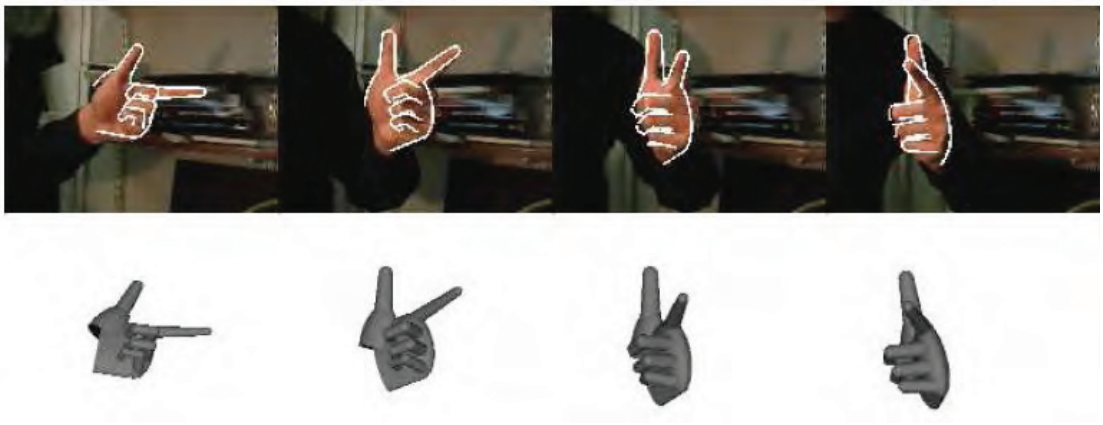

Abbildung 2.5: Verfolgung der Orientierung einer Zeigegeste mit einer einzelnen Kamara (aus [STTC06]). Die obere Reihe zeigt die Kamerabilder mit überlagerten Konturen, die untere Reihe zeigt die Rekonstruierte 3D-Pose der Hand.

dienoberflächen mittels eines definierten Satzes an statischen Handgesten steuern [WO03, KLP04] oder auch 2D-Steuerelemente verwenden, um dreidimensionale Objekte zu manipulieren [KK05, OZR02]. Die Verwendung von dynamischen Gesten [ICLB05, OSK02] ist dagegen seltener erwähnt und wird eher für die gestengesteuerte Kontrolle von mobilen Robotern verwendet [HMLW03, MOC06].

2.5 Herausforderungen

Die Erkennung und Verfolgung der menschlichen Hand ist eine herausfordernde aber auch notwendige Aufgabe, um eine Interaktion mittels Gestenerkennung zwischen Anwender und Computer zu realisieren. Bereits existierende Methoden konzentrieren sich meist auf die Verwendung von technischen Hilfsmitteln wie beispielsweise Datenhandschuhen [SZ94] oder farblich markierten Handschuhen [FRF08], um den Berechnungsaufwand so weit wie möglich zu minimieren und so die Echtzeitfähigkeit des Systems zu gewährleisten. Gestenerkennungssysteme, die eine gerätefreie Interaktion durch die Verwendung von Kamerasystemen realisieren, müssen dagegen ein hohes Maß an aufkommenden Datenströmen in kürzester Zeit mittels Software verarbeiten. Um die Verarbeitungszeit soweit zu reduzieren, dass ein videobasiertes System den Anforderungen an eine intuitive Interaktion genügt, ist daher a priori-Wissen notwendig. Für ein Farbkamerasystem kann beispielsweise die Tatsache ausgenutzt werden, dass die Farbe der menschlichen Haut nur ein begrenztes Intervall des Farbraum abdeckt. Damit ist es möglich, durch einen einfachen bildverarbeitenden Segmentierungsschritt Objekte wie den Kopf oder die Hände des Anwenders vom Hintergrund zu trennen. Nichtsdestotrotz sehen sich videobasierte Systeme zur Gestenerkennung einer Reihe von grundlegenden Schwierigkeiten und Herausforderungen gegenüber:

- Dimension des Suchraums

Wie im vorigen Abschnitt beschrieben ist die menschliche Hand ein komplexes kinematisches System mit bis zu 27 Freiheitsgraden. Auch wenn zwischen den Gelenken ein hohes Maß an Abhängigkeiten besteht und damit die Bewegungsfreiheit der Gelenke stark eingeschränkt ist, bleibt doch der Suchraum für eine Handgeste beliebig groß und erzeugt dadurch ein hohes Maß an Berechnungsaufwand.

- Umgebungsbedingungen

  Für eine robuste Erkennung von sowohl starren als auch deformierbaren Objekten wie der menschlichen Hand spielen oft Umgebungsparameter eine wichtige Rolle. Insbesondere sich während der Interaktion ändernde Lichtbedingungen können die Erkennung und Verfolgung des Objektes erschweren. Grund dafür ist unter anderem, dass Beleuchtungsschwankungen zu insgesamt geänderten Kamerabildern führen und somit eine Segmentierung der Hand vor einem bekannten Hintergrund der Szene erschwert wird.

- Selbstverdeckungen

  Durch die hohe Anzahl an Freiheitsgraden der Gelenke der menschlichen Hand kommt es bei Kamerasystemen oft zu Verdeckungen einzelner Handsegmente. Diese Verdeckungen können prinzipiell nur durch eine höhere Anzahl von Kameras, die in dem Systemaufbau verwendet werden, reduziert werden. Dadurch erbezeichnethöht sich allerdings auch das Datenaufkommen und damit die notwendige Rechenleistung des Systems.

- Verarbeitungszeit

  Die Verwendung von Kameras zur Objekterkennung stellt eine große Herausforderung an die Verarbeitungsgeschwindigkeit dar. Bereits der Einsatz von zwei Kameras in einem Stereokamerasystem erzeugt eine große zu verarbeitende Datenmenge. Dies ist insbesondere der Fall, wenn für die Interaktion in Echtzeit eine Bildwiederholrate von zwanzig Bildern pro Sekunde und mehr gefordert werden muss. Durch den daraus resultierenden hohen Rechenaufwand kommt es zudem oft zu einer Systemverzögerung von mehr als 200 Millisekunden, die vom Anwender als unnatürlich empfunden wird.

- Geschwindigkeit der Handbewegungen

  Die Bewegungen der menschlichen Hand können hohe Geschwindigkeiten erreichen. Die Hand selbst kann sich mit bis zu 5 Metern pro Sekunde im Raum bewegen (Translation), wobei das Handgelenk eine Drehung von bis zu 300° pro Sekunde zulässt [EBN+07] (Rotation). Auch für die handinternen Gelenkrotationen sind schnelle Statuswechsel zwischen einzelnen Posen möglich. So kann beispielsweise in wenigen Millisekunden von der offenen zur geschlossenen Hand gewechselt werden. Heutige Standardkameras liefern Bildern in einer Auflösung von 640*480 Pixeln mit in der Regel 30 Bildern pro Sekunde. Daraus ergeben sich in aufeinander folgenden Bilder eines Videostroms Probleme insbesondere bei der Verfolgungen von Handgesten im Raum und handinternen Bewegungen.

Abbildung 2.6: Handposenverfolgung mittels Optimierungsverfahren *Stochastic Meta-Descent* (SMD) [BKMM+04].

2.6 Positionsbestimmung

Eine elementare Aufgabe für ein Gestenerkennungssystem ist die Positionsbestimmung der Hand im Kamerabild und die Trennung der Handsilhouette vom Hintergrund des Bildes. Ungeachtet der Anzahl der verwendeten Kameras ist eine oft verwendete Methode die Segmentierung der Hand anhand von Hautfarbe [FWCL07, CVB04, MOC06]. Insbesondere bei der Verwendung von Farbzuordnungstabellen ist diese Methode trotz des höheren Datenvolumens von Farbkameras für Echtzeitanwendungen geeignet. Andere Systeme verwenden Graustufenkameras und eine zusätzliche Beleuchtung, um die Segmentierung der Hand zu realisieren. Ein häufig beschriebenes Problem bei beiden Ansätzen, das einen weiteren Verarbeitungsschritt notwendig macht, ist eine präzise Trennung von Hand und Unterarm [Fun02, DS99], da es ohne weiteres Wissen häufig zu einer Segmentbildung kommt, bei der die Hand und der Arm als ein einziges Objekt erkannt werden.
Um den Segmentierungsschritt zu vereinfachen, setzen viele in der Literatur beschriebenen Prototypen und Systeme Einschränkungen an die Umgebung voraus. Ein typisches Beispiel dafür ist die Verwendung von statischen Hintergrundinformationen, die über Referenzbilder [BH08] oder adaptive Hintergrundbestimmung [ARHN08, CBV03] eine Segmentierung des Anwenders ermöglichen. Auch andere Annahmen wie eine konstante Beleuchtung der Szene oder die Einschränkung, dass die zu verfolgende Hand das einzige hautfarbene Objekt in den Kamerabildern ist, sind gebräuchlich.
Eine weitere Klasse von Algorithmen zur Positionsbestimmung sind klassische objekterkennende Verfahren, die auf den zweidimensionalen Bildern des Kamerasystems angewendet werden und oft eine Hintergrundextraktion überflüssig machen. Wichtige Beispiele sind Genetische Algorithmen [KM07, LH99], Template Matching-Verfahren [CT01, BI98] oder Optimierungsstrategien [BKMM+04, OH99]. Diese Verfahren liefern bei der Suche nach der Handgeste meist deutlich mehr Information als ausschließlich die Position der Hand im Bild, und erleichtern mit ihren Ergebnissen zusätzliche Parameter für die folgende Klassifikation unterschiedlicher Gesten.

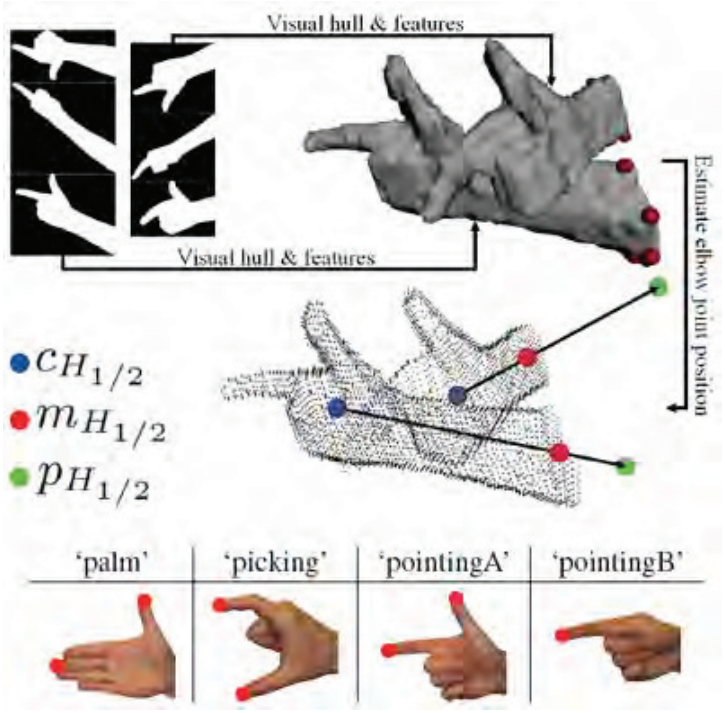

Abbildung 2.7: Schlattman und Klein [SK07] unterscheiden vier Gesten durch Merkmalsbestimmung in der visuellen Hülle der Hand.

2.7 Merkmalsbestimmung

Nach den bildverarbeitenden Schritten wie beispielsweise der Segmentierung oder einer Kantenextraktion ist die Merkmalsbestimmung der nächste wichtige Schritt eines Gestenerkennungssystems. Extrahierte Merkmale können dann in einem weiteren Schritt zur Unterscheidung von vordefinierten Gesten oder zur Schätzung der vollständigen Handpose verwendet werden. Grundsätzlichen stehen zwei verschiedene Ansätze der Merkmalsbestimmung zur Verfügung. Merkmale können als zweidimensionale Parameter aus den Bildinformationen gewonnen werden. Klassische 2D-Merkmale sind die Bestimmung einzelner Parameter wie des Segmentschwerpunktes, der äußeren Umrandung des Segments oder des segmentumschließenden Rechtecks, Farb- und Texturinformationen, sowie lokalisierbare Merkmalspunkte des Modells [GT05]. Oft verwendete Methoden, den Merkmalsraum in seiner Dimension zu reduzieren und damit die erforderliche Rechenleitung zu minimieren sind die Hauptkomponentenanalyse (PCA) [CAHS06, DT02], Hu- oder Zernike-Momente [Hu62, ABEN08] und Fourierdeskriptoren [HE04, UO99]. Oft wird eine Kombination aus mehreren merkmalsbestimmenden Verfahren gewählt, um die gewünschten Merkmale zu ermitteln. O'Hagan et al. [OZR02] beispielsweise kombinieren Template Matching-Verfahren, Momentenanalyse und Geometrische Merkmalsextraktion, um das Handgelenk, die Fingerspitzen und die Basis der einzelnen Finger zu ermitteln.

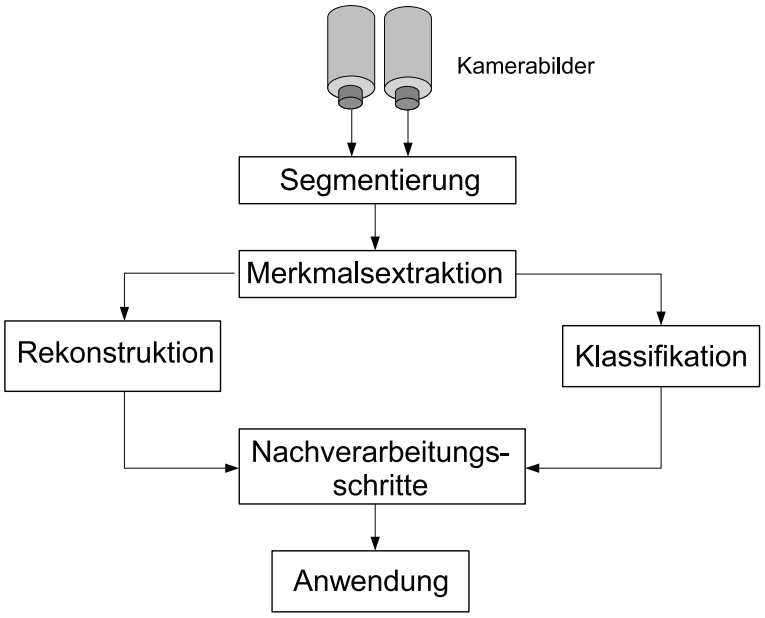

Abbildung 2.8: Skizze der Verarbeitungskette einer Klassifizierungsanwendung.

Neben der Möglichkeit, Merkmale zunächst auf 2D-Basis zu bestimmen und erst in einem zweiten Schritt eine 3D-Rekonstruktion durchzuführen ist es, die Merkmale direkt im dreidimensionalen Raum anhand eines 3D-Modells zu ermitteln. Schlattmann und Klein [SK07] beispielsweise verwenden ein Stereokamerasystem mit drei bis sechs Kameras, um die visuelle Hülle [Lau94] der Hand zu rekonstruieren. Der immense Rechenaufwand für diesen Schritt wird durch die Verwendung des Grafikprozessors zur Berechnung ermöglicht [LMS03]. Die in der dreidimensionalen visuellen Hülle ermittelten Merkmale werden verwendet, um vier unterschiedliche Handgesten zu unterscheiden (siehe Abbildung 2.7).

2.8 Gesten-Klassifikation

Die Klassifikation verwendet zuvor ermittelte Merkmale, um verschiedene Handgesten zu unterscheiden. Für diesen Schritt ist immer eine Trainingsphase notwendig, in der extrahierte Merkmale mit a priori-Wissen einer Klasse zugeordnet werden. Zur Laufzeit der Gestenerkennung werden dann neue Merkmale erzeugt und die Wahrscheinlichkeit der Klassenzugehörigkeit bestimmt. In der Literatur sind viele unterschiedliche Verfahren zur Klassifikation von Handgesten beschrieben. Die am häufigsten verwendete Methode ist das Hidden-Markov-Modell (HMM) [Fin03]. Beispiele für die Klassifikation mit Hidden-Markov-Modellen sind in [EAHAM08, CESJG05, ELD01, SP95] zu finden. Elmezain et al. beispielsweise verwenden eine Klassifikation mittels HMM, um neun statische Handposen für die arabischen Ziffern 0 bis 9 in einem Stereokamerasystem mit zwei Kameras in Echtzeit zu unterscheiden und erreichen dabei Erkennungsraten von über 95%.

Eine zweite häufig verwendete Methode der Klassifizierung sind künstliche neuronale Netze [EPdRH02]. Foong et al. [FTJL08] verwenden ein neuronales Netz für den Prototypen eines Systems, dass universelle Gesten einer Gebärdensprache erkennt und klassifiziert, um sie in synthetisch erzeugte Sprache umzuwandeln. Die Klassifizierung erfolgt in einer Trainingsphase anhand von zuvor aufgenommenen Testvideosequenzen einer einzelnen Farbkamera und erreicht so eine Erkennungsrate von knapp 80%. Das von Carbini et al. [CVB04] beschriebene System erlaubt einem Anwender vor einem großen Ausgabebildschirm mittels Zeigegeste mit der Anwendung zu interagieren. Sie verwenden ein neuronales Netz, um nach einer Hautfarbensegmentierung zu entscheiden, ob es sich bei den entstehenden Segmenten im Bild um den Kopf oder eine der Hände des Anwenders handelt. Durch die Verwendung eines Stereokamerasystems wird daraufhin die Zeigerichtung als Vektor zwischen Kopf und zeigender Hand im dreidimensionalen Raum berechnet.
Neben Hidden-Markov-Modellen und künstlichen neuronalen Netzen sind noch weitere Klassifikationsmethoden wie Clustering-Methoden [WSE+05], Fuzzylogik [BDC06] und anderen in der Literatur beschrieben, die bei der Erkennung und Verfolgung von Handgesten eingesetzt werden. Abbildung 2.8 zeigt den typischen Ablauf eines Gestenerkennungssystems, das durch Klassifikation von zweidimensionalen Merkmalen eine vorgegebene Anzahl von Handgesten unterscheidet. Zunächst wird auf Bildbasis ein Segmentierungsschritt durchgeführt, um die Handgeste vom Hintergrund zu trennen und die Merkmale der Geste zu bestimmen. Die Merkmale werden zum einen zur Bestimmung der verwendeten Handpose als auch zur Rekonstruktion der für die Interaktion relevanten 3D-Parameter verwendet. Nach eventuellen Nachverarbeitungsschritten wie beispielsweise einer Glättung der Ergebnisse, wird die Geste an die Anwendung übergeben.

2.9 Kommerzielle Systeme

In diesem Abschnitt sollen abschließend drei kommerzielle, videobasierte Gestenerkennungssysteme vorgestellt werden, die eine Interaktion mit großen Ausgabeflächen und den dort dargestellten Anwendungen ermöglichen. Alle drei Systeme bedienen sich der Zeigegeste als alleinige Interaktionsmodalität, zum Teil mit der Möglichkeit, beidhändig zu interagieren. Die Systeme arbeiten mit Zwei-Kamerasystemen, die eine berührungslose Interaktion ohne die Verwendung von Hilfsmitteln erlauben. Als kommerzielle Produkte heben die Werbetexte nicht nur den intuitiven, berührungslosen und nicht-invasiven Charakter der Interaktion und die dadurch entstehende hygienische Nutzung des Systems hervor, sondern unterstreichen auch die Attraktivität der neuartigen Interaktionsform, die den meisten Menschen bisher nur aus Science-Fiction-Filmen bekannt ist. Alle drei Systeme verweisen dabei auf Steven Spielbergs Blockbuster “Minority Report” (USA 2002), obwohl in diesem Film die gestenbasierte Interaktion durch Handschuhe, die mit Leuchtdioden ausgestattet sind, realisiert wurde. Allen hier vorgestellten Systemen ist außerdem gemeinsam, dass aufgrund ihres kommerziellen Charakters und den wahrscheinlich daraus resultierenden Fragen nach Patenten oder Patentanmeldungen keine oder kaum wissenschaftlichen Publikationen über die verwendeten Verfahren veröffentlicht sind.

Abbildung 2.9: Der *iPoint Presenter* des Fraunhofer Instituts für Nachrichtentechnik, HHI in Berlin ermöglicht eine beidhändige Interaktion.

2.9.1 iPoint Presenter

Das im März 2008 der Öffentlichkeit vorgestellte System des Fraunhofer Institut für Nachrichtentechnik, Heinrich-Hertz-Institut[3] in Berlin *iPoint Presenter* arbeitet mit zwei Kameras, welche die Position der Zeigefinger des Nutzers ermittelt und deren Bewegung verfolgen. Die verwendeten Kameras können über dem Anwender, also beispielsweise an der Decke des Interaktionsraums angebracht werden. Daneben existiert die Möglichkeit, den *iPoint Presenter* mit seiner Hardware als kompaktes Gehäuse zwischen Anwender und Ausgabebildschirm zu positionieren (siehe Abbildung 2.9, links). Diese Variante ermöglicht einen schnellen und mobilen Einsatz des Systems an unterschiedlichen Orten ohne die Notwendigkeit, Kameras zu montieren und auszurichten. Eine Besonderheit des Systems ist die Möglichkeit der *Multipointing-Interaktion*. Das System ist ähnlich einem *Multi Touch Table* in der Lage, bis zu acht Finger des Nutzer gleichzeitig zu erkennen und zu verfolgen. Damit können Nutzer durch beidhändige Interaktion die auf dem Bildschirm dargestellten Objekte rotieren, skalieren, greifen und loslassen und auch virtuelle Schaltflächen drücken, indem sie einen oder zwei Finger zur Interaktion verwenden (siehe Abbildung 2.9, rechts). Als Anwendungsfelder wird für den *iPoint Presenter* neben Informationssystemen in öffentlichen Bereichen, der Steuerung von Anwendungen im medizinischen Kontext und einer interaktiven Produktpräsentation in Geschäften auch die berührungslosen Steuerung von Maschinen in Produktionsbereichen, beispielsweise bei stark verschmutzten Händen oder in schwer zugänglichen Bereichen genannt. Als Forschungsinstitut sieht das Fraunhofer Institut für Nachrichtentechnik, HHI den *iPoint Presenter* als Demonstrator für die entwickelte Trackingtechnologie und die erarbeiteten Konzepte der Gestensteuerung und vertreibt daher das System nicht nur als fertiges Produkt, sondern bietet außerdem eine kundenorientierte Forschung und Weiterent-

[3] http://www.hhi.fraunhofer.de/

Abbildung 2.10: *GestPoint*®-System der amerikanischen Firma GestureTek™.

wicklung an.

2.9.2 GestureTek

Die in Kalifornien in den Vereinigten Staaten von Amerika ansässige Firma GestureTek[4] ist nach eigenen Angaben der Weltmarktführer auf dem Gebiet der kamerabasierten, kommerziellen Gestenerkennungssysteme und versteht sich als Unternehmen für angewandte Computer Vision-Technologien. GestureTek™ bietet eine Vielzahl von unterschiedlichen kamerabasierten Interaktionssystemen wie beispielsweise interaktive Tische oder beleuchtete Fußböden an. Als Handgestenerkennungssystem hat die Firma unter dem Produktgruppennamen *GestPoint*® verschiedene videobasierte Systeme im Angebot, die eine Interaktion durch eine Zeigegeste ermöglichen. Das System *HoloPoint*™ verwendet zwei Kameras, die fest in die Enden eines Interaktionsrahmens aus Metall eingebaut sind. Dieser halboffene Rahmen wird zwischen Anwender und Ausgabebildschirm positioniert. Hält der Anwender seine Hand in den Rahmen, erkennen die Kameras eine Störung in einem aus schwarzen und weißen Linien bestehenden Muster an der Innenseite des Rahmens und erlauben so eine Berechnung des Interaktionspunktes aus dem Bildschirm (siehe Abbildung 2.10, links). Durch die Verwendung des kontrastreichen Musters im Interaktionsrahmen ist das System robust gegenüber Lichtschwankungen der Umgebung. Als Anwendungsgebiete werden neben Konferenzräumen und Museen auch Einkaufszentren und Empfangshallen von Firmengebäuden genannt.
Eine Variante des Systems wird unter dem Namen *GestPoint Overhead Tracking* vertrieben, bei der auf den Interaktionsrahmen am Boden verzichtet wird. Stattdessen wird ein knapp zwei Meter langer Balken, in der die Kamerahardware des Systems untergebracht ist, an der Decke des Raums über der Position zur Interaktion montiert (siehe Abbildung 2.10, rechts).

[4] http://www.gesturetek.com/

Abbildung 2.11: Zwei *PointAt*-Systems im Palazzo Medici Riccardi, einem Museum in Florenz. Die Anwendungen erlauben eine interaktive Untersuchung digitalisierter Fresken mittels Zeigegestenerkennung.

Eine schwarze Fußmatte, die auf dem Boden liegt, dient zum einen dem Nutzer als Indikator für die vorgegebene Stelle zur Interaktion, zum anderen wird dadurch ein ausreichender Kontrast zwischen der interagierenden Hand und dem Hintergrund gewährleistet.

2.9.3 Natural Interaction: PointAt

Auch das italienische Unternehmen *Natural Interaction*[5] mit Sitz im Chianti in der Toskana bietet eine Vielzahl von unterschiedlichen videobasierten Interaktionsformen wie beispielsweise interaktive, begehbare Böden, Multi-Touch-Tische oder eine Zeigegestenerkennung an. Allerdings ist das Unternehmen weniger auf den Vertrieb der zur interaktiven Interaktion notwendigen Systeme selbst ausgerichtet. Vielmehr sieht sich das Unternehmen als Forschungsinitiative mit dem Ziel, Werkzeuge für die Schnittstelle zwischen Computersystemen und auch nicht technisch geprägten Menschen an öffentlichen Orten zu entwickeln. Dementsprechend stehen bei den von *Natural Interaction* entwickelten Systemen immer die Anwendungen selbst im Vordergrund. Das Gestenerkennungssystem *PointAt* verwendet ein kalibriertes Stereokamerasystem mit zwei Kameras im Nah-Infrarot-Bereich, um für eine Anwendungsdarstellung den Interaktionsraum ausreichend abdunkeln zu können. Zusätzlich wird das Interaktionsvolumen mit Nah-Infrarotlicht beleuchtet. Das Verfahren der Erkennung und Verfolgung einer Zeigegeste beruht auf der Segmentierung der Körpersilhouette des Anwenders vom schwächer beleuchteten Hintergrund [CBV03]. Die Zeigerichtung wird über eine Merkmalsbestimmung der Position des Kopfes und des ausgestreckten Armes des Nutzers in Echtzeit ermittelt. Eine bestimmte Handpose ist während der Interaktion nicht erforderlich, da sich die Zeigerichtung über die Pose des ausgestreckten Arms und der ermittelten Position des Kopfes des Anwenders bestimmt wird. Das System ist in der Lage, durch

[5] http://www.naturalinteraction.org/

längeres Deuten auf in der Anwendung vordefinierte Zielregionen auf dem Ausgabebildschirm (*"clickable regions"*) Selektionsereignisse auszulösen, die zu einer entsprechenden Reaktion der darstellenden Anwendung führen. *PointAt* verwendet neben einem Standard-PC zwei Webcams als bildgebende Sensoren und ist damit das preiswerteste der hier vorgestellten Systeme.

2.10 Zusammenfassung

In diesem Kapitel wurde anhand der wissenschaftlichen Literatur ein Überblick über den aktuellen Stand der Forschung und Technik für videobasierte Gestenerkennungssysteme gegeben. Es wurden sowohl die in der Literatur verwendeten Handmodelle als auch die Schwierigkeiten und Herausforderungen für Gestenerkennungssysteme, die zur intuitiven Interaktion eingesetzt werden sollen, erläutert. Neben den gebräuchlichen technischen Einteilungen der existierenden Prototypen und Systemen wurde in diesem Kapitel eine eigene Taxonomie entwickelt, die Verfahren zur Gestenerkennung aus technischer Sicht und aus Anwendersicht einordnet. Aus technischer Sicht unterteilen sich demnach Verfahren in Systeme zur

- Gestenerkennung mit $3+n$ Freiheitsgraden, also der Bestimmung der Hand im Raum bei einer gleichzeitigen Unterscheidung von n verschiedene statischen Gesten und zur
- Gestenerkennung mit 26 Freiheitsgrade, also der Bestimmung der vollständigen kinematischen Pose der menschlichen Hand mir 15 Gelenken.

Aus Anwendersicht wird eine Einteilung der Verfahren in

- Systeme zur Anwendungssteuerung, also der Bedienung einer Anwendung über eine grafische Benutzeroberfläche und
- Systeme zur Objektmanipulation, um direkt mit virtuellen Objekten zu interagieren,

vorgeschlagen. Neben einer Vielzahl von Prototypen und in wissenschaftlichen Publikationen beschriebenen Systemen wurden abschließend auch drei kommerzielle Systeme vorgestellt, die eine einfache Interaktion in Echtzeit mittels einer Zeigegeste ermöglichen.

Obwohl in der wissenschaftlichen Literatur ein Vielzahl von unterschiedlichen Verfahren und Systemen zur videobasierten Gestenerkennung beschrieben ist, bleibt die Frage nach deren Verwendbarkeit für eine intuitive und einfache Interaktion zwischen Mensch und Computer weitgehend offen. Insbesondere die im vorigen Kapitel 1 definierten Anforderungen an die Verfahren für deren intuitive Verwendbarkeit (Verzicht auf technische Hilfsmittel, minimaler Trainingsaufwand und Echtzeitfähigkeit) erschweren den Einsatz der meisten Verfahren oder machen ihn sogar unmöglich. Insbesondere für eine intuitive Interaktion mit virtuellen Welten, wenn der Anwender mit einigem Abstand vor einem großen Ausgabegerät steht, ergeben sich durch den so entstandenen großen Suchraum Fragen, die für die Entwicklung neuer Verfahren gelöst werden müssen:

- Die Initialsuche nach der interagierenden Handgeste muss in Echtzeit erfolgen, damit die gewünschte Interaktion ohne Zeitverzögerung starten kann, ohne dass der Anwender Kenntnis über den Inhalt der derzeitigen Kamerabilder hat. Dieses Problem wird in Kapitel 4, *Interaktion durch Rekonstruktion von Aktiven Formen* und Kapitel 5, *Interaktion durch Punktprojektion* behandelt.

- Für eine intuitive Interaktion mit dreidimensionalen virtuellen Welten müssen neben der klassischen Zeigegeste auch weitere Gesten wie eine geschlossene und eine offene Hand in Echtzeit unterschieden werden können, um virtuelle Objekte im dreidimensionalen Raum neu positionieren zu können. Dieses Problem wird im Kapitel 6, *Interaktion durch Merkmalbasierte Gesten-Klassifikation* behandelt.

- Für feinmotorische Aufgaben in virtuellen Welten wie beispielsweise dem Greifen und Loslassen mit Hilfe des Pinzettengriffs muss ein Verfahren auf feine Unterschiede im kinematischen Modell der menschlichen Hand reagieren können, ohne dass der Echtzeitanspruch des Verfahrens eingeschränkt wird. Dieses Problem wird in Kapitel 7, *Interaktion durch Momenten-Analyse* adressiert.

Als erster Schritt zur Beantwortung dieser offenen Fragen wird in dieser Arbeit zur Entwicklung neuer Verfahren zur Gestenerkennung ein kalibriertes Stereokamerasystem verwendet, das es ermöglicht, die zur intuitiven Interaktion notwendigen, dreidimensionalen Parameter der Gesten zu bestimmen. Die mathematischen und technischen Grundlagen optischer Stereokamerasysteme werden deshalb im folgenden Kapitel 3 behandelt.

Kapitel 3

Optische Stereokamerasysteme

Alle in den folgenden Kapiteln entwickelten Verfahren verwenden neben klassischen Methoden der Bildverarbeitung und des Bildverstehens auch dreidimensionale Parameter, um eine intuitive Gestenerkennung zu realisieren. Während die bildbasierten Methoden genutzt werden, um automatisch relevante Informationen in den Kamerabildern zu bestimmen und korrespondierende Bildpunkte zu ermitteln, wird für die Berechnung der für die Interaktion notwendigen Parameter im Raum ein Stereokamerasystem bestehend aus zwei kalibrierten Kameras eingesetzt. Die Vorteile einer Verwendung eines Stereokamerasystems zur Erkennung von Handgesten liegen auf der Hand. Zum einen erlaubt die Möglichkeit einer extrinsischen Kalibrierung von zwei Kameras weitaus genauere Rekonstruktionsergebnisse als die Verwendung von nur einer Kamera. Während Einkamerasysteme in der Regel nur grobe Schätzungen über die dreidimensionale Position eines Bildpunktes geben können, erreicht ein kalibriertes Stereokamerasystem durch eine Triangulierung von korrespondierenden Bildpunkten 3D-Rekonstruktionsgenauigkeiten im Millimeterbereich. Dadurch kann auch die Bestimmung der relevanten Merkmale einer Geste im Raum präziser bestimmt werden und erleichtert damit eine intuitive Interaktion durch den Anwender. Zum anderen kann der Kameraaufbau deutlich flexibler gehalten werden. Während eine einzelne Kamera oft für den Anwender sichtbar positioniert werden muss, um eine bestimmte Position und Orientierung der Geste durch den Nutzer zu gewährleisten, können die Kameras eines Stereosystems außerhalb des Sichtbereichs des Anwenders angebracht werden. Durch diese Tatsache erhöht sich der immersive Charakter des Systems erheblich und fördert eine intuitive Interaktion zwischen Mensch und Computer.
Aus diesem Grund wird in diesem Kapitel die technische Grundlage für die folgenden Verfahren der Gestenerkennung beschrieben und gezeigt, wie zwei Kameras intern und extern kalibriert werden können, um aus korrespondierenden 2D-Bildpunkten einen Punkt im Raum zu bestimmen. Dieses Kapitel beschäftigt sich daher mit

- dem verwendeten Kameramodell, dem Kameraaufbau und grundlegenden Überlegungen zur Anordnung der Kameras,

- der internen (intrinsischen) Kamerakalibrierung, die beispielsweise dazu verwendet wird, um die Verzeichnung der Kameralinsen zu berechnen und auszugleichen,

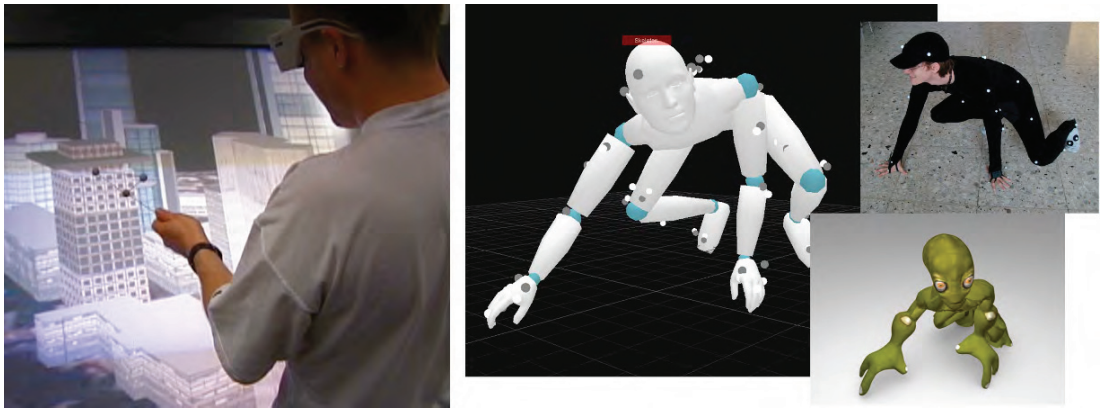

Abbildung 3.1: Infrarottrackingsystem und Motion Capturing-System. Interaktion durch 3D-Rekonstruktion von retroreflektierenden Markierung bekannter Geometrie [SM02], links und Übertragung von rekonstruierten Markierungen auf einen virtuellen Charakter.

- der externen (extrinsischen) Kamerakalibrierung, mit der die beiden Kameras zum einen zueinander in Relation gesetzt werden können und zum anderen durch die Definition eines Weltkoordinatensystems auch eine korrekte Berechnung der rekonstruierten Punkte im realen Interaktionsraum erlauben und

- der Möglichkeit, aus korrespondierenden Punkten der Bildebenen die Position des zugehörigen Punktes im Raum zu ermitteln (Triangulation).

Das Kapitel schließt mit einer kurzen Einführung der zur Gestenerkennung verwendeten Standardmethoden der Bildverarbeitung, die dazu verwendet werden, um automatisch korrespondierende Bildpunkte wie beispielsweise die Spitze des Zeigefingers oder der Schwerpunkt der Handgeste im Raum, zu bestimmen.

3.1 Einleitung

Stereokamerasysteme werden dazu verwendet, Punkte im Raum präzise zu rekonstruieren und zu verfolgen (engl.: *tracking*). Als Stereokamerasystem wird dabei ein Aufbau von zwei oder mehr Kameras bezeichnet, die in einer initialen Phase kalibriert werden. Sind die Kameras im definierten Weltkoordinatensystem fest aufgebaut, also ihre Position und Orientierung zueinander und zum Weltkoordinatensystem unveränderlich, muss eine Kalibrierung des Systems nur einmal durchgeführt werden. Eine erneute Kalibrierung wird erst erforderlich, wenn sich Position oder Orientierung einer der Kameras ändert. Stereokamerasysteme werden für eine Vielzahl von Anwendungen eingesetzt. Bekannte Beispiele sind Infrarottrackingsysteme zur Interaktion mit virtuellen 3D-Welten oder Motion Capturing-Systeme, welche die Bewegungen des menschlichen Körpers aufnehmen und zur Animation von virtuellen Charakteren verwenden.

Infrarottrackingsysteme verwenden meist ein Stereokamerasystem bestehend aus zwei Kameras [Sch06], wobei die Kameraobjektive mit Infrarotfilter versehen sind, die einen Großteil des sichtbaren Lichts blockieren. Wird die reale Szene mit zusätzlichen Infrarotlichtscheinwerfern beleuchtet, können Infrarotlicht-reflektierende Markierungen einfach in den Kamerabildern erkannt und zur 3D-Rekonstruktion verwendet werden. Ein solches Trackingsystem ist beispielsweise in der Lage, die Position und Rotation eines Interaktionsgerätes bestehend aus drei oder mehr retro-reflektierenden Markierungen, deren Abstände im Raum bekannt sind, zu bestimmen und zur Interaktion mit einer virtuellen 3D-Welt einzusetzen (siehe Abbildung 3.1, links).
Motion Capturing-Systeme sind heute der aktuelle Stand der Technik, um die Bewegungen eines Akteurs aufzuzeichnen und diese Bewegungen auf einen virtuellen Charakter zu übertragen. Dieses Verfahren wird beispielsweise bei der Filmproduktion oder auch zur Animation von Charakteren in Computerspielen eingesetzt. Bei optischen Motion Capturing-Systemen werden mit zehn oder mehr Kameras eine Vielzahl von kalibrierten Kameras verwendet, um Verdeckungsproblematiken auszugleichen, wenn der Anwender sich frei im Interaktionsraum bewegt und einzelne Markierungen nur durch die Verwendung mehrerer Kameras rekonstruiert werden können. Der Akteur trägt während der Bewegungsaufnahme einen speziellen Anzug, der mit Infrarotlicht-reflektierenden Markierungen versehen ist. In der rekonstruierten 3D-Punktewolke versucht die entsprechende Software dann die Körperpose des Anwenders zu ermitteln und für die Übertragung auf ein virtuelles Modell bereitzustellen (siehe Abbildung 3.1, rechts).

Im Rahmen dieser Arbeit wird grundsätzlich von einem System mit zwei kalibrierten Kameras ausgegangen. Wie in den folgenden Kapiteln gezeigt werden wird, sind für eine intuitive Interaktion mit Anwendungen, die auf einem Ausgabebildschirm direkt vor dem Anwender dargestellt werden, zwei Kameras ausreichend, um alle relevanten Parameter in den Kamerabilder zu erfassen und dreidimensional zu rekonstruieren. Um die hohe Präzision der Rekonstruktion zu gewährleisten, ist neben der intrinsischen und extrinsischen Kalibrierung auch eine Synchronisation der verwendeten Kameras notwendig. Nur wenn sichergestellt werden kann, dass ein von den Kameras aufgenommenes Bildpaar tatsächlich zeitgleich entstanden ist, kann der aus korrespondierenden Bildpunkten rekonstruierte 3D-Punkt auch präzise und korrekt berechnet werden. Unsynchronisierte Kameras mit einer Bildgröße von 640*480 Bildpunkten weisen in der Regel einen zeitlichen Versatz von etwa 40 Millisekunden auf, die benötigt werden, um nach der Belichtung der ersten Kamera die Bilddaten in den Speicher des Rechners zu übertragen, bevor die zweite Kamera ihr Bild aufnimmt und an den Rechner übermittelt. Die menschliche Hand kann bei schnellen Bewegungen Geschwindigkeiten von bis zu 5 m/s erreichen [EBN+07]. Damit ergibt sich für unsynchronisierte Kameras ein möglicher Unterschied von bis zu 20 Zentimetern zwischen den einzelnen Bildern eines Bildpaares. Um die daraus folgenden Fehler bei der 3D-Rekonstruktion zu vermeiden, werden die Kameras durch ein externes Signal zum gleichzeitigen Belichten veranlasst. Obwohl die Bilddaten auch in diesem Fall nacheinander in den Speicher des Rechners übertragen werden, ist damit sichergestellt, dass beide Bilder der Szene zum gleichen Zeitpunkt der Interaktion aufgenommen wurden und korrespondierende Bildpunkte eines beweglichen Objektes wie etwa der menschlichen Hand auch einen tatsächlichen Punkt im Raum beschreiben.

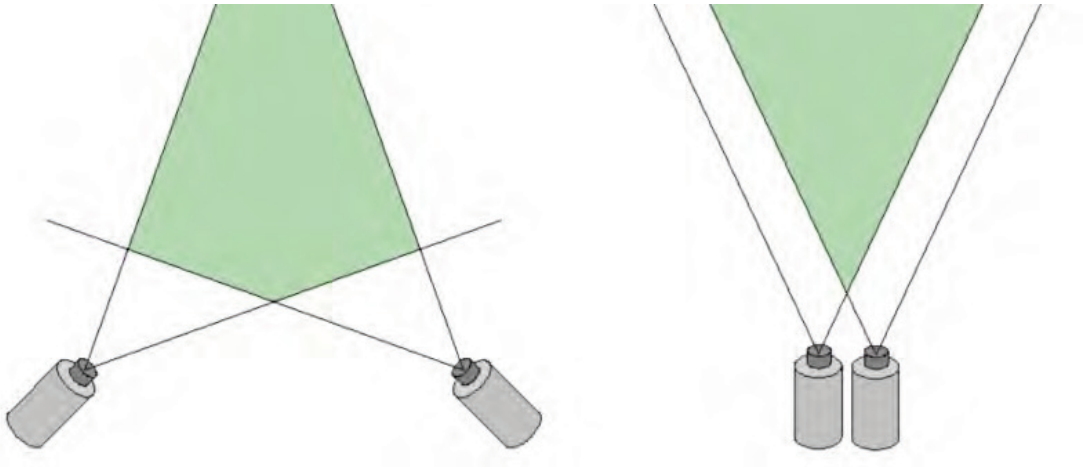

Abbildung 3.2: Unterschiedliche Positionierung und Orientierung zweier Kameras in der Aufsicht. Der Interaktionsbereich ist grün unterlegt. Links: Die Kameras stehen ungefähr im 90° Winkel zueinander. Der Interaktionsraum ist nahe bei den Kameras breit und verjüngt sich mit zunehmender Entfernung von den Kameras. Rechts: Die Kameras sind nahezu parallel ausgerichtet. Der Interaktionsraum wird in zunehmender Entfernung der Kameras größer.

3.1.1 Kameraaufbau und Interaktionsvolumen

Für sowohl die Präzision der 3D-Rekonstruktion als auch für Größe der Interaktionsvolumen spielt die Positionierung und Orientierung der Kameras im Systemaufbau eine entscheidende Rolle. Als Interaktionsvolumen wird dabei die Schnittmenge des von beiden Kameras erfassten Raums bezeichnet. Trivialerweise wird zunächst gefordert, dass sich die optischen Achsen, also die Vektoren, die orthogonal auf dem Zentrum der Kamerasensoren stehen, im Raum schneiden. Ist dies nicht der Fall, kann es im schlimmsten Fall dazu kommen, dass die Kameras kein gemeinsames Sichtvolumen besitzen und so eine Triangulierung von korrespondierenden Bildpunkt nicht möglich ist. Mit der erfüllten Forderung nach sich schneidenden optischen Achsen spielt die Position und Ausrichtung der Kameras zueinander eine wichtige Rolle. Wie in Abbildung 3.2 skizziert, erhöht zwar ein kleiner Abstand der Kameras zueinander das Sichtvolumen, allerdings kommt es durch die Verwendung von diskreten Bildpunkten zur 3D-Rekonstruktion insbesondere bei Punkten mit hohen Tiefenwerten zu größeren Fehlern bei der Bestimmung eines 3D-Punktes. Erhöht man den Abstand der Kameras zueinander, wird zwar das Interaktionsvolumen kleiner, aber es erhöht sich die Präzision der 3D-Rekonstruktion.
Die Entscheidung des geeigneten Kameraaufbaus wird im Fall der Gestenerkennung durch deren Aufgabe bestimmt. Da ein System zur Interaktion nicht um der Gestenerkennung selbst verwendet wird, sondern immer die Grundlage für eine einfache Interaktion mit ei-

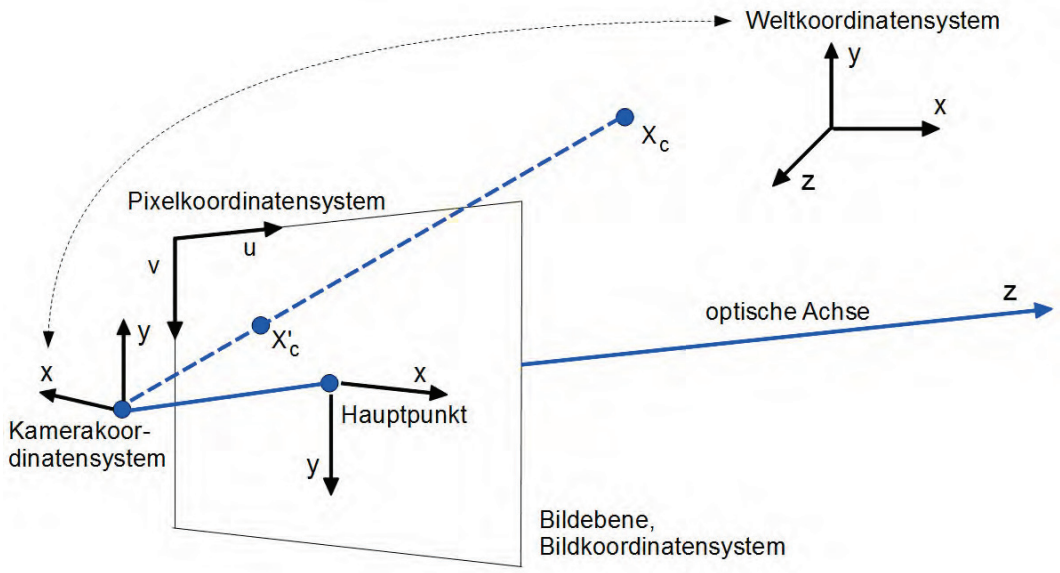

Abbildung 3.3: Skizze des Lochkameramodells. Die Abbildung zeigt das Kamera-, Bild- und Pixelkoordinatensystem und die Projektion eines Razumpunktes X_c.

ner auf einem Ausgabegerät dargestellten Anwendung darstellt, wird durch die Position des Bildschirms auch das Interaktionsvolumen des Anwenders bestimmt. Es kann davon ausgegangen werden, dass sich der Anwender und damit auch die Hand des Anwenders in einem beschränkten Bereich des Raumes vor dem Ausgabebildschirm befindet. Aus diesem Grund wird für die in den folgenden Kapiteln beschriebenen Verfahren ein Kameraaufbau von etwas 90° zueinander gewählt (siehe Abbildung 3.2, links). Dabei werden die Kameras in etwa drei Meter Höhe über dem Boden mit einem Abstand von anderthalb Metern rechts und links vom Anwender angebracht. Dieser Aufbau stellt nicht nur ein ausreichendes Interaktionsvolumen für eine Vielzahl von unterschiedlichen Nutzern und deren Art mittels Handgesten zu interagieren, bereit, sondern ermöglicht auch eine präzise 3D-Rekonstruktion mit einem mittleren Rekonstruktionsfehler von unter einem Millimeter [Sch06].

3.2 Kameramodell

Die mathematischen Grundlagen von Stereokamerasystemen und deren Kalibrierung und Anwendungen ist in der Literatur immer wieder beschrieben und diskutiert worden [SHB07, HZ04, Fau93, Tsa86, Tsa87, KU02, Sch06], sodass umfassende Literatur zu diesem Thema vorhanden ist. Der erste Schritt zur Beschreibung eines Stereokamerasystems ist die Festlegung des mathematischen Kameramodells. Das in dieser Arbeit verwendete Modell basiert auf dem Prinzip der Lochkamera und wird für eine präzise Triangulierung um intrinsische Parameter wie etwa der Linsenverzeichnung erweitert. Als externe (extrinsische)

Kameraparameter werden

$T \in \mathbb{R}^3$	der Brennpunkt und Ursprung des Kamerakoordinatensystems im Weltkoordinatensystem und
$R \in SO_3$	die Rotation der Kamera im Weltkoordinatensystem

verwendet. Als intrinsische Parameter werden

$f \in \mathbb{R}$	die Brennweite (fokale Länge) der Kamera,
$P = (u_0, v_0) \in \mathbb{R}^2$	der Hauptpunkt des Bildes (engl.: *principle point*), also der Schnittpunkt der optischen Achse mit der dazu orthogonal stehenden Bildebene,
$r_u, r_v \in \mathbb{R}$	Skalierungsfaktoren in x- und y-Richtung auf der Bildebene,
$s \in \mathbb{R}$	ein Verzerrungsfaktor (engl.: *skew*) der Kameralinse und
$\kappa \in \mathbb{R}$	die radiale Verzeichnung der Kameralinse

gewählt. Der Hauptpunkt $P = (u_0, v_0)$ entsteht praktisch durch, dass die Kameralinse nicht genau zentriert über dem Bildsensor der Kamera zum Liegen kommt. Die Skalierungsfaktoren r_u und r_v ermöglichen eine Umrechnung der Sensordaten in zugehörige Pixelkoordinaten des Kamerabildes und umgekehrt. Insbesondere die Linsenverzeichnung wird in der Literatur unterschiedlich behandelt. Grundsätzlich ist es möglich, die Verzeichnung der Linse durch die Verwendung von mehreren Parametern $\kappa_1, ..., \kappa_n$ genauer zu modellieren. Da sich der Einfluss auf die Rekonstruktionsgenauigkeit des Systems jedoch mit zunehmender Anzahl der Parameter verkleinert und so eine Minimierung des Fehlers nicht entscheidend erhöht, wird in dieser Arbeit aus Gründen der Effizienz der Berechnung auf die Verwendung von mehr als einem Linsenverzeichnungsparameter verzichtet. Abbildung 3.3 zeigt den Aufbau des Lochkameramodells ohne den Einfluss der Linsenverzeichnung.

Ohne den Parameter der Linsenverzeichnung lässt sich ein Punkt $X_w = (x_w, y_w, z_w)^T$ im Weltkoordinatensystem durch

$$X_c = R^{-1} \cdot X_w - R^{-1} \cdot T \tag{3.1}$$

auf die Bildebene in Pixelkoordinaten projizieren, wenn alle Kameraparameter durch geeignete Kalibrierungsschritte bekannt sind. Bei dieser Transformation liegt die Bildebene mit dem Abstand der fokalen Länge f orthogonal zur optischen Achse der Kamera vor dem Projektionszentrum, also der Position der Kamera im Raum. Mit Hilfe der Strahlensätze ist die Projektion durch $(-f \cdot x_c/z_c, -f \cdot y_c/z_c)^T$ zu berechnen. Die restlichen intrinsischen Kameraparameter lassen sich in der Kamerakalibrierungsmatrix K mit

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} r_u & s & u_0 \\ 0 & r_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -f \cdot x_c/z_c \\ -f \cdot y_c/z_c \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} -f \cdot r_u & -f \cdot s & u_0 \\ 0 & -f \cdot r_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{=K} \cdot \begin{pmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{pmatrix} \tag{3.2}$$

zusammenfassen. Diese Matrix wird so genannt, da alle intrinsischen Konstanten der Kalibrierung in einer einzigen Matrix enthalten sind. Die endgültigen kartesischen Pixelkoordinaten

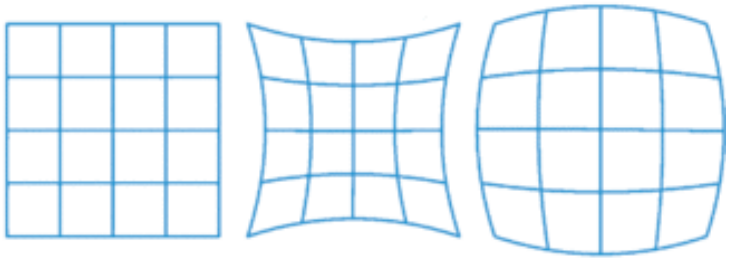

Abbildung 3.4: Schematische Darstellung verschiedener Linsenverzeichnungen. Ideales Bildgitter ohne Linsenverzeichnung (links), kissenförmige (mitte) und tonnenförmige Verzerrung (rechts).

ergeben sich somit als $(u, v)^T = (u'/w', v'/w')^T \in \mathbb{R}^2$. Entgegen der eher üblichen diskreten Darstellung von Pixelkoordinaten als ganzzahlige Werte $p = (u, v)^T \in \mathbb{N}^2$ wird hier das Bildkoordinatensystem reell angenommen. Nicht nur durch die Projektion von 3D-Punkten mittels der Kamerakalibrierungsmatrix K entstehen reellwertige Bildpunkte, auch liefern geeignete bildverarbeitende oder bildverstehende Verfahren durch entsprechende Gewichtung von Pixelmengen reellwertige Punktkoordinaten (vergleiche auch Abschnitt 3.6.3).

Die bisherigen Formeln berücksichtigen noch nicht den letzten der oben genannten Parameter der intrinsischen Kalibrierung, die radiale Linsenverzeichnung, da dieser Wert nicht durch die Kamerakalibrierungsmatrix K beschrieben werden kann. Linsenverzeichnung (auch Linsenverzerrung genannt) ist eine durch die Form der Linse des Kameraobjektivs bestimmte lokale Veränderung des Abbildungsmaßstabes. Je nach Art der Änderung von der optischen Achse weg zu den Bildrändern spricht man von einer kissenförmigen oder tonnenförmigen radialen Verzeichnung der Linse (siehe Abbildung 3.4). Während die kissenförmige Verzeichnung meist bei Objektiven mit großen Brennweiten (Teleobjektiven) zu beobachten ist, tritt die tonnenförmige Verzeichnung meist bei Objektiven mit kurzen Brennweiten (Weitwinkelobjektiven) auf. In dem hier verwendeten Kameraaufbau werden aufgrund der kurzen Distanzen zwischen Anwender, Ausgabebildschirm und den Kameras Objektive mit einer kurzen fokalen Länge von etwa vier Millimeter gewählt, die bereits eine deutlich wahrnehmbare, tonnenförmige Verzeichnung aufweisen (siehe Abbildung 3.5). Um die kalibrierte Verzeichnung bei der Projektion eines 3D-Puntes auf die Bildebene zu berücksichtigen, werden die durch die Gleichung 3.2 gewonnenen 2D-Koordinaten $u_u = {}^{-f \cdot x_c}/_{z_c}$ und $v_u = {}^{-f \cdot y_c}/_{z_c}$ zunächst mit

$$\begin{aligned} u_d &= \frac{2u_u}{\sqrt{1-4(u_u^2+v_u^2)\cdot\kappa}} \\ v_d &= \frac{2v_u}{\sqrt{1-4(v_u^2+v_u^2)\cdot\kappa}} \end{aligned} \tag{3.3}$$

transformiert [Len87] und danach weiter zu Pixelkoordinaten umgerechnet. Der Index d bezeichnet dabei einen verzerrten (engl.: *distorted*), ein Index u einen unverzerrten (engl.: *undistorted*) Punkt im Bildkoordinatensystem. Mit der Umkehrfunktion der Gleichungen aus

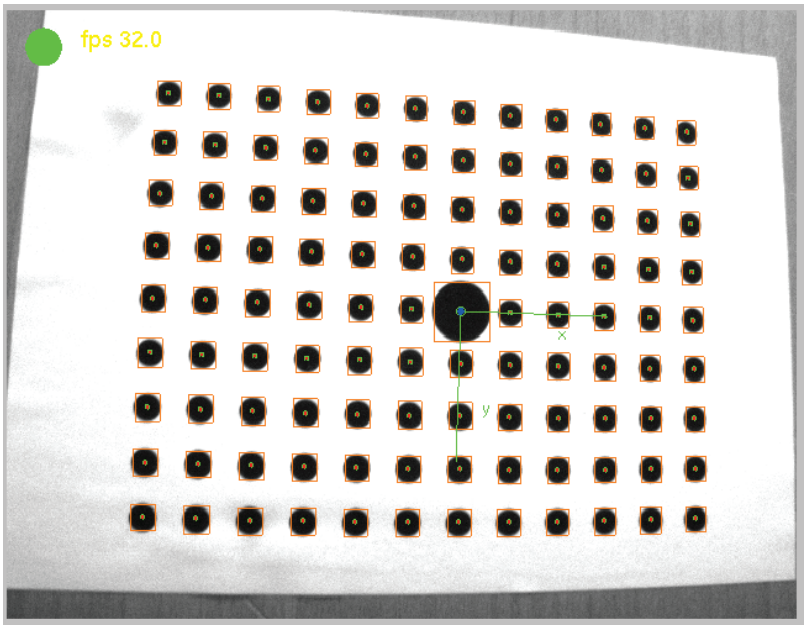

Abbildung 3.5: Kalibrierungsmuster mit Überlagerungen zur intrinsischen Kamerakalibrierung. Deutlich zu erkennen ist eine tonnenförmige Linsenverzeichnung.

3.3 ergibt sich aus einem projizierten, verzerrten Punkt ein unverzerrter Bildpunkt:

$$u_u = \frac{u_d}{1+(u_d^2+v_d^2)\cdot\kappa}$$
$$v_u = \frac{v_d}{1+(v_d^2+v_d^2)\cdot\kappa}, \tag{3.4}$$

der beispielsweise bei der Triangulierung von korrespondierenden Bildpunkten zur Rückprojektion in den Raum verwendet werden kann (vergleiche Abschnitt 3.5).

3.3 Intrinsische Kamerakalibrierung

Wie im vorigen Abschnitt beschrieben, werden mittels der intrinsischen Kalibrierung die internen Kameraparameter bestimmt. Eine intrinsische Kamerakalibrierung erfolgt meist nach den in von Tsai [Tsa86] und Lenz [Len87] beschriebenen Verfahren. Dabei werden mit Hilfe von bildverarbeitenden Verfahren wie einer schwellwertbasierten Segmentierung (vergleiche Abschnitt 3.6.3) Punkte eines geometrisch bekannten Kalibriermusters (engl.: *calibration pattern*) auf Bildbasis bestimmt und daraus die internen Kameraparameter bestimmt. Das in dieser Arbeit verwendete Verfahren nach Zhang [Zha00] erweitert die Standardmethoden um die Möglichkeit, mehrere Aufnahmen desselben Musters ohne vordefinierte Position und Ausrichtung des Musters bezüglich der Kamera zu verwenden, um die intrinsischen Parameter zu berechnen. Abbildung 3.5 zeigt das hier verwendete Muster mit insgesamt 108

Abbildung 3.6: Aufzeichnung von Punktkorrespondenzen zur extrinsischen Kamerakalibrierung durch Segmentierung einer Leuchtdiode, die im Interaktionsraum geschwenkt wird. Bereits aufgenommene Punktepaare werden zu Kontrollzwecken durch Bildüberlagerung visualisiert.

Punkten in jeweils neun Zeilen und zwölf Spalten. Der Ursprung des Musters ist durch eine einzelne größere Markierung in der Mitte des Musters festgelegt. Während der Aufnahme der einzelnen Bilder des Musters wird davon ausgegangen, dass sich die Ausrichtung des Musters in Bezug auf die Kamera nicht signifikant ändert, also die lokale y-Achse nach unten und die lokale x-Achse nach rechts zeigt, wie in den Bildüberlagerungen in Abbildung 3.5 zu sehen ist. Für eine intrinsische Kalibrierung nach Zhang werden n ähnliche Bilder (beispielsweise $n = 20$) aufgenommen. In jedem der n Bilder werden die Pixelkoordinaten der 108 Markierungen, deren 3D-Position in dem lokalen Koordinatensystem mit einem Tiefenwert von $z = 0$ bekannt ist, bestimmt. Damit ergeben sich $108 * n$ 3D- zu 2D-Punktkorrespondenzen, aus denen in einem nichtlinearen Optimierungsschritt die zuvor genannten intrinsischen Parameter ermittelt werden können. Für diesen Optimierungsschritt wird in dieser Arbeit das Levenberg-Marquardt-Verfahren [Mar63, PTVF92] verwendet, das neben einer akzeptablen Laufzeit auch ein robustes Konvergenzverhalten gewährleistet.

3.4 Extrinsische Kamerakalibrierung

Mit den zuvor bestimmten intrinsischen Kameraparametern und den in Abschnitt 3.2 gezeigten Formeln zu Projektion von Raumpunkten auf die Bildebene können in einem weiteren Schritt die Positionen und Orientierungen der Kameras des Stereosystems zueinander als extrinsische Kameraparameter berechnet werden. Für das in dieser Arbeit verwendete Stereokamerasystem wird eine halbautomatische Kalibrierung nach Azarbayejani [AP95] verwendet. Der extrinsische Kalibrierungsschritt beruht auf einer nichtlinearen Minimierung eines Residuums, das durch die Analyse von korrespondierenden Bildpunkten entsteht. Halbautomatisch wird die Kalibrierung deshalb genannt, weil die notwendigen Punktkorrespondenzen interaktiv durch den Anwender bereitgestellt werden. Eine einfache Methode zur Erzeugung

dieser Punktkorrespondenzen ist es, eine Lichtquelle im Sichtvolumen der beiden Kameras zu schwenken. Durch eine schwellwertbasierte Segmentierung werden die nach ihrer Intensität gewichteten Schwerpunkte der Lichtpunkte in beiden Kamerabildern bestimmt (vergleiche Abschnitt 3.6.3). Abbildung 3.6 zeigt die Aufnahme solcher Korrespondenzen: Zur Erzeugung der Lichtpunkte wird eine Taschenlampe mit einer Leuchtdiode im Interaktionsraum bewegt. Durch die Verwendung von synchronisierten Kameras ist dabei sichergestellt, dass auch bei schnelleren Bewegungen des Kalibrierobjektes die zu segmentierenden Punkte gleichzeitig entstehen und damit einen realen Punkt im Raum beschreiben können. Zur Bestimmung des Residuums wird für jedes korrespondierende Punktepaar (U_i^1, U_i^2) der 2D-Punkt der ersten Kamera U_i^1 mit Hilfe der intrinsischen Kameraparameter und der aktuellen Transformationsparameter mit den in Abschnitt 3.2 beschriebenen Formeln auf die Bildebene der zweiten Kamera abgebildet. Der hier entstehende 2D-Punkt $\tilde{U}_i^2$ weist einen umso kleineren Abstand zum originalen Punkt U_i^2 auf, je besser die Transformationsparameter gewählt sind. Damit ergibt sich für m aufgezeichnete, korrespondierende Bildpunkte erneut ein nichtlineares Optimierungsproblem, dass die Summe der m Abstände zwischen Originalpunkt und projiziertem Punkt in der Bildebene der zweiten Kamera

$$\sum_{i=1..m} r_i = \sum_{i=1..m} \|\tilde{U}_i^2 - U_i^2\| \tag{3.5}$$

minimiert. Wie bereits bei der intrinsischen Kalibrierung wird auch für die Minimierung der beschriebenen Residuumsfunktion für die extrinsische Kalibrierung das Levenberg-Marquardt-Verfahren [Mar63, PTVF92] eingesetzt. Das Ergebnis des Optimierungsschrittes ist bei bekannten intrinsischen Parametern beider Kameras die Translation und Rotation der Kameras zueinander. Ohne weiteres Wissen über den Kameraaufbau ist somit das Weltkoordinatensystem zunächst identisch mit dem lokalen Koordinatensystem der ersten Kamera. Für die Verwendbarkeit in den in den folgenden Kapiteln beschriebenen Verfahren zur intuitiven Gestenerkennung ist es aber notwendig, das Weltkoordinatensystem entsprechend der Anwendung zu definieren. Soll beispielsweise die Position einer erkannten Handgeste für die Interaktion mit einer virtuellen dreidimensionalen Welt verwendet werden, ist es notwendig, ein für die Anwendung verwendbares Weltkoordinatensystem bereitzustellen, in dem auch die Position und Ausrichtung des Ausgabebildschirm festgelegt ist. Dies kann erreicht werden, indem nach der extrinsischen Kalibrierung für das zu definierende Weltkoordinatensystem der Ursprung, ein festgelegter Punkt auf der x-Achse und ein beliebiger Punkt auf der y-Achse des Koordinatensystems angegeben. Dies geschieht erneut durch Aufnahme von entsprechenden Punktkorrespondenzen. Die dritte Achse des Weltkoordinatensystems kann dann als Kreuzprodukt der beiden interaktiv angegebenen Achsen berechnet werden. Durch die bereits erfolgte Kamerakalibrierung lassen sich beliebig skalierte Distanzen zwischen den Punkten bestimmen und mit tatsächlich gemessenen Distanzen vergleichen, sodass eine geeignete Skalierung der Welt festgelegt werden kann.

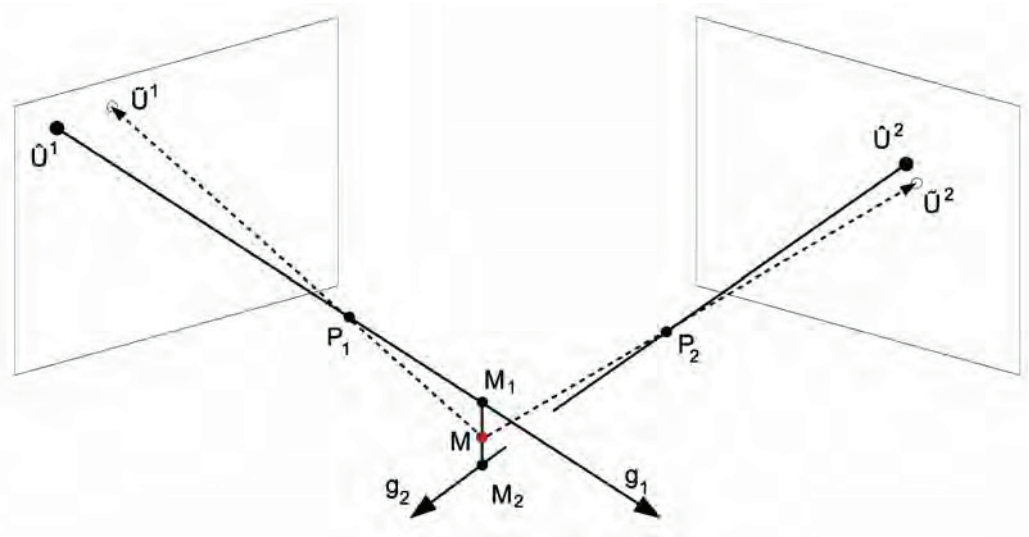

Abbildung 3.7: 3D-Rekonstruktion und Rekonstruktionsfehler aus zwei Kameras mit Projektionszentren P_1 und P_2.

3.5 3D-Rekonstruktion

Sind aus der intrinsischen und extrinsischen Kamerakalibrierung alle notwendigen Parameter des Stereokamerasystems bekannt, kann aus zwei korrespondierenden Punkten der Kamerabildebenen ein dreidimensionaler Punkt im Weltkoordinatensystem rekonstruiert werden. Diese Berechnung der 3D-Koordinaten wird oft auch als *Triangulierung* bezeichnet. In diesem Abschnitt wird davon ausgegangen, dass die beiden Punkte U^1 und U^2, aus denen ein Raumpunkt rekonstruiert werden soll, durch Anwendung der intrinsischen Parameter bereits im Kamerakoordinatensystem vorliegen. Die Bestimmung korrespondierender Punkte ist dagegen Bestandteil der einzelnen Verfahren der Gestenerkennungsmethoden und wird in den folgenden Kapiteln 4 bis 6 entwickelt. Als erster Schritt der Rekonstruktion werden beide 2D-Punkte um einen Tiefenwert erweitert, um 3D-Koordinaten zu erhalten:

$$\bar{U}^1 = \begin{pmatrix} U^1 \\ -f_1 \end{pmatrix} \in \mathbb{R}^3 \quad \text{und} \quad \bar{U}^2 = \begin{pmatrix} U^2 \\ -f_2 \end{pmatrix} \in \mathbb{R}^3 \tag{3.6}$$

Dieser Schritt ist möglich, da sich die Bildpunkte auf einer zu der jeweiligen (x, y)-Ebene parallelen Kamerabildebene liegen, wobei f_1 und f_2 die entsprechenden Brennweiten der Kameras darstellen und den Abstand der Kamerabildebenen zu der Projektionszentren (engl.: *center of projection*, cop) der Kameras definieren. Die 3D-Punkte können in einem weiteren Schritt durch Anwendung der extrinsischen Kalibrierung in das Weltkoordinatensystem transformiert werden:

$$\hat{U}^1 = R^1 \cdot \bar{U}^1 + T^1 \quad \text{und} \quad \hat{U}^2 = R^2 \cdot \bar{U}^2 + T^2 \tag{3.7}$$

mit der Translation T^j vom Ursprung des Weltkoordinatensystems zum Projektionszentrum P_j der Kamera j und der Rotation R^j der Kamera j im Weltkoordinatensystem. Durch die

Punkte P_1 und $\hat{U}^1$ lässt sich nun eine Gerade g_1 im Raum aufspannen. Analog entsteht durch die Punkte P_2 und $\hat{U}^2$ die Gerade g_2 (siehe Abbildung 3.7). Der Schnittpunkt der beiden Geraden ist der gesuchte Raumpunkt. Dieser mathematisch ideale Fall, dass sich die beiden Geraden im Raum schneiden, ist in der Realität jedoch nie der Fall, da durch die Berechnung der Punkte aus diskreten Pixelkoordinaten Fehler entstehen müssen. Aus diesem Grund wird als 3D-Rekonstruktion der Punkt im Raum verwendet, der den kürzesten Abstand zu den beiden Geraden aufweist. Dazu werden zunächst die Richtungsvektoren

$$\vec{r_1} = \overrightarrow{P_1 - \hat{U}^1} \quad \text{und} \quad \vec{r_2} = \overrightarrow{P_2 - \hat{U}^2} \tag{3.8}$$

der Geraden g_1 und g_2 berechnet. Der zu $\vec{r}_1$ orthogonale Vektor in der Ebene, die von den Vektoren $\vec{r}_1$ und $\vec{r}_2$ aufgespannt werden, berechnet sich durch den Entwicklungssatz für das Vektorprodukt durch

$$\vec{n}_1 = \vec{r}_1 \times (\vec{r}_1 \times \vec{r}_2) = \vec{r}_1 \cdot (\vec{r}_1 \cdot \vec{r}_2) - \vec{r}_2 \cdot (\vec{r}_1 \cdot \vec{r}_1) \tag{3.9}$$

Als nächstes wird die zu $\vec{n}_1$ orthogonale Ebene durch den Punkt P_1 mit der Geraden g_1 geschnitten, wobei die Ebene durch

$$\overrightarrow{X - P_1} \cdot \vec{n}_1 = 0 \tag{3.10}$$

beschrieben wird. Setzt man die parametrische Gleichung der Geraden

$$g_2 = \{X \in \mathbb{R}^3 : X = P_2 + t_2 \cdot \vec{r}_2, \quad \text{mit} \quad t_2 \in \mathbb{R}\} \tag{3.11}$$

in Gleichung 3.10 ein, erhält man die Gleichung

$$\overrightarrow{P_2 + t_2 \cdot \vec{r}_2 - P_1} \cdot \vec{n}_1 = 0, \tag{3.12}$$

die sich nach

$$t_2 = \frac{\overrightarrow{P_1 - P_2} \cdot \vec{n}_1}{\vec{r}_2 \cdot \vec{n}_1} \tag{3.13}$$

auflösen lässt. Mit t_2 findet sich der zur Geraden g_1 am nächsten liegende Punkt

$$M_2 = P_2 + t_2 \cdot \vec{r}_2, \tag{3.14}$$

der auf der Geraden g_2 liegt. Analog lässt sich der auf der Geraden g_1 und zur Geraden g_2 am nächsten liegende Punkt

$$M_1 = P_1 + t_1 \cdot \vec{r}_1 \tag{3.15}$$

berechnen. Mit dem arithmetischen Mittel der beiden 3D-Punkte

$$M = \frac{M_1 + M_2}{2} \tag{3.16}$$

ergibt sich dann der angenäherte Schnittpunkt für die 3D-Rekonstruktion von korrespondierenden Bildpunkten [Koh96, SMS01, Sch06] (siehe Abbildung 3.7).

Die beiden Punkte M_1 und M_2 können als Qualitätsmaß für die Güte der 3D-Rekonstruktionsgenauigkeit verwendet werden, indem die Distanz zwischen den beiden Punkten im Raum als 3D-Rekonstruktionsfehler

$$err = |M_1 - M_2| \tag{3.17}$$

aufgefasst wird. Damit können Aussagen über die Güte der Kamerakalibrierung und der Bestimmung der korrespondierenden Bildpunkte getroffen werden. Allerdings muss dabei beachtet werden, dass zwar ein hoher Rekonstruktionsfehler auch auf eine schlechte Qualität der Rekonstruktion schließen lässt, der Umkehrschluss allerdings nicht gilt. Durch die Windschiefe der beiden Gerade g_1 und g_2 kann es vorkommen, dass der ermittelte 3D-Punkt an der falschen Position im Raum liegt, obwohl der minimale Abstand der beiden Geraden klein ist [FKW04]. Einen anderen Ansatz zur Bestimmung der Güte der 3D-Rekonstruktion schlägt Dorfmüller [Dor99] vor. Wird der rekonstruierte 3D-Punkt zurück auf die Bildebene der Kamera projiziert, kommt die Projektion nicht auf dem ursprünglichen zur Triangulierung verwendeten 2D-Punkt zum Liegen (siehe Abbildung 3.7, Punkte $\tilde{U}^1$ und $\tilde{U}^2$). Die daraus entstehende Distanz zwischen dem Originalpunkt und dem projizierten 2D-Punkt dient somit als Maß für den entstandenen Rekonstruktionsfehler. Dieses Verfahren hat den großen Vorteil, dass der Fehler direkt als Überlagerungen auf den Kamerabildern dargestellt werden kann und damit eine direkte visuelle Kontrolle des Rekonstruktionsfehlers ermöglicht.

3.6 Bildverarbeitung

Im diesem Abschnitt des Kapitels sollen abschließend grundlegende Verfahren der Bildverarbeitung beschrieben werden, die für das Verständnis der folgenden Kapitel und der darin entwickelten Verfahren zur Gestenerkennung notwendig sind. Für die in den vorigen Abschnitten erläuterten Methoden zur dreidimensionalen Rekonstruktion von korrespondierenden Bildpunkten ist es notwendig, Punktkorrespondenzen automatisch zu erzeugen. Methoden, um beispielsweise aus einer Menge von unterschiedlichen 2D-Punkten durch geeignete Interpretation die zusammengehörigen Punkte zu identifizieren, ist Aufgabe der gestenerkennenden Verfahren selbst und wird daher in den jeweiligen Kapiteln dieser Arbeit erläutert. Voraussetzung dafür sind jedoch die in der Literatur ausgiebig beschriebenen bildverarbeitenden Verfahren (zum Beispiel [Jäh01, Tön05]), die eine Korrespondenzanalyse interessanter Bildpunkte zulassen. Daher werden hier kurz die in den Verfahren verwendeten Methoden zur Bildglättung, Kantenextraktion, Berechnung von Differenzbildern und zur Segmentierung vorgestellt. Die Kameras des hier verwendeten Stereokamerasystems liefern Luminanzbilder mit 256 verschiedenen Graustufen. Fasst man die durch die Kamera erzeugten Bilder als diskrete Funktionen $g(x, y)$ zweier Veränderlicher auf, liefert ein Wertepaar (x, y) des Definitionsbereichs einen Helligkeitswert $g \in \{0, ..., 255\}$ des Definitionsbereichs. Durch diese Beschreibung von Bildern als mathematische Abbildungen lassen sich zum einen einfache Bildeigenschaften wie beispielsweise der Abstand zweier Bildpunkte durch die euklidische Distanz

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{3.18}$$

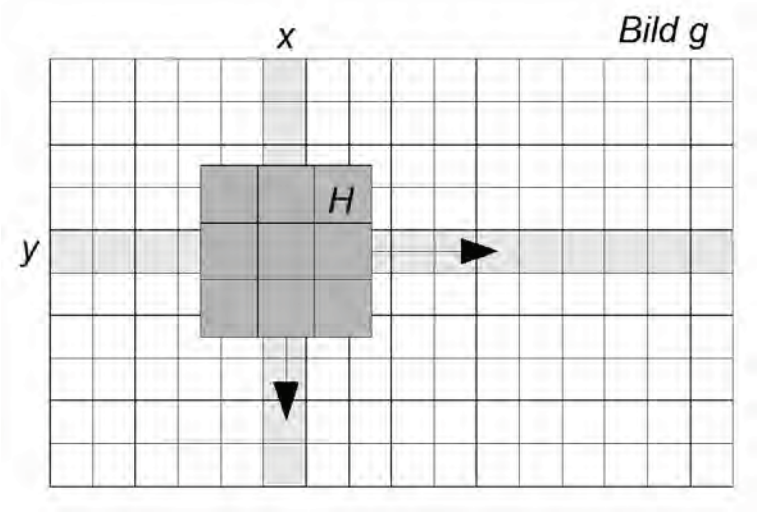

Abbildung 3.8: Faltung zwischen Bildfunktion und Filterkern. Der Filterkern H wird über die Bildfunktion g geschoben. Das Ergebnispixel errechnet sich über die gewichtete additive Verknüpfung des Pixels (x, y) mit seinen acht Nachbarn.

die, wie im vorigen Abschnitt beschrieben, zur Bestimmung des 3D-Rekonstruktionsfehlers verwendet werden kann. Zum anderen ermöglicht die Beschreibung auch, ein Bild durch vorgegebene Operationen, die auf die einzelnen Pixel des Bildes angewendet werden, in ein anderes Bild zu überführen, also bildverarbeitende Methoden anzuwenden, um die Informationen des Originalbildes entsprechen der jeweiligen Aufgabe zu verändern.

3.6.1 Filterung im Ortsbereich

Viele Bildverarbeitungsalgorithmen wie beispielsweise die Glättung verrauschter Bilder oder die Extraktion von Kanten aus einem Grauwertbild können durch eine Filterung im Ortsbereich des Bildes erreicht werden. Ein Punkt des bearbeiteten Bildes ergibt sich dabei durch eine gewichtete, additive Verknüpfung des Original-Bildpunktes und seiner Nachbarn. Diese Verknüpfung, die einer Faltung der Bildfunktion mit einer zuvor definierten Filtermaske H entspricht, kann wie folgt beschrieben werden: Sei $S_e = s_e(x, y)$ das Eingabebild, $S_a = s_a(x, y)$ das Ausgabebild und $H = h(u, v)$ die Filtermaske der bildverarbeitenden Operation, dann gilt:

$$s_a(x,y) = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} s_e(x+k-u, y+k-v) \cdot h(u,v) \tag{3.19}$$

wobei m die Größe der quadratischen Filtermaske angibt. Für den Parameter k gilt $k = \frac{m-1}{2}$.

Eine klassische Anwendung der Filterung im Ortsbereich ist die Glättung (engl.: *smoothing*) eines Bildes. Durch die Glättung werden Störungen, die bei der Aufzeichnung durch

Abbildung 3.9: Kantenextraktion einer Zeigegeste. Original-Luminanzbild der aufnehmenden Kamera (links), Differenzenoperator (Mitte) und Sobel-Operator (rechts). Die Kantenbilder sind der besseren Sichtbarkeit wegen invertiert dargestellt.

die Videosensoren der Kameras entstehen, eliminiert. Beispielsweise kann Rauschen in den Bildern zwar bei der bloßen Betrachtung eines Bildes kaum wahrgenommen werden, aber einen immensen Einfluss auf die Ergebnisse der weiteren Verarbeitung des Bildes haben. Mit Glättungsoperatoren steht deshalb ein einfaches Mittel zur Verfügung, um die Qualität eines Bildes zu verbessern. Bekannte Glättungsfilter sind der *gleitende Mittelwert*

$$H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \tag{3.20}$$

der aus dem zu bearbeitende Pixel und seinen acht Nachbarn das arithmetische Mittel berechnet und der *Gauß-Tiefpass*

$$H = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \tag{3.21}$$

Der Name kommt daher, dass seine Elemente grob die Gauß'sche Glockenkurve annähern. Bei diesem Filterkern wird das zugrunde liegende Pixel am stärksten, weiter entfernt liegende Pixel schwächer gewichtet. Das Ergebnis der Faltung muss natürlich geeignet skaliert werden, damit wieder Helligkeitswerte zwischen 0 und 255 entstehen und somit ein darstellbares Bild entsteht. Der berechnete Wert wird daher für den gleitenden Mittelwert mit $1/9$ und für den Gauß-Tiefpass mit $1/16$ gewichtet. Die beiden hier vorgestellten Filterkerne sind beide Matrizen der Größe 3×3. Verwendet man größere Filtermasken, verstärkt sich der Glättungseffekt zwar, allerdings geht aber auch mehr Originalinformation des Bildes verloren. Zudem erhöht sich die Verarbeitungszeit, so dass größere Filterkernen als 3×3 für eine Echtzeitanwendung wie der Gestenerkennung nicht verwendet werden.

Kantenfilter sind in der Bildverarbeitung eine weit verbreitete Technik, um Umrisse von Objekten in einem Bild zu beschreiben. Um diese Kanten zu detektieren, sucht man nach

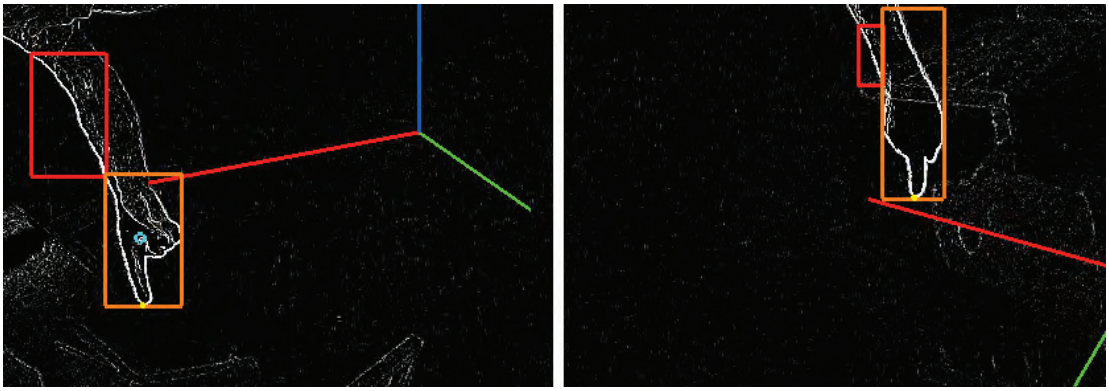

Abbildung 3.10: Differenzbilder des Stereokamerasystems während einer Zeigegeste. Dem Bildschirmfoto sind bereits Segmentierungsergebnisse und das Weltkoordinatensystem überlagert.

starken Helligkeitsübergängen in der Grauwertfunktion des Bildes. Der einfachste Kantenfilter ist der Differenzenoperator. Bildet man eine partielle Differentiation in Zeilen- und Spaltenrichtung nach, so erhält man zwei Filterkerne

$$H_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad H_y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{3.22}$$

Soll das entstandene Kantenbild wieder Grauwerte zwischen 0 und 255 annehmen, müssen die Ergebnisse der einzelnen Pixel geeignet skaliert werden. Aus den entstandenen Kantenpixeln kann nun der Betrag

$$g(x,y) = \sqrt{s_x(x,y)^2 + s_y(x,y)^2} \tag{3.23}$$

und die Richtung des Gradienten

$$tan\phi = \frac{s_x(x,y)}{s_y(x,y)} \tag{3.24}$$

berechnet werden. Der wohl bekannteste und am häufigsten verwendete Kantenfilter ist der Sobel-Operator mit den Filterkernen

$$H_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \text{und} \quad H_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{3.25}$$

3.6.2 Differenzbildanalyse

Für die Analyse von beweglichen Objekten in einer Sequenz von Einzelbildern spielen Differenzbilder eine wesentliche Rolle. Oft wird bei einer Differenzbildanalyse aus zwei kurz hintereinander aufgenommenen Kamerabildern ein Differenzbild mit

$$s_a(x,y) = |s_i(x,y) - s_{i-1}(x,y)| \tag{3.26}$$

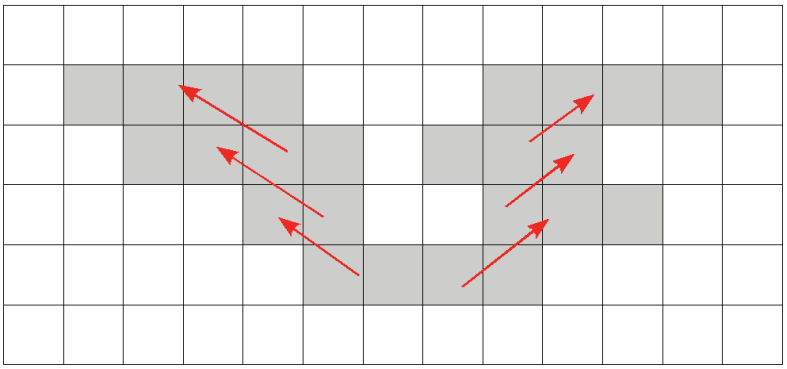

Abbildung 3.11: Einzelne Segment-Intervalle werden zu einem Segment zusammengefügt.

erstellt. Dieses Verfahren ermöglicht es, den Vorgang der Bewegung eines Objekts in ein stationäres Bild umzuwandeln. Für die Verfahren der Gestenerkennung ist allerdings eine Differenz von zwei nacheinander aufgenommenen Bildern nicht sinnvoll, da es sein kann, dass der Anwender des Systems beispielsweise für längere Zeit die Hand nicht bewegt und damit auch keine Konturen der Hand in den entstehenden Differenzbildern zu finden sind, die für eine Interaktion weiter analysiert werden könnten. Aus diesem Grund werden für die in den folgenden Kapiteln beschriebenen Verfahren Differenzbilder zwischen aktuellen Kamerabildern und einem Referenzbild berechnet. Das Referenzbild wird dabei normalerweise beim Start des Systems ermittelt, da zu diesem Zeitpunkt sichergestellt werden kann, dass sich kein Anwender im Interaktionsvolumen der Kameras befindet. Mit dieser Methode verbunden sind allerdings auch die typischen Schwächen bei der Verwendung von Referenzbildern, wie beispielsweise eine starke Abhängigkeit von sich änderndem Umgebungslicht. Schwankungen in einzelnen Bildern können ausgeglichen werden, indem nicht ein einzelnes Bild als Referenzbild herangezogen wird, sondern zunächst eine Reihe von n Bildern aufgenommen wird, aus denen ein Mittelwert-Referenzbild mit

$$s_a(x,y) = \frac{1}{n}\sum_{i=0}^{n} s_i(x,y) \tag{3.27}$$

berechnet wird. Auch die Verwendung von Kantenbildern zur Bestimmung der Differenzbilder anstelle der Verwendung der originalen Luminanzbilder der Kameras erhöht die Robustheit der Differenzbildanalyse gegenüber Schwankungen der Umgebungsbedingungen.

3.6.3 Segmentierung

Die Segmentierung eines digitalen Bildes ist oft der erste Schritt, um Aussagen über den Inhalt des Bildes zu gewinnen. Bei der Segmentierung wird versucht, ähnliche Pixel zu Segmenten zusammenzufassen und so Objekte im Vordergrund vom Hintergrund zu trennen. Zwar liefern einfache Segmentierungsalgorithmen noch keine Beschreibung der Objekte, die

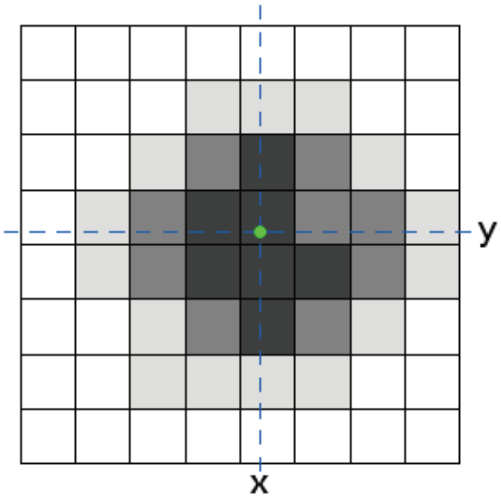

Abbildung 3.12: Nach Pixel-Intensitäten gewichteter Schwerpunkt eines Segmentes. Durch die nach Helligkeiten gewichtete Berechnung entsteht ein Schwerpunkt im Subpixelbereich.

Trennung der Objekte vom Hintergrund ist aber der erste Schritt in diese Richtung, da einzelne Segmente klassifiziert und ausgewertet werden können. Die einfachste Art der Segmentierung ist das *Schwellwertverfahren*. Dabei wird jedes Pixel des Bildes untersucht und mit einem vorgegebenen Schwellwert c verglichen. Liegt der Helligkeitswert des Pixels über dieser Schranke, wird der Punkt dem Vordergrund zugeordnet und beispielsweise weiß markiert. Ist das Pixel dunkler als der Schwellwert, wird es zum Hintergrund gezählt und schwarz markiert. Diese Binarisierung

$$s_a(x,y) = \begin{cases} 255 & \text{falls} \quad g = s_e(x,y) \geq c \\ 0 & \text{sonst} \end{cases} \tag{3.28}$$

des Kamerabildes liefert allerdings (wie in der klassischen Bildverarbeitung üblich) als Ergebnis der Operation erneut eine Bildfunktion. Für die Bestimmung von für die Gestenerkennung relevanten Parameter über die einzelnen Segmente wie beispielsweise deren Schwerpunkte, die Konturen der Segmente oder die umschließenden Rechtecke ist eine Analyse des binarisierten Bildes erforderlich. Ein einfacher Weg, um zum Vordergrund gehörende Pixel zu Segmenten zusammenzufassen und deren Segmentinformationen zu bestimmen ist es, wenn während der Segmentierung nicht nur die Formel 3.28 zur Erzeugung eines neuen Pixelwertes des Ausgabebildes berechnet wird, sondern während des Segmentierungsprozesses zusätzlich auch Informationen über zusammenhängende Pixel in jeder Zeile des Bildes gespeichert werden. Für die so entstehenden Intervalle werden neben Anfangs- und Endposition auch die Summe der Helligkeiten gespeichert. In einem Nachverarbeitungsschritt wird dann analysiert, welche der Intervalle sich in aufeinanderfolgenden Bildzeile überschneiden. Sich überschneidende Intervalle werden dann zu einem Segment zusammengefügt (siehe Abbildung 3.11). Neben der Umrandung (Kontur) des Segments enthält damit jedes Segment auch die Summe der Luminanzwerte aller zum Segment gehörenden Pixel des Originalbildes. Damit ist

es möglich neben dem geometrischen Schwerpunkt des Segments auch einen nach seiner Helligkeitsverteilung gewichteten Schwerpunkt des Segments

$$x_c = \frac{\sum_j m_j \cdot x_j}{\sum_j m_j} \quad \text{und} \quad y_c = \frac{\sum_j m_j \cdot y_j}{\sum_j m_j} \tag{3.29}$$

mit den Helligkeitswerten m_j zu berechnen. Die daraus resultierenden subpixelgenauen Schwerpunktinformationen führen bei der zuvor beschriebenen 3D-Rekonstruktion von korrespondierenden Bildpunkten in beiden Bildern des Stereokamerasystems zu präzisen objektbeschreibenden Positionen im Raum.

3.7 Zusammenfassung

In diesem Kapitel wurden die technischen Grundlagen für die Verwendung eines Stereokamerasystems als Teil der Verfahren zur intuitiven Interaktion durch Gestenerkennung beschrieben. Damit sind die in den folgenden Kapiteln beschriebenen Verfahren in der Lage, aus korrespondierenden Bildpunkten präzise 3D-Punkte im Interaktionsvolumen der beiden Kameras zu bestimmen. Die Verwendung von dreidimensionalen Informationen ist insbesondere bei der Interaktion mit virtuellen Welten notwendig, um die Position der Handgeste zu bestimmen und beispielsweise die Richtung einer Zeigegeste berechnen zu können. Für das hier verwendete Stereokamerasystem mit zwei Graustufenkameras wird das mathematische Modell der Lochkamera um intrinsische Parameter wie beispielsweise der Brennweite der Kamera und der radialen Linsenverzeichnung erweitert, um eine exakte 3D-Rekonstruktion von 2D-Punktepaaren zu ermöglichen. Dabei spielt zunächst die geeignete Wahl des Kameraaufbaus eine wichtige Rolle. Für die in dieser Arbeit vorgeschlagenen Verfahren werden die Kameras in einem Winkel von etwa 90° zueinander rechts und links vom Anwender positioniert. Die gewählte Hohe der Kameras über dem Anwender erlaubt es dem System, die Kameras außerhalb des direkten Sichtfeldes des Anwenders zu platzieren und somit den intuitiven und gerätefreien Charakter der Interaktion zu unterstreichen. Der gewählte Aufbau ermöglicht dennoch sowohl ein ausreichend großes Interaktionsvolumen als auch eine präzise 3D-Rekonstruktion von korrespondierenden Bildpunkten. Dazu werden in einer initialen Kalibrierungsphase des Systems die intrinsischen und extrinsischen Kameraparameter bestimmt. Die intrinsische Kalibrierung wird genutzt, um die internen Kameraparameter zu ermitteln. Während dieser Schritt für jede Kamera nur ein einziges Mal notwendig ist, wird die extrinsische Kalibrierung neu berechnet, falls sich der Aufbau der Kameras ändert, also Position oder Orientierung einer der Kameras sich ändert. Die extrinsische Kalibrierung wird durch ein einfaches, halbautomatisches Verfahren ermöglicht, bei dem korrespondierende Bildpunkte durch Schwenken eines Lichtpunktes im Sichtvolumen der Kameras analysiert werden. Beide Kalibrierverfahren sind als Optimierungsaufgaben aufzufassen, die durch eine nichtlineare Minimierung mit dem Levenberg-Marquardt-Verfahren in für die Anwendung akzeptabler Zeit gelöst werden können. Mit geeigneten Kamerakalibrierungsdaten des Systems ist es dann möglich, korrespondierende Punkte der beiden Kamerabilder zu einem exakten dreidimensionalen Punkt zu rekonstruieren und dessen Rekonstruktionsfehler zu untersuchen.

Während die Verfahren der Korrespondenzanalyse von Bildpunkten als Teil der gestenerkennenden Verfahren in den folgenden Kapiteln beschrieben wird, schließt dieses Kapitel mit den dafür notwendigen, bildverarbeitenden Methoden wie Kantenextraktion, Differenzbildanalyse und der Segmentierung von Graustufenbildern.

Kapitel 4

Interaktion durch Rekonstruktion von Aktiven Formen

Im folgenden Kapitel wird ein Verfahren vorgestellt, dass eine Zeigegeste anhand von Aktiven Formen (engl.: *Active Shape Model*, ASM) erkennt und verfolgt. Aktive Formen sind eine bewährte Methode zur Anpassung von deformierbaren Objekten. Eine der ersten Anwendungen, die Aktive Formen zur Gestenerkennung verwenden, wurde bereits vor über zehn Jahren von T. Heap entwickelt[1]. Das System verfolgt die Kontur einer geöffneten Hand, die direkt in die Kamera des Systems gezeigt wird. Zwar arbeitet das System in Echtzeit und verfolgt robust die Kontur der Hand, allerdings zeigt diese Anwendung exemplarisch auch die Probleme, die das Verfahren der Aktiven Konturen im Allgemeinen bei einer Verwendung für eine intuitive Interaktion hat. Zum einen benötigen Aktive Formen eine gute initiale Schätzung als Startpose, damit eine Anpassung der Kontur an die realen Bildinformationen gelingen kann. In der erwähnten Anwendung wird dies durch eine aktive Initialisierung durch den Anwender gelöst. Der Anwender muss auf die auf einem Monitor angezeigten Bilder der Kamera schauen, denen eine Startkontur der Hand in der Bildmitte überlagert ist. Das System wartet dann darauf, dass der Anwender seine geöffnete Hand mit der dargestellten Kontur in Einstimmung bringt, er also die vorgegebene Kontur wie einen virtuellen Handschuh über die eigene, reale Hand zieht. Zum anderen ist der Erfolg von sowohl der Initialisierung als auch der Verfolgung der Geste sehr stark von der Position und Orientierung der Hand im Interaktionsraum abhängig. Der Anwender muss die Handgeste direkt vor der Kamera ausführen und dabei darauf achten, dass die Handfläche parallel zur Kamerabildebene orientiert ist und dass dabei die Finger der Hand nach oben zeigen.
In diesem Kapitel soll ein Verfahren entwickelt werden, dass beide Probleme löst und sowohl eine vollständig automatische Initialisierung für eine Anpassung mittels Aktiven Formen realisiert als auch die verwendeten Kameras außer Sichtweite des Anwenders positionieren kann, um den immersiven Charakter für eine intuitive Interaktion zu gewährleisten.
In dem in diesem vorgestellten Verfahren wird die Initialschätzung sowohl durch die Verwendung eines Stereokamerasystems und damit einer geeigneten Positionierung und Orientierung der Kameras als auch durch ein nichtlineares Optimierungsverfahren zur Bestimmung

[1] University of Leeds Handtracker Demo, http://www.comp.leeds.ac.uk/vision/proj/ajh/tracker.html

von geeigneten Startwerten für die Translation, Rotation und Skalierung der Startkonturen gewährleistet. Während die Initialsuche und die Anpassung der Konturvorlagen zunächst rein auf Bildebene, also im Zweidimensionalen abläuft, um den Echtzeitanspruch des Verfahrens zu gewährleisten, wird zur Bestimmung der 3D-Position der Zeigegeste und zur Bestimmung der Zeigerichtung im Raum eine geeignete Parameterschätzung im kalibrierten Stereokamerasystem vollzogen. Das Verfahren gliedert sich in vier aufeinander folgende Schritte:

1. Beschreibung der Zeigegeste und Berechnung der Deformationsmöglichkeiten.

   Dieser Schritt des Verfahrens errechnet anhand von Trainingsbildern des kalibrierten Stereokamerasystems eine Mittelwertgeste und deren Deformationsmöglichkeiten auf Bildebene. Der zeitaufwendige Schritt der manuellen Erzeugung von Trainingskonturen der Geste muss nur ein einziges Mal durchgeführt werden, solange sich die Positionen und die Orientierungen der Kameras im Systemaufbau nicht entscheidend ändert.

2. Auffinden von Initialkonturen in den Kamerabildern.

   Dieser Schritt des Verfahrens verwendet neben Segmentinformationen zum Auffinden der Fingerspitze das heuristische Optimierungsverfahren *Simulated Annealing*, um initial die Position, Orientierung und Skalierung der Zeigegeste in einem Kamerabild zu bestimmen. Da *Simulated Annealing* nicht die Echtzeitanforderungen für die Interaktion erfüllt, wird die Initialsuche nur zu Beginn der Gestenverfolgung eingesetzt oder wenn während der folgenden Feinanpassungen eine ungültige Kontur der Geste ermittelt wird.

3. Anpassung der Initialkonturen an die realen Konturen in den Kamerabildern.

   Dieser Schritt des Verfahrens passt eine Initialkontur an die realen Bildinformationen der Geste in Echtzeit an. Zur Laufzeit der Gestenverfolgung entspricht die Initialkontur dem Ergebnis der Anpassung aus dem letzten Kamerabildpaar. Nur wenn eine sinnvolle Anpassung der Geste misslingt, werden Initialkonturen wie in Schritt 2 beschrieben neu ermittelt.

4. Berechnung der Gestenposition und der Zeigerichtung im dreidimensionalen Raum.

   Als letzter Schritt des Verfahrens wird die Position der Hand als Punkt im Raum und die für eine Interaktion notwendige Zeigerichtung als Strahl im Raum berechnet. Dabei wird als 3D-Position der Geste die Fingerspitze angenommen, die gleichzeitig auch als Startpunkt für den zur Interaktion verwendeten Zeigestrahl dient.

4.1 Einleitung

Aktive Formen (engl.: *Active Shapes*) sind ein seit Langem bekanntes Werkzeug des Bildverstehens, um deformierbare Objekte in Bildern zu erkennen, deren Erscheinungsbild zwar in Allgemeinen gut bekannt sind, die Konturen des Objektes jedoch Variationsmöglichkeiten aufweisen, die eine exakte Beschreibung des Objektes nicht zulassen. Die Objektdeformationen entstehen entweder durch Unterschiede der Instanzen realer Objekte selbst oder bei-

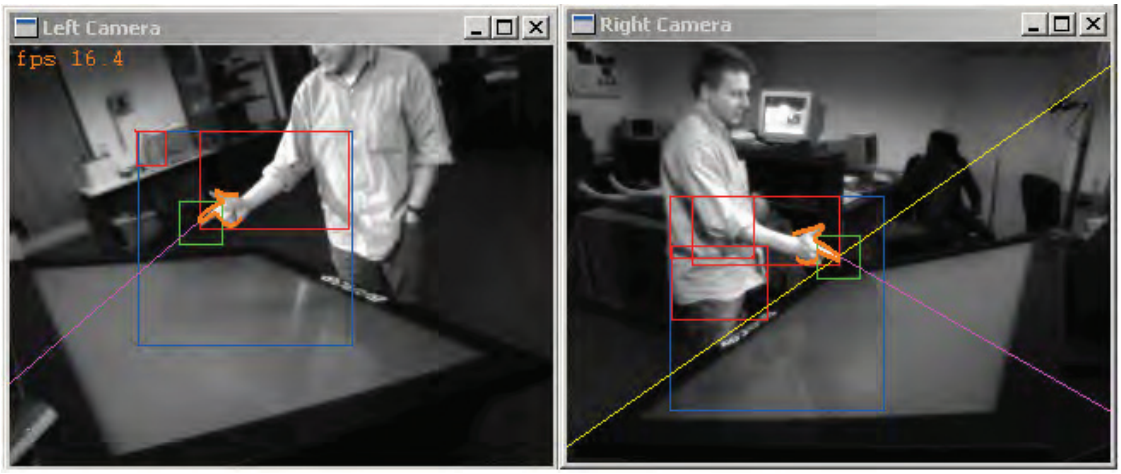

Abbildung 4.1: Interaktion an einem *Virtual Table*. In den Kamerabildern sind neben der angepassten Geste und ihrer Zeigerichtung auch Ergebnisse der Bildverarbeitungsschritte und im rechten Bild die Epipolarlinie der Fingerspitze angezeigt.

spielsweise durch eine leichte Änderung der Lage des Objektes im Raum bezüglich der aufnehmenden Kamera. Das Verfahren wurde bereits 1992 von Cootes und Taylor [Tay92, CTCG95] als Erweiterung der von Kaas et al. entwickelten Aktiven Konturen (engl.: *Active Contours*) [KWT87] als Anwendung zur automatischen Lagebestimmung von Transistoren auf Platinen beschrieben. Eine Erweiterung auf dreidimensionale Oberflächenmodelle von Objekten erfolgte 1997 von Heap und Hogg [HH97]. Blake und Isard [BI98] verwenden aktive Formen, um mit einer einzelnen, orthogonal zum Objekt ausgerichteten Kamera die Position und Orientierung einer Pose der menschlichen Hand im Raum in Echtzeit zu schätzen und zu verfolgen.

Die Anpassung einer Kontur an die realen Bildinformationen mittels aktiver Formen geschieht anhand einer statistischen Beschreibung der Geste. Aus einer Vielzahl von Trainingsgesten wird zunächst ein Punktverteilungsmodell (*Point Distribution Model*, PDM) [CTCG95] berechnet, das eine Mittelwertgeste und deren Deformationsmöglichkeiten beschreibt. Da das Erscheinungsbild der Geste sowohl von der Position und Ausrichtung der Kameras, als auch von der Orientierung der Hand im Raum abhängt, ist es wichtig, vorab den Kameraaufbau zu definieren. Für das hier vorgestellte Verfahren ist ein Winkel zwischen den Kameras von ca. 120° bis 150° symmetrisch zum Anwender vorgesehen. Die Kameras werden dabei in etwa in Schulterhöhe des Anwenders angebracht (siehe Abbildung 4.1). Dieser Aufbau ermöglicht nicht nur eine robuste Kamerakalibrierung, sondern gewährleistet außerdem, dass ausreichend Informationen der Zeigegeste für die Suche und Anpassung im Bild verfügbar sind. Der symmetrische Kameraaufbau ermöglicht außerdem, dass für beide Kameras das gleiche statistische Modell verwendet werden kann und lediglich eine Spiegelung der im Punktverteilungsmodell beschriebenen Geste erfolgen muss.

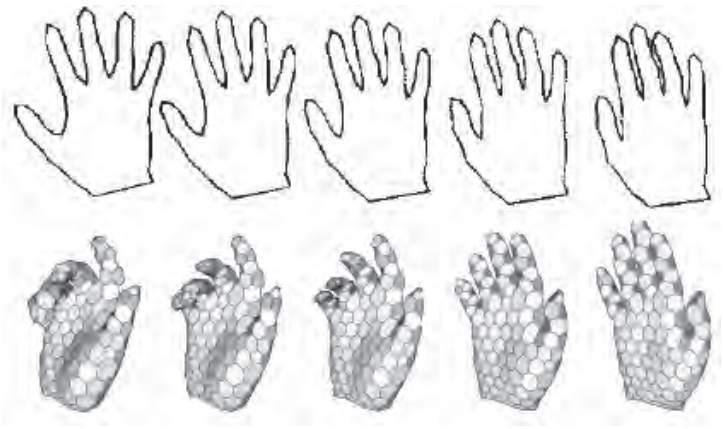

Abbildung 4.2: Berechnete Deformationsmöglichkeiten der Hand. Die obere Reihe zeigt Variationen von 2D-Konturen der Hand [CTCG95], die untere Reihe zeigt mögliche Deformationen eines 3D-Oberflächenmodells der Hand [HH97].

4.2 Beschreibung der Zeigegeste

Der erste Schritt des Verfahrens ist die Beschreibung der Zeigegeste und die entsprechende Erstellung des Trainingsdatensatzes für das Punktverteilungsmodell. Dieser Teil des Verfahrens ist sehr zeitaufwendig, da offensichtlich eine große Anzahl von Trainingsgesten interaktiv erstellt werden muss, um sicherzustellen, dass semantisch äquivalente Stützpunkte in jeder Trainingsgeste an der richtigen Stelle zum Liegen kommen. Die Grundform der Zeigegeste ermittelt sich dabei aus der späteren Anwendung und dem gewählten Kameraaufbau. Praktisch bedeutet dies, dass nach der Kalibrierung des Stereokamerasystems repräsentative Bildpaare mit Gesteninformationen aufgenommen werden müssen, in denen möglichst alle während der Interaktion auftretenden Repräsentationen der Zeigegeste vorkommen. Das hier gewählte Modell verwendet die von Menschen am häufigsten verwendetet Zeigegeste, bei welcher der Anwender während des Deutens mit dem Zeigefinger den Daumen abspreizt. Die Umrandung der Geste ist mit insgesamt 26 Stützpunkten mit unterschiedlicher Gewichtung der Punkte realisiert. Wie in Abbildung 4.3 zu sehen ist, sind 26 Stützpunkte ausreichend, um die wesentlichen Eigenschaften der Kontur der Zeigegeste zu beschreiben und genügend Informationen über das Krümmungsverhalten der Kontur bereit zu stellen. Besonders markante Positionen der Stützpunkte innerhalb der Geste, an denen die stärksten Richtungswechsel der Kontur zu erwarten sind, werden als Referenzgebiete der Geste bezeichnet und werden während der späteren Modellanpassung stärker gewichtet, da diese Konturregionen besonders wichtig für die 3D-Parameterbestimmung wie beispielsweise der Zeigerichtung sind. Als Referenzgebiete werden die Spitze des Daumens und des Zeigefingers, der Übergang von Zeigefinger zum Daumen und der Übergang vom Handballen zum Zeigefinger genutzt. Referenzgebiete werden durch drei dicht aufeinander folgende Stützpunkte markiert, die übrigen Stützpunkte werden auf den verbleibenden freien Strecken der Kontur äquidistant gefüllt.

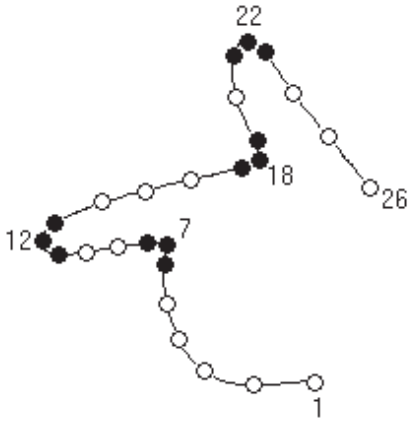

Abbildung 4.3: Beschreibung der Zeigegeste durch 26 Stützpunkte. Schwarz markierte Stützpunkte werden bei der Modellanpassung stärker gewichtet als weiße Stützpunkte.

Um die äußere Form c der Zeigegeste mathematisch zu beschreiben, wird sie durch eine Folge von $n = 26$ ausgewählten Punkten $p_i = (x_i, y_i)$ dargestellt, die auf der äußeren Umrandung der Hand liegen. Damit kann die Kontur einer Geste durch einen 52-dimensionalen Vektor

$$c = (x_1, y_1, x_2, y_2, ..., x_i, y_i, ..., x_n, y_n)^T \tag{4.1}$$

dargestellt werden. Nach einer manuellen Eingabe von Trainingsgesten liegen die einzelnen Punktkoordinaten der Gesten als Pixelwerte, also im Bildkoordinatensystem vor. Damit sind die Trainingsgesten aber noch nicht vergleichbar, da in der statistischen Beschreibung des PDM die Variationen der internen Form, nicht aber der Orientierung und Lage im Bildraum modelliert werden soll. Deshalb müssen die Gesten mit der Norm

$$|c| = \sqrt{\sum_{i=1}^{n}(x_i^2 + y_i^2)} = 1 \tag{4.2}$$

in ein objekteigenes Koordinatensystem transformiert werden. In diesem Koordinatensystem soll nun eine geeignete Transformation $T_{t_x,t_y,\theta,s}$ der einzelnen Konturen c^i gefunden werden, so dass der Abstand der einzelnen Punkte einer Kontur c^1 zu den Punkten einer anderen, ähnlichen Kontur c^2 minimal wird. Die Angleichung der Konturen erreicht man, indem die Transformationsfunktion

$$T_{t_x,t_y,\theta,s}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} s\,cos\theta & -s\,sin\theta \\ s\,sin\theta & s\,cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{4.3}$$

mit geeigneten Werten für die Translation in x- und in y-Richtung t_x, t_y, einer Rotation um den Drehwinkel θ und einer Skalierung um den Faktor s auf jeden einzelnen Punkt einer Kontur c^2 angewendet. Damit ergibt sich die Minimierungsfunktion

$$E = \sum_{i=1}^{n}(c_i^1 - Rc_i^2 - (t_x, t_y)^T)^T \ * \ (c_i^1 - Rc_i^2 - (t_x, t_y)^T) \tag{4.4}$$

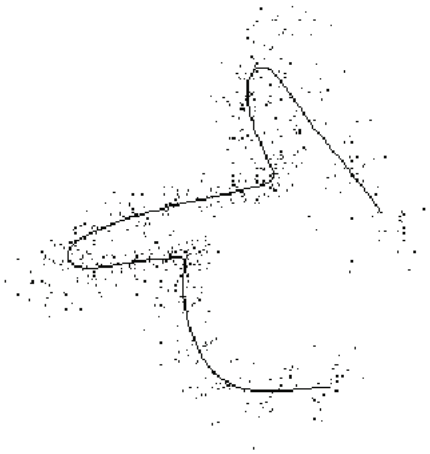

Abbildung 4.4: Punkteverteilung des PDM der Zeigegeste. Gezeigt ist die Mittelwertkontur und die ausgerichteten Einzelpunkte der ersten 25 Trainingskonturen.

mit

$$R = \begin{pmatrix} s\,cos\theta & -s\,sin\theta \\ s\,sin\theta & s\,cos\theta \end{pmatrix}$$

für die n Punkte der Konturen. Minimiert werden soll also die Summe aller Abstandsquadrate der Konturpunkte. Durch die Multiplikation des Abstandsvektors $c_i^1 - Rc_i^2 - (t_x, t_y)^T$ mit seinem transponierten Vektor entsteht außerdem als Ergebnis der Funktion ein skalarer Wert, was den Rechenaufwand erheblich vereinfacht. Dieses Optimierungsproblem für die Minimierung der Fehlerquadrate kann beispielsweise mit dem Levenberg-Marquardt-Algorithmus [Lev44, Mar63] gelöst werden, da die partiellen Ableitungen der Funktion 4.4 analytisch zu bestimmen sind.
Mit dieser allgemeinen Beschreibung zum Ausrichten von zwei ähnlichen Konturen kann nun die Ausrichtung aller Trainingskonturen an die Mittelwertkontur, also das arithmetische Mittel der einzelnen Punktkoordinaten x_i und y_i

$$\bar{c} = (\bar{x}_1, \bar{y}_1, ..., \bar{x}_n, \bar{y}_n) \tag{4.5}$$

mit

$$\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_j^i \qquad \text{und} \qquad \bar{y}_j = \frac{1}{m}\sum_{i=1}^{m} y_j^i \tag{4.6}$$

mit dem folgenden Algorithmus [CTCG95] erstellt werden:

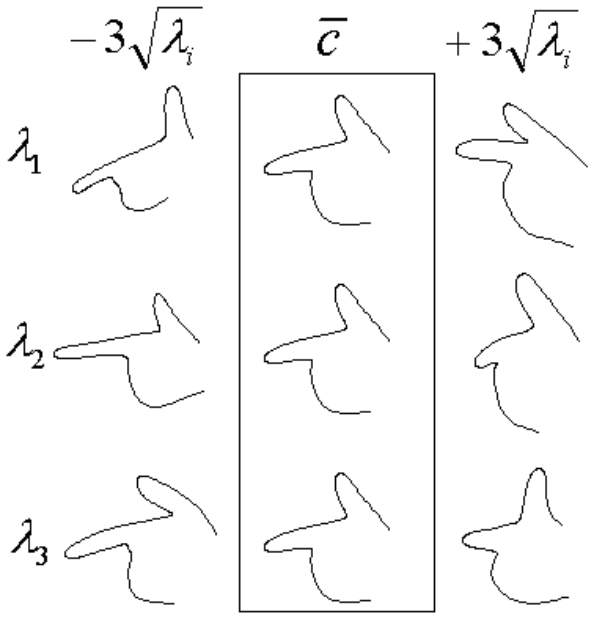

Abbildung 4.5: Variationsmöglichkeiten der Zeigegeste. Dargestellt sind die Grenzen der als gültig erklärten Gesten von $-3\sqrt{\lambda_i}$ bis $+3\sqrt{\lambda_i}$ und die Mittelwertgeste $\bar{c}$ (Spalten) jeweils für die drei größten Eigenwerte λ_1 bis λ_3 (Zeilen).

Algorithmus 1 : Näherungsweises Ausrichten der Trainingskonturen

1: Transliere, rotiere und skaliere paarweise jede der m Trainingskonturen c^i für $i = 2, 3, ..., m$, so dass alle Konturen bestmöglich mit c^1 übereinstimmen. Daraus folgt eine Menge von Konturen $\{c^1, \hat{c}^2, \hat{c}^3, ..., \hat{c}^m\}$.

2: Berechne den Mittelwert $\bar{c}$ der transformierten Konturen.

3: Transliere, rotiere und skaliere die berechnete Mittelwertkontur $\bar{c}$, so dass $\bar{c}$ bestmöglich mit c^1 übereinstimmt. Daraus folgt der korrigierte Mittelwert $\bar{c}'$.

4: Transliere, rotiere und skaliere $\hat{c}^2, \hat{c}^3, ..., \hat{c}^m$, so dass jede Kontur $\hat{c}^i$ bestmöglich mit $\bar{c}'$ übereinstimmt.

5: Wiederhole ab Schritt 2, bis die Mittelwertkontur konvergiert, also zwei aufeinander folgende Mittelwertskonturen sich um weniger also ein vorgegebenen ϵ unterscheiden.

4.3 Berechnung der Deformationsmöglichkeiten

Das Ergebnis dieser vorbereitenden Schritte sind m gegenseitig ausgerichtete Konturen der Zeigegeste $\hat{c}^1, \hat{c}^2, ..., \hat{c}^m$ mit ihrer Mittelwertkontur $\bar{c}$. Durch den Vergleich der Einzelpunkte

mit dem Mittelwert lässt sich für jede Kontur c^i ein Differenzvektor

$$dc^i = \hat{c}^i - \bar{c} \tag{4.7}$$

bestimmen. Um die Abhängigkeit der einzelnen Komponenten in den Konturen zu beschreiben, lässt sich nun für je zwei Koordinatenreihen der Konturen die Kovarianz und damit der semantische Zusammenhang der beiden Merkmale ermitteln. Eine positive Kovarianz bedeutet dabei, dass hohe Werte des einen Merkmals auch hohen Werten des anderen Merkmals entsprechen. Eine negative Kovarianz bedeutet dagegen, dass hohe Werte für das erste Merkmal niedrigen Werten für das zweite Merkmal entsprechen. Aus den Kovarianzen der Differenzvektoren lässt sich nun die Kovarianzmatrix

$$S = \frac{1}{m} \sum_{i=1}^{m} dc^i dc^{i^T} \tag{4.8}$$

der Größe $2n \times 2n$ aufstellen. Diese symmetrische Matrix beschreibt also die Abhängigkeit der einzelnen Konturkoordinaten untereinander. Da hier insbesondere die Abhängigkeiten zwischen den Wahrscheinlichkeitsverteilungen der Punktkoordinatenreihen interessieren, um die interne Konturdeformationen des Trainingsgesten zu beschreiben, werden die Daten mittels einer Hauptkomponentenanalyse (*Principal Component Analysis*, PCA) dekorreliert [Jol02]. Aus der quadratischen Kovarianzmatrix S bestimmt man die Eigenwerte λ_i und Eigenvektoren e_i für die Hauptkomponentenanalyse, indem die Gleichung

$$Se_i = \lambda_i e_i \tag{4.9}$$

gelöst wird. Mit der Matrix

$$P = \begin{pmatrix} e_1 & \dots & e_{2n} \end{pmatrix}, \tag{4.10}$$

deren Spalten die Eigenvektoren von S enthalten, existiert also für jeden Vektor c (und damit auch für jede beliebige Darstellung einer Zeigegeste in Vektorschreibweise) ein Deformationsvektor b, für den

$$c = \bar{c} + Pb \tag{4.11}$$

gilt. Die einzelnen Komponenten von b sind also ein Maß dafür, wieviel Variation für den entsprechenden Eigenvektor und damit für eine bestimmte Koordinate des Modells benötigt wird, um c zu erreichen. In der Anwendung bedeutet dies, dass durch geeignete Wahl der Werte für b die Mittelwertskontur $\bar{c}$ zur einer neuen Kontur c deformiert werden kann. Ist eine beliebige Kontur c vorgegeben, kann durch Auflösung der Gleichung (4.11) nach b das Maß der Deformation bestimmt und somit überprüft werden, ob eine Kontur eine gültige Zeigegeste beschreibt. Dafür müssen geeignete Grenzen für die Komponenten b_i des Deformationsvektors definiert werden. Da die Varianz von b_i von dem entsprechenden Eigenwert λ_i abhängt, können diese Grenzen durch die Varianz um den Mittelwert interaktiv bestimmt werden. Für das hier beschriebene Modell der Zeigegeste hat sich

$$-3\sqrt{\lambda_i} \;\leq\; b_i \;\leq\; 3\sqrt{\lambda_i}, \tag{4.12}$$

als praktikabel erwiesen (siehe Abbildung 4.5). Deformationsvektoren mit Komponenten b_i innerhalb dieser Grenzen erzeugen also unter Verwendung der Gleichung 4.11 eine gültige neue Kontur einer Zeigegeste, während Gesten als ungültig deklariert werden müssen, falls bei einer Anpassung an die Bildinformationen das Ergebnis eine Kontur ist, dessen Deformationsvektor b mindestens eine Komponente außerhalb dieses Intervall enthält.

Eine Möglichkeit, den hohen Rechenaufwand drastisch zu senken, liefert die Tatsache, dass in der Regel nur wenige Eigenwerte und Eigenvektoren den größten Teil der Variationsmöglichkeiten abdecken. Ordnet man die Eigenwerte der Größe nach, so dass $\lambda_i > \lambda_{i+1}$ ist und berechnet die Gesamtsumme aller Eigenwerte λ_{total}, lässt sich ein Index t berechnen, für den

$$\sum_{i=1}^{t} \lambda_i \geq \alpha * \lambda_{total} \qquad 0 \leq \alpha \leq 1 \tag{4.13}$$

gilt. Der Wert α gibt dabei an, welchen Anteil die Summe der ersten t Eigenwerte an λ_{total} ausmachen. Zusätzlich zur Sortierung der Eigenwerte müssen auch die entsprechenden Eigenvektoren e_i in der Matrix P vertauscht werden. Dies ist möglich, da die Eigenvektoren linear unabhängig sind. Daraus ergibt sich eine Reduktion der Spaltenzahl der Eigenvektormatrix auf t Spalten:

$$P_t = \begin{pmatrix} e_1 & ... & e_t \end{pmatrix} \tag{4.14}$$

und auch eine Verkleinerung des Deformationsparameters

$$b_t = (b_1, b_2, ..., b_t)^T \tag{4.15}$$

Dieses kompaktere und schneller zu berechnende Modell erlaubt nun auch eine Näherung der Gleichung (4.11):

$$x \approx \bar{x} + P_t b_t. \tag{4.16}$$

Die Wahl des Parameters α und damit auch der Größe des reduzierten Systems bestimmt also, wie viel Variation, die im Trainingsdatensatz tatsächlich vorhanden ist, im Modell wiedergegeben werden kann. Für das Modell der Zeigegeste decken schon die ersten 5 Eigenwerte mehr als 95% und die ersten 10 Eigenwerte mehr als 98% der Variationsmöglichkeiten ab (siehe Tabelle 4.1).

Abbildung 4.6: Anpassungsergebnisse bei unterschiedlichen Initialkonturen. Links: Anpassung gelingt durch gute Wahl der Startkontur für das ASM. Rechts: Anpassung misslingt durch zu große Translation der Startkontur.

Index i	$(\lambda_i/\lambda_{total}) * 100$	Kumulierte Häufigkeit
1	73.2	73.2
2	10.3	83.5
3	6.0	89.5
4	3.6	93.2
5	2.1	95.2
6	1.0	96.3
7	0.8	97.0
8	0.5	97.5
9	0.5	97.9
10	0.4	98.4

Tabelle 4.1: Relative und kumulierte Häufigkeiten der größten 10 Eigenwerte

Für das hier vorgestellte Punktverteilungsmodell der Zeigegeste hat sich die Reduktion auf acht Dimensionen und damit einen Abdeckung von mehr als 97% der Variationsmöglichkeiten als ausreichend erwiesen und erfüllt somit auch den Echtzeitanspruch des Verfahrens für eine intuitive Interaktion.

4.4 Finden einer Startpose

Aktive Formen verlangen für die Anpassung einer Kontur an die realen Bildinformationen eine gute initiale Schätzung von Position, Orientierung und Skalierung der Geste (siehe Abbildung 4.6). Dieser Schritt des Verfahrens steht aber im Gegensatz zu der gewünschten intuitiven Interaktion, für die das gesamte Verfahren eingesetzt werden soll. Dem Anwender darf nicht zugemutet werden, dass zur Anwendung selbst, mit der er interagieren möchte, ständig die mit den Konturen überlagerten Kamerabilder eingeblendet werden, um eine visuelle Kontrolle der Anpassung zu ermöglichen. Typische Handtracker, die auf dem Verfahren der aktiven Formen beruhen, setzen aber genau dies voraus. Insbesondere beim Starten des Systems wird in der Regel die Mittelwertgeste im Kamerabild eingeblendet und der Anwender wird

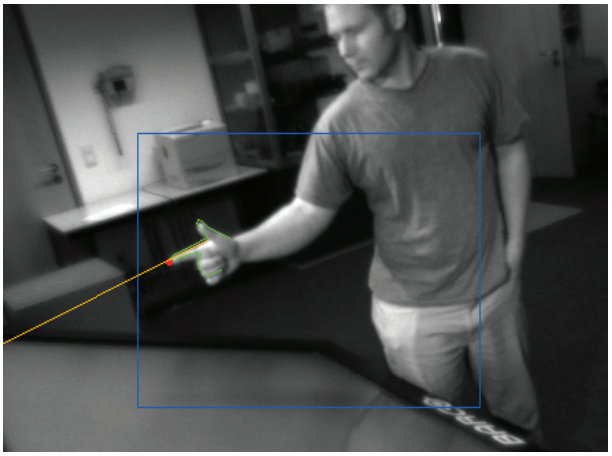

Abbildung 4.7: Zeigegeste liegt zwischen Anwender und Ausgabegerät. Eine geeignete Wahl der Position und Orientierung der Kamera lässt somit Aussagen über die Position der Geste in der Bildebene zu.

dazu aufgefordert, seine Hand entsprechend im Raum zu positionieren, also die Initialkontur wie einen virtuellen Handschuh anzuziehen. Da intuitive Interaktion aber voraussetzt, dass das Verfahren vollautomatisch abläuft und der Anwender sich nicht um technische Belange der Gestenerkennung kümmern darf, muss für das Auffinden der Startkonturen für das ASM ein automatisierter Weg gefunden werden. In dem hier beschriebenen Verfahren geschieht dies durch

1. eine initiale Bestimmung der Fingerspitze durch eine geeignete Wahl des Aufbaus des Stereokamerasystems und

2. durch das heuristische Optimierungsverfahren *Simulated Annealing*.

4.4.1 Initiale Bestimmung der Fingerspitze

Für die Wahl des Aufbaus der Kameras ist hauptsächlich die Orientierung der Kameras von Interesse. Die Variationsmöglichkeiten für die Positionierung der Kameras ist bereits durch die Verwendung des Punktverteilungsmodells teilweise stark eingeschränkt, da zu jedem Zeitpunkt des Verfahrens genügend Informationen der Zeigegeste in den Kamerabildern verfügbar sein müssen. Beispielsweise muss vermieden werden, dass der Zeigefinger, der für die Berechnung der 3D-Zeigerichtung benötigt wird, in einem der beiden Kamerabildern zu stark verkürzt wird, wenn die Hand in Richtung der Kamera gedreht wird. Gleiches gilt für die Skalierung der Kameras, die zum einen durch den Abstand der Kameras zum Objekt, zum anderen durch die gewählte Brennweite der Kameraobjektive definiert ist. Ist die Geste zu klein im Bild zu sehen, geht durch die geringe Bildinformation Präzision für die spätere 3D-Rekonstruktion verloren. Auf der anderen Seite darf aber auch die Hand nicht zu groß im Bild erschienen, um zu vermeiden, dass die Hand während der Interaktion versehentlich aus dem Kamerabild und damit aus dem Interaktionsvolumen bewegt wird.

Die Orientierung der Kameras hingegen kann uneingeschränkt dazu verwendet werden, das Auffinden einer Startpose zu erleichtern. Man kann sich die einfache Annahme zunutze machen, dass eine Zeigegeste während der Interaktion immer vor dem Körper des Anwenders stattfindet. Insbesondere die Tatsache, dass die Interaktion durch die Darstellung der Anwendung geprägt ist und somit der Anwender während der Interaktion immer dem Ausgabegerät zugewandt ist, erleichtert Aussagen über die Position der Hand in den Kamerabildern zu treffen. Bei einer Ausrichtung der Kameras wie sie beispielsweise in den Abbildungen 4.1 und 4.7 zu sehen ist, liegt eine zur Interaktion verwendete Zeigegeste in den Bildern der ersten Kamera links vom Anwender und entsprechend rechts vom Anwender in den Bildern der zweiten Kamera. Für das hier vorgestellte Verfahren wird diese Tatsache bei geeigneter Wahl der Positionierung und Orientierung des Kameras verwendet, um eine erste Abschätzung der Lage der Zeigegeste im Bildkoordinatensystem zu ermitteln. Dazu werden beim Start des Systems, wenn sichergestellt werden kann, dass sich zur Zeit kein Anwender im Interaktionsvolumen befindet, für beide Kameras Referenzbilder der Szene aufgenommen und deren Kanteninformation extrahiert. Diese Kanteninformationen werden als Sobel-Operator über eine Filterung der Luminanzbilder im Ortsbereich mit den horizontalen und vertikalen Filterkernen

$$H_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \text{und} \quad H_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{4.17}$$

berechnet. Der Sobel-Kantenfilter liefert genügend breite Kanten und stellt somit sicher, dass bei der folgenden Segmentierung des Kantenbildes an einem vorgegebenen Schwellwert ausreichend große Segmente entstehen. Zur Laufzeit des Systems werden für neue Kamerabilder ebenfalls die Kantenbilder ermittelt und segmentiert. Als nächstes werden aus den segmentierten Referenzkantenbildern und Laufzeitkantenbildern Differenzbilder berechnet, die erneut segmentiert werden, um Information über Änderungen in der Szene zu erhalten. Da diese Änderungen durch die Verwendung von Referenzbildern immer bezüglich des leeren Interaktionsvolumens berechnet werden, kann davon ausgegangen werden, dass die entstehenden Segmente durch die Anwesenheit des Anwenders entstehen. Durch die Wahl des Kameraaufbaus setzt das Verfahren also lediglich voraus, dass sich zwischen Anwender und Ausgabegerät seit dem Start der Anwendung keine neuen Objekte befinden, die eine Interaktion stören können. Die Analyse der Differenzbilder verwendet das jeweils am weitesten links beziehungsweise rechts liegende Segment eines Bildes und ermittelt den Startpunkt des Segments in der ersten bzw. letzten Spalte innerhalb des umschließenden Rechtecks (engl.: *bounding box*). Dieser Punkt der Bildebene wird als Fingerspitze der Zeigegeste angenommen und für den folgenden Optimierungsschritt verwendet.
Da es dennoch vorkommen kann, dass sich in den Interaktionsbereichen der Kamerabilder Segmente befinden, die weiter links oder rechts von der gewünschten Segmenten mit der Geste befinden, wird vor dem nächsten Schritt des Verfahrens eine Überprüfung mittels des Stereokamerasystems vorgenommen. Dafür werden die beiden ermittelten Punktkoordinaten zu einem 3D-Punkt rekonstruiert und anhand der Definition des Weltkoordinatensystems überprüft, ob der entstandene 3D-Punkt eine logische Zuordnung als Fingerspitze zulässt.

Ist dies nicht der Fall, wird abwechselnd für beide Differenzbilder ein Segment eliminiert und die Fingerspitzenrekonstruktion erneut durchgeführt.

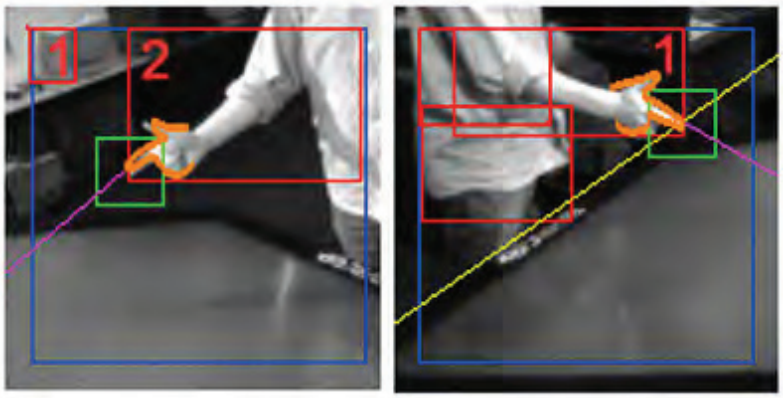

Abbildung 4.8: 3D-Rekonstruktion und Überprüfung der Segmentinformationen zur ersten Bestimmung der Fingerspitze.

Abbildung 4.8 zeigt dieses Verfahren an einem Beispiel: Im linken Kamerabild ist "fälschlicherweise" in der linken oberen Ecke des blau markierten Interaktionsbereichs ein Segment (rot markiert mit einer '1') aufgetaucht, dass zunächst als richtiges, die Geste enthaltendes Segment angenommen wird. Im rechten Kamerabild enthält das am weitesten rechts liegende Segment bereits die Zeigegeste. Für beide Segmente wird nun der äußerste Punkt des Segments (also der Punkt, an dem das Segment sein umschließendes Recheck an der linken bzw. rechten Seite berührt) bestimmt und im Stereokamerasystem zu einem 3D-Punkt rekonstruiert. Da der Ergebnispunkt im Weltkoordinatensystem aber außerhalb des Interaktionsvolumens liegt, wird im linken Kamerabild das erste Segment eliminiert und statt dessen das mit einer '2' markierte Segment verwendet. Die erneute 3D-Rekonstruktion ergibt nun einen Punkt, der innerhalb des Interaktionsvolumens liegt und akzeptiert wird. Damit ist die erste Bestimmung der Fingerspitze erfolgt.

4.4.2 Simulated Annealing

Mit einer ersten Abschätzung der Fingerspitze kann nun in einem weiteren Schritt des Verfahrens eine grobe Schätzung der gesamten Zeigegeste ermittelt werden. Der Suchraum für die Suche nach der Kontur der Zeigegeste ist aber durch das Wissen über die Position der Fingerspitze in den Kamerabildern bereits stark eingeschränkt und ermöglicht somit eine schnelle und beinahe echtzeitfähige Suche der Geste. Als Verfahren zum Auffinden der Zeigegeste wird das aus dem Metropolisalgorithmus [MRR+53] entwickelte Verfahren der simulierten Abkühlung (engl.: *Simulated Annealing*, SA) eingesetzt. Simulated Annealing [KGV83] ist ein heuristisches Verfahren der globalen nichtlinearen Optimierung, das zur approximativen Lösung von komplexen Optimierungsproblemen verwendet wird. Ziel der Optimierung ist beispielsweise die Minimierung einer Zielfunktion $E(X)$, wobei X ein mehrdimensionaler

Variablenvektor der zu minimierenden Funktion ist. Ein initialer Vektor x_0 wird über viele Iterationsschritte so geändert, dass in jedem Iterationsschritt einen neuer Lösungsvektor in der Nachbarschaft der Zielfunktion zufällig gewählt wird. Die Veränderung der Variablen von x_{i-1} nach x_i kann entweder zu einer Verbesserung oder zu einer Verschlechterung des Ergebnisses der Zielfunktion führen. Um zu vermeiden, dass der Algorithmus nun in einem lokalen Minimum stecken bleibt, sind auch Verschlechterungen der Zielfunktion zugelassen. Die Möglichkeit, eine Verschlechterung zu akzeptieren, sinkt dabei im Verlauf des Verfahrens bis am Ende der Optimierung nur Verbesserungen akzeptiert werden. Um diese Wahrscheinlichkeit zu modellieren, wird eine Temperaturschranke T eingeführt, die nach jeweils n Iterationsschritten um einen vorgegebenen Faktor δt abgesenkt wird. $E(x_i)$ wird immer neue Lösung akzeptiert, wenn $E(x_i) < E(x_{i-1})$ gilt. Ansonsten entstehen zwei Möglichkeiten für X: Entweder x_i wird mit der Wahrscheinlichkeit $P(x_i)$ akzeptiert oder abgelehnt mit $1 - P(x_i)$. P wird dabei entsprechend der Boltzmann-Statistik

$$P(\Delta E) = \exp\left(\frac{-\Delta E}{k_B T}\right) \tag{4.18}$$

mit $\Delta E = E(x_i) - E(x_{i-1})$ und der Boltzmannkonstante k_B berechnet. Damit ist zum einem die Akzeptanzschwelle für Verschlechterungen von dem Betrag der Verschlechterung abhängig, zum anderen vom zeitlichen Ablauf der Optimierung. Gegen Ende des Verfahrens, wenn T klein genug geworden ist, werden Verschlechterungen nur noch selten akzeptiert und es werden fast ausschließlich nur noch Verbesserungen der Zielfunktion in Richtung des nächsten lokalen Optimums angenommen.
Für die Suche nach der Zeigegeste in einem Kamerabild müssen nun noch die Zielfunktion und die Änderungsschritte im Variablenvektor von x_{i-1} nach x_i definiert werden. Die Zielfunktion ist in diesem Fall die Summe der Kantenintensitäten im Kamerabild für eine künstlich erzeugte und im Bild positionierte Kontur einer Zeigegeste. Dazu wird anhand des im Abschnitt 4.2 beschriebenen Punktverteilungsmodells der Zeigegeste eine gültige Kontur erzeugt und im Kamerabild positioniert, orientiert und geeignet skaliert. Der skalare Ergebniswert der Zielfunktion ergibt sich dann als

$$E(x_i) = \sum_{p_{x,y}} 255 - S_{x,y} \tag{4.19}$$

für alle Stützpunkte $p_{x,y}$ der Kontur und den entsprechenden Kantenintensitäten $S_{x,y}$ des zuvor berechneten Sobel-Kantenfilters mit dem Luminanzwert 255 als maximal darstellbaren Kantenwert.
Die zu optimierenden Unbekannten von E sind somit sowohl durch eine Position t_x, t_y, eine Orientierung α und eine Skalierung s der Kontur im Bildkoordinatensystem als auch durch die Komponenten des Deformationsvektors b der Kontur definiert und können als Vektor

$$x_i = (t_x, t_y, \alpha, s, b_1, ..., b_n)^T. \tag{4.20}$$

geschrieben werden. Als initialer Vektor x_0 wird für b die Mittelwertgeste des PDM gewählt, die Position der Geste ergibt sich aus der zuvor beschriebenen Bestimmung der Fingerspitze,

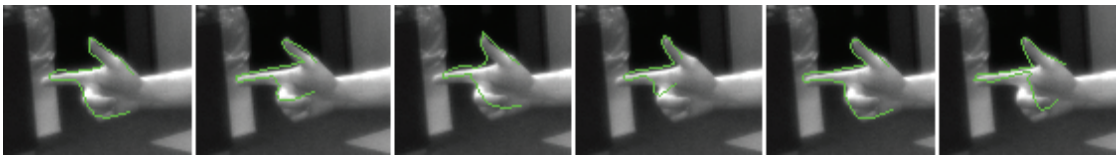

Abbildung 4.9: Finden der Zeigegeste mit *Simulated Annealing*. Sechs zufällige Ergebnisse des Verfahrens bei gleichen Ausgangssituationen und kurzem Auskühlschema.

die initiale Orientierung und Skalierung der Geste ergibt sich aus dem Aufbau des Stereokamerasystems. Eine zufällige Veränderung des Vektors ergibt sich durch eine zufällige Wahl einer der Komponenten von x_i und deren zufälligen Veränderung innerhalb eines gültigen Intervalls. Die Intervallgrenzen für die Komponenten b_i werden dabei aus dem Punktverteilungsmodell entnommen, die Intervallgrenzen für die Translation, Rotation und Skalierung der Geste aus dem Aufbau des Kamerasystems.
Neben der Definition der Lösungsraumstruktur ist für die Anwendung von Simulated Annealing die Festlegung des Abkühlungsschemas notwendig. Eine größere Anzahl von Iterationsschritten erhöht zwar die Genauigkeit der gefundenen Lösung, verbraucht aber auch wesentlich mehr Berechnungszeit. Meist wird ein Abkühlungsschema verwendet, bei dem der Temperaturparameterwert im Verlauf des Verfahrens regelmäßig durch die Multiplikation mit einer Zahl kleiner eins verringert wird. Dabei ist auch der initiale Temperaturwert und der Abbruch des Verfahrens festzulegen [JAMS89, PTVF92]. Um den Echtzeitanspruch der Anwendung nicht zu gefährden, wird für die Zeigegeste ein sehr schnelles Abkühlschema mit nur einigen tausend Iteration verwendet. Dies ist möglich, da zum einen der Suchraum durch die Bestimmung der Fingerspitze bereits stark eingeschränkt ist und zum anderen das Verfahren nicht zur Anpassung der Zeigegeste selbst verwendet wird, sondern ausschließlich eine geeignete Startkontur für das folgende *Active Shape Model* liefern soll. Ein Schema mit maximal 25 Temperaturschritten, bis zu 700 Iterationen pro Temperaturschritt und einem Absenken der Temperatur um 20% bei jedem Absenken der Temperatur liefert bei über 99% der Versuche eine für den folgende Anpassungsschritt ausreichende Startpose (Beispiele siehe Abbildung 4.9) und benötigt im Schnitt unter 80 Millisekunden für die Suche.

4.5 Feinanpassung der Zeigegeste

Die durch *Simulated Annealing* gewonnenen Posen dienen in einem weiteren Schritt des Verfahrens als Startwerte für die Feinanpassung der Zeigegesten durch Aktive Formen. Für die Anpassung innerhalb der vom PDM vorgegebenen Deformationsgrenzen werden iterativ die Stützpunkte der Kontur bestmöglich an die Kanteninformation der Bilder gezogen. Dafür werden folgende Parameter benötigt:

- Die Mittelwertgeste $\bar{x}$, die bereits aus dem PDM bekannt ist
- Die reduzierte Deformationsmatrix P_t, die bereits aus dem PDM bekannt ist
- Die bestmöglichen Lageparameter der Kontur (Translation, Rotation und Skalierung)

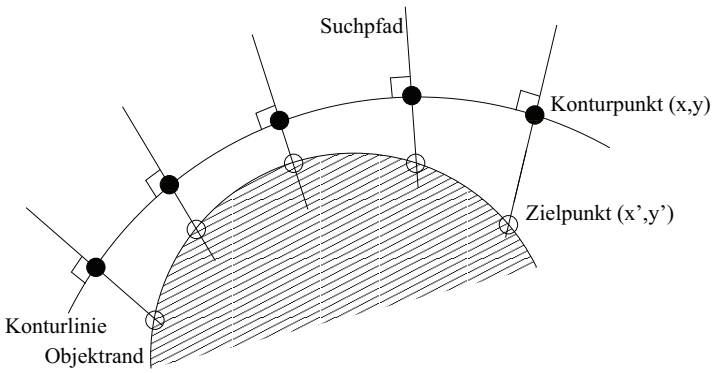

Abbildung 4.10: Suche der Zielpunkte entlang der Normalgeraden

- Der bestmögliche Deformationsvektor b_t der Kontur

Die Bestimmung der Lageparameter und des Deformationsvektors der Kontur ist erneut als ein Optimierungsproblem aufzufassen, dass prinzipiell mittels nichtlinearen Optimierungsverfahren gelöst werden kann. Allerdings ist der Rechenaufwand der gängigen Verfahren zu hoch, um eine akzeptable Lösung innerhalb einer Zeit zu ermitteln, die den Echtzeitanspruch der hier vorgestellten Anwendung erfüllt. Cootes beschreibt eine approximative Lösung, die iterativ, die Stützpunkte der Kontur in Richtung der stärksten Kanteninformation verschiebt und Lageparameter und Deformationsvektor in jedem Iterationsschritt angleicht. [Tay92, CT01]:

Algorithmus 2 : Anpassung einer Aktiven Form

1: Initialisiere eine Startpose (hier als Ergebnis der Suche mittels *Simulated Annealing*).

2: Für jeden Stützpunkt der Kontur suche entlang der Normalvektoren nach dem Pixel mit der höchsten Kantenintensität. Berechne den Verschiebungsvektor dc der Kontur.

3: Korrigiere die Lageparameter der Kontur, so dass die neu entstandene Kontur bestmöglich mit der vorigen Kontur übereinstimmt.

4: Bestimme den Deformationsvektor der Kontur, der bestmöglich eine gültige Kontur auf die Zielpunkte im Bild beschreibt.

5: Korrigiere den Deformationsvektor der Kontur entsprechend: $b_t + db_t$.

6: Wiederhole ab Schritt 2 bis Änderungen der Kontur vernachlässigbar klein sind.

Schritt 2: Da das Modell der Zeigegeste durch ihre äußere Umrandung beschrieben ist, muss eine gute Position für einen Konturpunkt einem Randpunkt der Geste im Bild entsprechen.

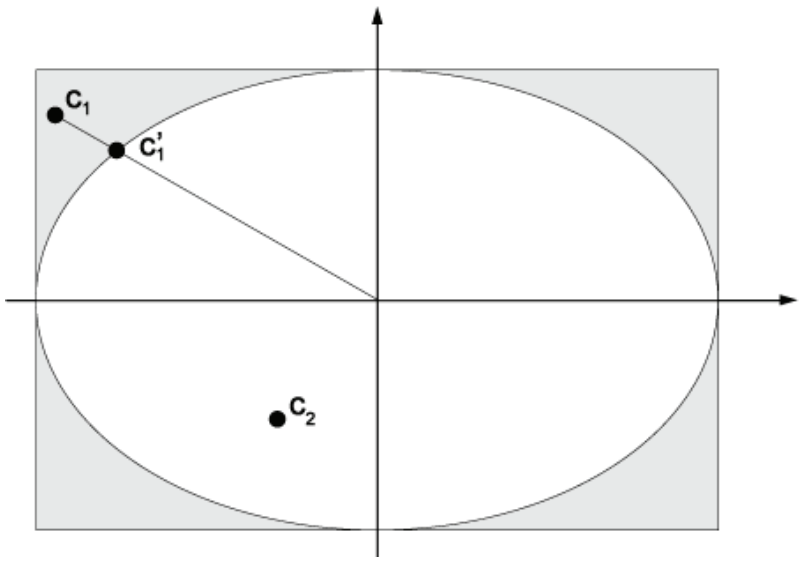

Abbildung 4.11: Korrektur einer Pose durch die Mahalanobis-Distanz als 2D-Beispiel. Vektor c_2 ist gültig, Vektor c_1 liegt außerhalb der Ellipse und wird auf den Rand der Ellipse abgebildet um eine gültige Pose c_1' zu erzeugen.

Um eine bessere Position für einen Konturpunkt zu finden, definiert man einen Suchpfad, der durch den jeweiligen Stützpunkt verläuft. Eine schnelle Berechnung des Suchpfades erhält man, wenn die Tangente der Kontur an der Position einer Stützstelle aus den Nachbarpunkten der Kontur approximiert wird und der dazu orthogonale Vektor in beide Richtungen der Kontur verwendet wird. Entlang dieses Suchpfades wird nun der Bildpunkt mit der höchsten Kantenintensität ermittelt (siehe Abbildung 4.10). Kann entlang des Suchpfades kein sinnvoller neuer Punkt gefunden werden, weil keine ausreichende Kanteninformation gegeben ist, wird der Stützpunkt nicht verschoben, sondern an seiner ursprünglichen Position belassen.

Schritt 3: Die neue Kontur, die durch die Verschiebungen entlang der Suchpfade entstanden ist, muss nun an die ursprüngliche Kontur angepasst werden. Um die dafür notwendigen Transformationen t_x, ty, α, s zu ermitteln, kann der in Abschnitt 4.2 beschriebene Algorithmus zum "Näherungsweisen Ausrichten der Trainingskonturen" verwendet werden.

Schritt 4: Die Verschiebung der Stützpunkte in Schritt 2 des Algorithmus geschieht allein über das Kriterium der Kantenintensitäten im Kamerabild. Damit sind als Ergebnis dieses Schrittes Deformationen der Kontur möglich, die nicht durch die statistische Beschreibung des PDM abgedeckt sind.

Schritt 5: Für den Fall, dass die Kontur nicht gültig ist, also mindestens eine der Komponenten des zur Kontur gehörigen Deformationsvektors b_t außerhalb der vorgegebenen Intervallgrenzen von $-3\sqrt{\lambda_i}$ und $+3\sqrt{\lambda_i}$ liegt, muss die bestmögliche Kontur innerhalb der vorgegebenen Intervallgrenzen gefunden werden. Ein geeignetes Kriterium für diesen Schritt

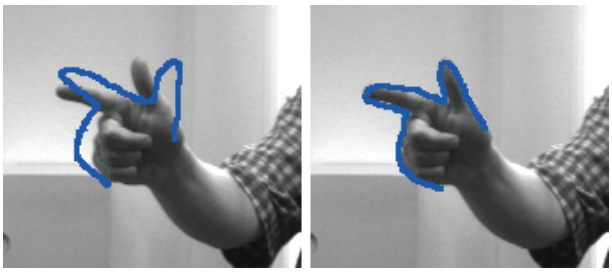

Abbildung 4.12: Initiale (links) und angepasste (rechts) Kontur auf Bildebene.

ist die Mahalanobis-Distanz [Mah36, Tay92]

$$D_m = \sum_{i=1}^{t} \left(\frac{b_i^2}{\lambda_i} \right) \leq D_{max} \tag{4.21}$$

des Konturvektors. In einem t-dimensionalen Raum existiert ein Hyperellipsoid, dessen Radien durch die Eigenwerte λ_i bestimmt werden. Eine Kontur c ist demnach gültig, wenn der Vektor b_t innerhalb dieses Hyperellipsoids liegt. Als zulässige Grenze gilt dann beispielsweise für die 3σ-Intervalle, dass die Mahalanobis-Distanz kleiner $D_{max} = 3.0$ sein darf. Liegt eine Kontur außerhalb des Ellipsoids und ist damit ungültig, kann sie dadurch in eine gültige Kontur umgewandelt werden, indem man den Deformationsvektor b_t so verändert, dass die Kontur genau auf dem Rand des Hyperellipsoids zum Liegen kommt (siehe Abbildung 4.11):

$$b_i \rightarrow b_i * \frac{D_{max}}{D_m} \quad i = 1, ..., t \tag{4.22}$$

Konvergenz in **Schritt 6** bedeutet, dass sich nach einem weiteren Iterationsschritt die neu berechnete Kontur für alle Koordinaten um weniger als ein Pixel unterscheidet, also kein sichtbarer Unterschied zwischen zwei aufeinanderfolgenden Konturen existiert.

Nach Beendigung der Anpassung durch Abbruch des Algorithmus nach dem in Schritt 6 beschriebenen Konvergenzkriteriums kann die Güte der Anpassung durch Auswertung der mittleren Abweichung der entstandenen Konturpunkte von der realen Kante im Bild gemessen werden. Diese Abweichung entsteht, da die Korrektur der Gestenkontur in Schritt 5 der Anpassung die Stützpunkte erneut von der maximalen Kantenintensität entfernt. Zur Berechnung der Abweichung werden noch ein letztes mal die Suchpfade der angepassten Kontur berechnet und deren Stützpunkte erneut in Richtung der maximalen Kantenintensitäten verschoben. Das arithmetische Mittel der euklidischen Distanzen der einzelnen Stützpunkte der beiden Konturen wird nun als Gütekriterium für die Anpassung verwendet. Als geeigneter Grenzwert hat sich ein Mittelwert von einem Pixel bei dem in diesem Verfahren verwendeten Modell der Zeigegeste mit 26 Stützpunkten erwiesen, über dem eine Anpassung mit dem ASM verworfen werden sollte. Tabelle 4.2 zeigt die mittlere Abweichung zwischen den finalen Konturen für zehn zufällig ausgewählte Anpassungen mittels eines ASM.

Abbildung 4.13: Beispiele der Feinanpassung mittels des *Active Shape Model*.

Anpassung	mittlere Abweichung [Pixel]	Dauer der Anpassung [msec]
1	0.39	5.0
2	0.38	5.3
3	0.12	5.3
4	0.29	4.5
5	0.15	5.2
6	0.29	3.8
7	0.36	8.1
8	0.23	5.1
9	0.48	4.4
10	0.11	5.2

Tabelle 4.2: Ergebnisse von zehn zufälligen, gültigen Feinanpassungen der Zeigegeste mittels ASM.

Für den Fall, dass die Anpassung der Kontur das Gütekriterium erfüllt, kann die entstandene Kontur akzeptiert und für die folgende 3D-Parameterschätzung verwendet werden. Überschreitet hingegen die mittlere Abweichung den Schwellwert von einem Pixel, muss die Anpassung verworfen und das Verfahren mit einer erneuten Suche nach der Startkontur mittels *Simulated Annealing* fortgesetzt werden.

4.6 3D-Parameterschätzung

Der letzte Schritt des Verfahrens verwendet nun erneut das kalibrierte Stereokamerasystem, um aus den beiden angepassten Konturen der Zeigegeste für die Interaktion geeignete 3D-Parameter zu bestimmen. Für die Interaktion selber sind die Position und Orientierung der Hand des Anwenders zunächst aber gar nicht relevant. Vielmehr interessiert für die Anwendbarkeit der Punkt im Raum, auf den der Anwender zeigt. Dies ist in den meisten Anwendungen der Schnittpunkt der Strahls im Raum mit der jeweiligen Anzeigefläche, mit welcher den Anwender interagiert. Zur Bestimmung dieses Schnittpunktes müssen neben den 3D-Koordinaten des Ausgabebildschirms außerdem

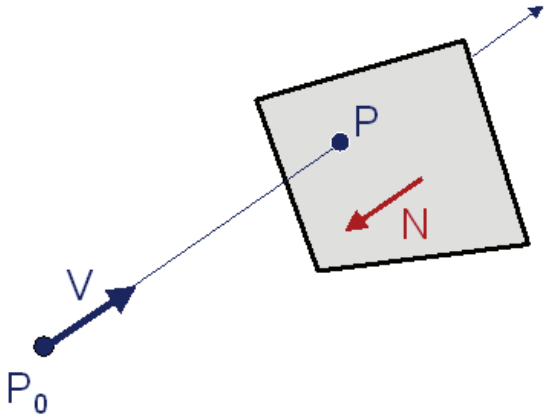

Abbildung 4.14: Berechnung des Schnittpunktes von Zeigestrahl und Ausgabebildschirm.

1. die Position der Geste im Raums und
2. die Zeigerichtung im Raum

bekannt sein. Die Position der Geste ist bereits durch die vorherigen Schritte des Verfahrens bekannt. Wenn als 3D-Position die Fingerspitze der Hand angenommen wird, kann entweder das Ergebnis aus der initialen Bestimmung der Fingerspitze (vergleiche Abschnitt 4.4.1) oder aber auch der Stützpunkt p_{12} der angepassten Kontur verwendet werden. Eine zweite Möglichkeit ist, die Schwerpunkte der angepassten Konturen als arithmetische Mittelwerte der Stützpunkte

$$x_c = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad y_c = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.23}$$

für beide angepassten Konturen zu berechnen und diese Pixelkoordinaten zu verwenden, um den Schwerpunkt der Geste im Raum zu schätzen. Für die Schätzung der Zeigerichtung werden die im letzten Schritt des Verfahrens ermittelten Konturen verwendet. Dabei wird davon ausgegangen, dass die Zeigerichtung allein durch den Zeigefinger bestimmt wird. Deshalb wird aus den korrespondieren Stützpunkten p_8 bis p_{17} der angepassten Konturen (siehe Abbildung 4.3) mittels des Stereokamerasystems eine 3D-Punktewolke aus zehn Punkten berechnet. Die Regressionsgerade dieser Punkte im Raum bestimmt dann die Zeigerichtung im Raum. Eine schnelle Berechnung dieser Geraden ist durch eine orthogonale lineare Regression im Raum durch die bereits beschriebene Hauptachsenanalyse (*Principal Component Analysis*, PCA) möglich [PP07].

Mit dem resultierenden Strahl im Raum $P_0 + tV$ beginnend an der 3D-Position P_0 der Zeigegeste und dem Richtungsvektor V, der algebraischen Beschreibung der Ebene in der sich der Ausgabebildschirm befindet $P \cdot N + d = 0$ mit dem gesuchten Schnittpunkt P, dem Normalvektor der Ebene N und dem Abstand d, lässt sich durch Substitution von P nach

$$(p_0 + tV) \cdot N + d = 0 \tag{4.24}$$

der Schnittpunkt

$$P = P_0 + tV \tag{4.25}$$

mit

$$t = \frac{-(P_0 \cdot N + d)}{(V \cdot N)} \tag{4.26}$$

berechnen (siehe Abbildung 4.14). Die Transformation dieses Schnittpunktes vom Weltkoordinatensystem in ein lokales Koordinatensystem des Ausgabegerätes ermöglicht damit die Verwendung der rekonstruierten Zeigegeste als Eingabemodalität zur Interaktion.

4.7 Echtzeitfähigkeit

Für die Verwendbarkeit des Verfahrens als Eingabemodalität für eine intuitive Interaktion mit dem Rechner müssen wie bereits beschrieben einige Anforderung an das System erfüllt sein. Die Forderung, dass das System ohne Trainingszeit für den Anwender auskommt, ist bei dem hier vorgestellten Verfahren erfüllt, da das Anlernen des Modells der Zeigegeste bereits vorab erfolgt ist. Bei dem verwendeten Modell der Zeigegeste muss lediglich sichergestellt sein, dass der Anwender weiß, wie die Zeigegeste während der Trainingsphase definiert wurde. Einem neuen Anwender des Systems muss also bekannt sein, dass er den Daumen während des Zeigens auf den Bildschirm abspreizen muss. Weiterhin wird durch das automatische Auffinden der Startpose sichergestellt, dass der Anwender sich nicht aktiv um den Start der Gestenverfolgung kümmern muss. Weder zum Starten der Interaktion noch während der Gestenverfolgung selber muss der Anwender den Status oder das Ergebnis der Gestenanpassung kontrollieren.
Eines der wichtigsten Kriterien für eine intuitive Interaktion ist die Echtzeitfähigkeit des Systems. Durch den hohen Anteil an bildverarbeitenden Schritten während des Verfahrens ist für die Verarbeitungszeit insbesondere die Größe des Interaktionsvolumens ausschlaggebend. Das Interaktionsvolumen im Raum bestimmt dabei die Region in den Kamerabildern, die für die einzelnen Schritte des Verfahrens untersucht werden müssen. Bei einem typischen Interaktionsbereich von $300 * 300$ Bildpunkten in beiden Kamerabildern ergeben sich typischerweise die folgenden einzelnen Verarbeitungszeiten für die einzelnen Schritte des Verfahrens:

Sobel-Kantenfilter	2.6 ms
Segmentierung	0.8 ms
Differenzbild	0.4 ms
Feinanpassung mittels ASM	5.3 ms
Simulated Annealing	108 ms

Die Schritte zur Bestimmung der Fingerspitze, zur Berechnung der Zeigerichtung und des Schnittpunktes mit dem Ausgabebildschirm sind mit $<= 0.1ms$ vernachlässigbar gering. Bei einem Stereokamerasystem mit zwei Kameras ergibt sich somit eine mittlere Verarbeitungszeit des Verfahrens von 18.2 ms pro Bildpaar während der Verfolgung des Zeigegeste ohne die Notwendigkeit einer initialen Suche. Das entspricht einer Bildwiederholrate von mehr als

Abbildung 4.15: Multimodale Interaktion im häuslichen Umfeld. Steuerung eines Fernsehers durch Verwendung von Sprach- und Gestenerkennung [SMS01].

50 Bildern pro Sekunde. Für den Fall, dass eine initiale Suche mittels *Simulated Annealing* notwendig ist, weil beispielsweise die Güte der letzten Anpassung nicht den Anforderungen des statistischen Modells entspricht oder weil die Interaktion durch den Anwender unterbrochen wurde, erhöht sich die Verarbeitungszeit des Systems auf knapp 250 ms, also eine Bildwiederholrate von ungefähr vier Bilder pro Sekunde. Da in der praktischen Anwendung die Suche nach einer initialen Startpose für die Feinpassung im Schnitt drei- bis vier mal pro Sekunde notwendig ist, ergibt sich eine durchschnittliche Frequenz von 20 Bildern pro Sekunde.

4.8 Zusammenfassung

In diesem Kapitel wurde ein Verfahren zur Erkennung und Verfolgung einer Zeigegeste im Raum entwickelt, das die Anforderungen an eine intuitive und gerätefreie Interaktion mit dem Computer erfüllt. Der Anwender benötigt kein technisches Hilfsmittel, da er ausschließlich mittels einer Zeigegeste interagiert. Das Verfahren beruht auf der Erkennung und Verfolgung einer zuvor definierten und angelernten Zeigegeste mittels aktiver Formen und gliedert sich in die folgenden Einzelschritte:

1. Beschreibung der Zeigegeste und Berechnung der Deformationsmöglichkeiten mittels eines statistischen Punktverteilungsmodells. Dieser Schritt des Verfahrens bestimmt die in den Kamerabildern zu suchende Kontur der Zeigegeste. Dazu wird eine Vielzahl von Trainingskonturen manuell angegeben. Eine statistische Beschreibung des

Trainingsdatensatz ist die Grundlage für die folgenden iterativen Schritte für das Auffinden von geeigneten Startkonturen und für die Feinanpassung der Konturen.

2. Auffinden und 3D-Rekonstruktion der Spitze des Zeigefingers in den einzelnen Kamerabildern sowohl durch die geeignete Wahl des Kameraaufbaus als auch durch die Analyse von Segmentinformationen und deren 3D-Rekonstruktion.

3. Auffinden von Initialkonturen durch das heuristische Optimierungsverfahren *Simulated Annealing*. Dabei liefern sowohl das Stereokamerasystem als auch die Position der Fingerspitze und die aus dem Punktverteilungsmodell gewonnenen Deformationsmöglichkeiten der Geste genügend Information, um den Suchraum für die Optimierung so stark einzuschränken, dass dieser Schritt des Verfahrens für eine Verwendung des gesamten Verfahren in Echtzeit genutzt werden kann.

4. Anpassung der Initialkonturen an die realen Konturen in den Kamerabildern. In diesem Schritt des Verfahrens werden die gefundenen Initialkonturen mittels eines *Active Shape Model* an die in den Bildern vorhandenen Kanteninformationen der Hand angepasst und liefern damit ein Repräsentation der Zeigegeste des Anwenders auf den Bildebenen.

5. In einem letzten Schritt des Verfahrens werden die Position und die Zeigerichtung der Geste im dreidimensionalen Raum unter Verwendung des Stereokamerasystems berechnet und zur Interaktion bereitgestellt.

Zwar ist das Auffinden der Initialkontur durch *Simulated Annealing* durch die hohe Anzahl der notwendigen Iterationsschritte nicht echtzeitfähig, jedoch muss dieser Schritt des Verfahrens nur angewendet werden, wenn während der Gestenverfolgung die Feinanpassung durch das *Active Shape Model* misslingt oder wenn die Interaktion neu gestartet wird. Damit erreicht das Verfahren insgesamt die zur intuitiven Interaktion notwendigen Wiederholraten von mehr als 20 Bildern pro Sekunde. Eine große Anzahl von unterschiedlichen Ausprägungen der Zeigegeste während der Berechnung des Punktverteilungsmodells ermöglicht zur Laufzeit des Systems eine klare Abgrenzung gültiger Konturen einer Zeigegeste von in den Kamerabildern auftretenden Störungen. Die Tatsache, dass während der Modellbildung Trainingsdaten von unterschiedlichen Personen in die Berechnung des Punktverteilungsmodells eingehen und somit auch unterschiedliche Ausprägungen der Zeigegeste modelliert werden, erlaubt es dem Verfahren, auch auf neue Anwender des Systems, von denen keine Trainingsdaten existieren, korrekt zu reagieren.
Damit konnte in dieser Arbeit ein Verfahren entwickelt werden, dass mit einer vollständig automatisch ablaufenden Initialsuche nach der Handgeste eines der Hauptprobleme der intuitiven Interaktion durch videobasierte Gestenerkennung löst. Das Verfahren erfüllt zudem die in Kapitel 1 aufgestellten Forderungen an ein System zur intuitiven Gestenerkennung. Durch die Verwendung eines kalibrierten Stereokamerasystems ist kein technisches Gerät außer dem Ausgabebildschirm für den Anwender sichtbar, was den immersiven Charakter der Anwendung unterstützt. Da die Modellbildung für die Zeigegeste bereits vor dem Einsatz des Systems durchgeführt wird, existiert für einen neuen Anwender kaum Trainingsaufwand. Auch

die noch offene Frage nach der Merkmalsextraktion und Klassifikation von Handgesten in Echtzeit wird in diesem Verfahren bereits adressiert. Zum einen liefert die 3D-Rekonstruktion der Zeigerichtung und der Position der Hand im kalibrierten Stereokamerasystem eine hohe Genauigkeit für die Verwendung einer Zeigegeste als Interaktionsmedium. Zum anderen ist das Verfahren in der Lage, zu entscheiden, ob eine gültige Zeigegeste verwendet wird oder nicht. Hier zeigt sich allerdings auch die Schwäche des Verfahrens. Durch den rechen- und damit zeitintensiven Schritt der Initialsuche durch das Optimierungsverfahren *Simulated Annealing* ist das System auf die Suche nach einer einzigen statischen Handgeste beschränkt. Für eine Suche nach zwei oder nach mehreren Konturmodellen müsste dieser Schritt des Verfahrens aber wiederholt nacheinander ausgeführt werden. Das System würde dadurch nicht mehr in der Lage sein, interaktive Bildwiederholraten zu erreichen. Für eine intuitive Verwendung der Zeigegeste durch viele verschiedene Anwender stellt dies aber den Flaschenhals des Verfahrens dar. Eine Zeigegeste wird von Anwendern sehr unterschiedlich ausgeführt. Das in diesem Kapitel exemplarisch angelernte Modell einer Zeigegeste erlaubt es allerdings nur, mit ausgestrecktem Zeigefinger und dabei abgespreitztem Daumen zu deuten. Die Verwendung von beispielsweise der geöffneten Hand als Geste wird dann vom System ignoriert. Um mehrere unterschiedliche Gesten als Zeigegeste interpretieren zu können, muss also ein Verfahren entwickelt werden, das die Geste weniger stark einschränkt und so mehr Variabilität während der Interaktion zulässt. Dieses Problem wird im folgenden Kapitel 5, *Interaktion durch Punktprojektion* behandelt.

Kapitel 5

Interaktion durch Punktprojektion

Im vorigen Kapitel wurde ein Verfahren vorgestellt, dass mit einer vollständig automatisch ablaufenden Initialsuche nach der Handgeste des Anwenders eines der wichtigen ungelösten Probleme für eine intuitive Interaktion zwischen Mensch und Computer löst. Dennoch bleibt auch bei dem Verfahren der *Interaktion durch Rekonstruktion von Aktiven Formen* die Frage nach der Möglichkeit des Systems, unterschiedliche Ausprägungen einer einzelnen statischen Geste zu erkennen und zu verfolgen, noch unbeantwortet. Da das Verfahren lediglich eine einzige, vom System vorgegebene Geste erkennt und verfolgt, wird vom Anwender bereits eine minimale Lernphase gefordert, in der dem Anwender die zu verwendende Handgeste gezeigt werden muss. Natürlicher und damit intuitiver ist es aber, wenn jeder neue Anwender des Systems auf seine eigene Art des Zeigen zurückgreifen kann und sich nicht den Vorgaben des Systems unterordnen muss. Um dieses Forderung zu erfüllen, adressiert das in diesem Kapitel entwickelte Verfahren mit einer präzisen Merkmalsextraktion in Echtzeit ein weiteres Problem der intuitiven Gestenerkennung. Während die automatische Initialsuche weitestgehend aus dem Verfahren des vorigen Kapitels übernommen und präzisiert wird, wird bewusst auf die Frage nach einer Klassifikation von unterschiedlichen Gesten verzichtet. Es kann nämlich davon ausgegangen werden, dass es etliche Anwendungen gibt, die ausschließlich mittels einer Zeigegeste intuitiv gesteuert werden können und damit aus dem Ziel der Anwendung selbst ausgeschlossen werden kann, dass andere Handgesten als eine Zeigegeste vom Nutzer verwendet werden. Zudem untersucht dieses Kapitel ein weiteres Problem, das bei der Erkennung von statischen Gesten auftritt, und schafft die Möglichkeit, auch ohne eine explizite Unterscheidung verschiedener Handposen Interaktions-Ereignisse abzuleiten wie beispielsweise die Selektion von Objekten oder Bildschirmregionen.
In diesem Kapitel wird ein Verfahren zur intuitiven Interaktion mittels Zeigen auf das Ausgabegerät entwickelt, das ohne jegliche Lernphase von jedem Benutzer verwendet werden kann. Das Verfahren beruht auf der Detektion und 3D-Rekonstruktion der Fingerspitze und deren Projektion auf die Interaktionsfläche des Ausgabegerätes. Durch den bewussten Verzicht auf objekterkennende und -verfolgende Algorithmen ist das Verfahren ohne Einschränkungen von jedem neuen Nutzer des Systems sofort und ohne Trainingsphase verwendbar. Um eine einfache und intuitive Interaktion auch für technisch unversierte Nutzer zu ermöglichen, werden Methoden zur Nachverarbeitung der berechneten Ereignisse vorgestellt, die beispiels-

weise durch Glättung auch bei größeren Entfernungen des Anwenders zum Ausgabegerät ein ruhiges und präzises Zeigen ermöglichen und so eine einfache und natürliche Interaktion ermöglichen. Neben der Berechnung der Zeigerichtung in Echtzeit werden außerdem zwei Methoden vorgestellt, die Ereignisse zur Selektion erzeugen können und damit den Anwender in die Lage versetzen, virtuelle Schaltflächen zu betätigen oder in Menüstrukturen zu navigieren und gewünschte Menüeinträge auszuwählen. Damit lässt sich das Verfahren insbesondere auch für die Interaktion mit Anwendungen verwenden, die bei der Verwendung eines klassischen Eingabegerätes wie der Computermaus ein kurzzeitiges Ereignis, wie das Drücken der Maustaste erfordern.

5.1 Einleitung

Eine der wichtigsten Voraussetzungen für eine einfache und gerätelose Interaktion mit einem Computersystem ist, dass ein neuer Anwender ohne aufwendige Lernphase in der Lage ist, das System zu bedienen. Wie bereits im vorigen Kapitel gezeigt wurde, ist beispielsweise die Erkennung und Verfolgen der Zeigegeste mittels aktiver Formen in der Lage, eine Zeigegeste der menschlichen Hand von anderen sich in den Kamerabildern bewegenden Objekten und von Störungen in den Kamerabildern zu unterscheiden. Nichtsdestotrotz kann es bei einigen Anwendern des Systems dazu kommen, dass das Verfahren nicht korrekt arbeiten kann. Dies ist zum einen der Fall, wenn der Anwender sich während der Interaktion nicht an die vorgegebene Art der Zeigegeste hält. Von einer intuitiven Interaktion wird aber gerade erwartet, dass sich nicht der Anwender an das System anpassen muss, sondern das System bestmöglich in der Lage ist, auf unterschiedliche Anwender korrekt zu reagieren. Eine weitere Fehlerquelle liegt in der Möglichkeit, dass die Konturen der Hand nicht durch die im Modell angelernten Ausprägungen der Geste abgedeckt sind. Dieser Fall kann beispielsweise eintreten, wenn der Anwender Handschuhe oder große Schmuckstücke an der Hand trägt, da in diesen Fällen die Größe der Hand beziehungsweise die entsprechende Kontur der Hand nicht im statistischen Modell enthalten sind. Soll aber ein solches System beispielsweise an einem öffentlichen Ort aufgestellt und damit von einer Vielzahl von unterschiedlichen Personen genutzt werden, reicht es natürlich nicht aus, wenn es bei den *meisten* Anwendern funktioniert. Das System muss prinzipiell in der Lage sein, auf *alle* Anwender korrekt zu reagieren.

Ein weiterer Aspekt, der bei einer gestenbasierten Interaktion berücksichtigt werden muss, ist die Möglichkeit der Interaktion selbst, die das System zur Verfügung stellen kann. Zwar ist die Zeigegeste die vom Menschen am häufigsten verwendete Geste, die Einsatzmöglichkeiten für Anwendungen, die ausschließlich die Zeigerichtung des Anwenders in Echtzeit berechnen und zur Interaktion verwenden, ist jedoch stark begrenzt. Typischerweise nutzen Computerprogramme, die über eine Computermaus gesteuert werden, neben der Positionsangabe des Cursors zusätzlich auch die Möglichkeit, dass durch Drücken der Maustaste an der jeweiligen Position des Cursors Aktionen durch ein Selektionsereignis ausgelöst werden können (Mausklick-Ereignis). Erst die Verwendung dieser positionsabhängigen Selektion eröffnet die Möglichkeiten, das System auch für Anwendungen zu benutzen, die ursprünglich für eine klassische Interaktion mit der Computermaus entwickelt wurden. Durch die Verwendbarkeit

Abbildung 5.1: Selektion in einem animierten Menü [MDS05a].

von Selektionsereignissen eröffnet sich die Möglichkeit, mit den üblicherweise verwendeten Komponenten einer grafischen Benutzeroberfläche (engl.: *Graphical User Interface*, GUI) wie Schaltflächen, Menüstrukturen (siehe Abbildung 5.1) oder Dialogfenstern, so wie in der ISO-Norm für die Ergonomie der Mensch-System-Interaktion beschrieben [Sch08], zu interagieren.

Um diese Ziele zu erreichen, wird in diesem Kapitel ein neues Verfahren vorgestellt, das unter Verwendung eines kalibrierten Stereokamerasystems die Fingerspitze der Zeigegeste erkennt und die Zeigerichtung zur Interaktion auf die Ausgabefläche des Bildschirms projiziert [Mal04, MS04, MDS05a, MDS05b]. Dabei wird bewusst auf objekterkennende Algorithmen, wie beispielsweise in Kapitel 4 vorgestellt, verzichtet, um die Fehlertoleranz gegenüber neuen Anwendern des System zu maximieren. Die Auswertung der Ergebnisse der Berechnung erlaubt die Anwendung unterschiedlicher Verfahren zur Erzeugung von Selektionsereignissen analog zu einem Mausklick-Ereignis und zur Nachverarbeitung wie beispielsweise der Glättung der Cursorbewegung, die zu einer störungsfreien Interaktion führt und sich der Anwender damit nicht um die Interaktion selbst kümmern muss, sondern direkt und intuitiv mit der Anwendung interagieren kann.

5.2 Methodik der Punktprojektion

Der folgende Abschnitt dieses Kapitels befasst sich zunächst mit den Methoden, die für die eigentliche Erkennung der Handgeste verantwortlich sind, bevor im nächsten Abschnitt Erweiterungen entwickelt werden, mit denen die Möglichkeiten für potentielle Anwendungen des Verfahrens deutlich erhöht werden können. Das Verfahren der *Interaktion durch Punktprojektion* verwendet als ersten Schritt den bereits im vorigen Kapitel erwähnten Algorithmus der Detektion und 3D-Rekonstruktion der Fingerspitze, um die Position der Geste im Raum

Abbildung 5.2: Markierungen auf dem Boden leiten den Anwender intuitiv zur Interaktionsposition.

zu bestimmen. Der wesentliche Unterschied liegt aber darin, dass hier dieser Schritt nicht zur Initialisierung sondern zur direkten Bestimmung eines Gestenmerkmals verwendet wird. Die folgenden Schritte der Projektion und der Nachverarbeitung der Projektionsergebnisse ermöglichen dann die Verwendung des Verfahrens als Eingabemodalität für eine intuitive Interaktion.

5.2.1 Detektion der Fingerspitze

Der wichtigste und zentrale Punkt für die intuitive Interaktion mittels einer Zeigegeste ist in diesem Verfahren die Detektion der Fingerspitze in den einzelnen Kamerabildern, deren 3D-Rekonstruktion und Projektion auf die Ausgabefläche des darstellenden Bildschirms. Im Verfahren des vorigen Kapitels wird die Detektion und Rekonstruktion der Fingerspitze im Raum ausschließlich als ein erster Schritt für eine Einschränkung des Suchraums für die folgende Bestimmung einer Startkontur mit *Simulated Annealing* verwendet. Insbesondere die 3D-Rekonstruktion der korrespondierenden Bildpunkte einer potentiellen Fingerspitze wird ausschließlich dazu verwendet, grobe Fehler der bildverarbeitenden Schritte zur detektieren und auszugleichen, indem lediglich überprüft wird, ob der rekonstruierte 3D-Punkt innerhalb des vorab definierten Interaktionsvolumens des Anwenders liegt. Die eigentlichen Merkmale der Geste werden dann in einem späteren Schritt des Verfahrens durch Aktiven Formen bestimmt. In dem hier vorgestellten Verfahren ist nun aber die rekonstruierte Fingerspitze des Anwenders bereits Teil des Ergebnisses der Merkmalsbestimmung selbst. Daher ist sowohl eine robuste Bestimmung der Fingerspitze in den Kamerabildern als auch eine präzise Rekonstruktion im 3D-Raum elementare Voraussetzung für eine hohe Erkennungsrate des Systems und damit auch für die intuitive Verwendbarkeit des Verfahrens durch den Anwender.
Für diesen Schritt des Verfahrens spielt die geeignete Wahl des Kameraaufbaus eine entscheidende Rolle, um sicherzustellen, dass nach den bildverarbeitenden Schritten die 3D-Rekonstruktion der identifizierten Pixelkoordinaten mit möglichst hoher Wahrscheinlichkeit

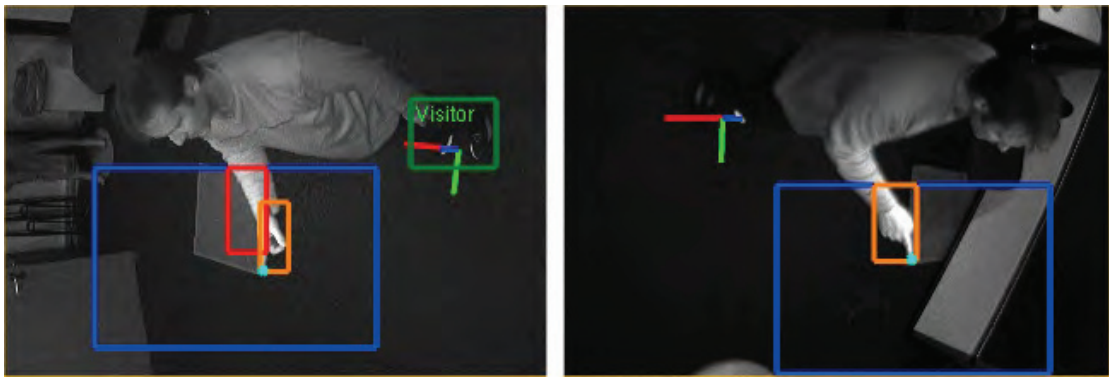

Abbildung 5.3: Kamerabilder mit überlagerten Ergebnissen der bildverarbeitenden Schritte. Neben den Interaktionsbereichen sind unter anderem auch die Ergebnisse der Segmentierung und die erkannte Fingerspitze dargestellt.

auch die tatsächliche Fingerspitze im Raum gefunden wird. Die Positionierung der Kameras ergibt sich dabei aus der Art der Anwendung selbst. Bei der Verwendung der Zeigegeste als Eingabemodalität für eine dargestellte Anwendung kann ausgeschlossen werden, dass sich der Anwender frei im Raum bewegt. Seine Position ist durch das Ausgabegerät, mit dem er interagieren möchte, auf einen fest vorgegebenen Bereich direkt vor dem Bildschirm beschränkt. Insbesondere bei sehr großen Anzeigesystemen wie beispielsweise Projektionsleinwänden oder großen LCD-Anzeigesystemen muss dem Anwender eine optimale Position vor dem Bildschirm vorgegeben werden, von der aus er den optimalen Anstand zum Ausgabegerät einhält und damit die gesamte Anwendung gut überblicken kann. Diese Position kann einfach durch beispielsweise Markierungen in Form von Fußabdrücken auf dem Boden vor dem Bildschirm realisiert werden, auf die der Anwender intuitiv tritt (siehe Abbildung 5.2). Auch die Orientierung des Anwenders im Raum ist bereits durch das System selbst vorgegeben. Der Anwender steht vor dem Ausgabegerät, um mit einer Anwendung zu interagieren, ist also auf den Bildschirm fokussiert. Damit ist auch die Richtung der Handgeste bereits vorab grob feststellbar.

Das hier vorgestellte Verfahren macht sich diese Einschränkungen zu Nutze, um die Fingerspitze robust und in Echtzeit zu detektieren und zu verfolgen. Durch geeignete Positionierung und Orientierung des Kameras des Stereosystems können ausreichend Informationen gewonnen werden, die bei einer Analyse der Ergebnisse der bildverarbeitenden Schritte wie Kantenextraktion, Differenzbilder und Segmentierung verwendet werden können, um die Fingerspitze der Zeigegeste zu identifizieren. Um sicherzustellen, dass die Fingerspitze auf Bildbasis lokalisiert und vom Körper des Anwenders separiert werden kann, werden die beiden Kameras rechst und links vom Anwender leicht vor dem Anwender selbst positioniert. Eine ausreichende Höhe der Kameras stellt dabei sicher, dass der Anwender die Kameras nicht im direkten Sichtfeld hat und dadurch in der Interaktion gestört und von der Anwendung abgelenkt wird. Die Orientierung der Kameras wird so gewählt, dass sichergestellt werden kann, dass die Fingerspitze während des Zeigens auf den Bildschirm einer vorgegebenen Richtung

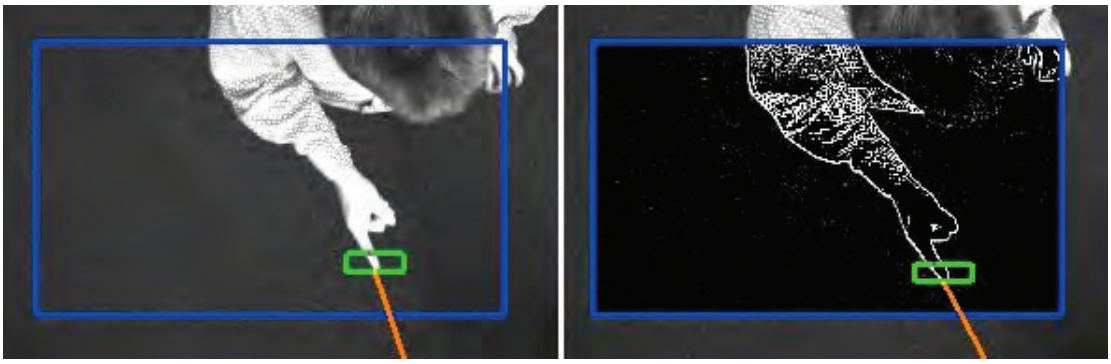

Abbildung 5.4: Bildverarbeitung zur Detektion der Fingerspitze. Den Bildern (Originalbild links und Kantenextraktion rechts) ist die Region der Fingerspitze und eine Annäherung der Zeigerichtung überlagert.

in den Kamerabildern folgt. Wie in Abbildung 5.3 zu sehen ist, können die Kameras beispielsweise so gedreht werden, dass nach einer Segmentierung von sich bewegenden Objekten die Fingerspitze immer als der tiefster Punkt im Bildkoordinatensystem angenommen werden kann. Um eventuell auftretende Störungen zu eliminieren, werden mehrere mögliche Punktkorrespondenzen auf die Wahrscheinlichkeit untersucht, dass deren 3D-Rekonstruktion im Interaktionsvolumen des Anwenders liegt.

Die Liste der möglichen Punktkorrespondenzen, die zur 3D-Rekonstruktion verwendet werden, ermittelt sich aus einigen bildverarbeitenden Schritten: Bei Start des Systems werden zunächst Referenzbilder des Interaktionsvolumens ohne einen Anwender aufgenommen. Von diesen Referenzbildern werden außerdem die Kanteninformationen mittels einer Faltungsoperation berechnet und ebenfalls gespeichert. Zur Laufzeit des Systems werden von neu eingehenden Luminanzbildern der Kameras ebenfalls die Kantenbilder erzeugt. Durch die Berechnung der Differenzbilder von Referenzkantenbildern und den Kantenbildern der aktuellen Kamerabilder können Änderungen der Szene in Bezug auf das leeren Interaktionsvolumen robust auf Bildbasis ermittelt werden. Die Differenzbilder dienen nun als Grundlage für einen binarisierenden Segmentierungsschritt an einem vorgegebenen Schwellwert. Das Ergebnis dieser Segmentierung ist für jede Kamera eine Liste von Segmenten, deren Endpunkte in Richtung der vorab definierten Suchrichtung als potentielle Kandidaten für die Rekonstruktion der Fingerspitze dienen. Dabei gilt zunächst die einfache Annahme, dass die Endpunkte der beiden Segmente in Zielrichtung (siehe Abbildung 5.5) die Fingerspitze auf der Bildebene beschreiben. Durch die Wahl des Kameraaufbaus setzt das Verfahren lediglich voraus, dass sich zwischen Anwender und dem darstellenden Ausgabebildschirm keine neuen oder beweglichen Objekte während der Interaktion befinden. Nichtsdestotrotz kann es vorkommen, dass sich durch entweder ändernde Lichtverhältnisse oder beispielsweise andere Personen, die sich im Blickfeld der Kameras aufhalten, Segmente in den Kamerabildern entstehen können, die diese Annahme verletzen. Um diese Fehlerquelle zu eliminieren, werden die einzelnen Endpunkte der Segmentlisten einer Analyse unterzogen, welche die

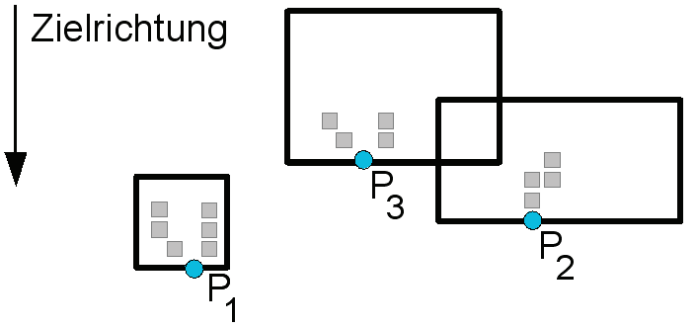

Abbildung 5.5: Segmentreihenfolge und Segmentendpunkte zur Bestimmung der Fingerspitze ergeben sich aus der vorgegebenen Zielrichtung.

rekonstruierten 3D-Kandidaten mit dem Interaktionsvolumen des Systems abgleicht. Dies geschieht unter der Annahme, dass zwei Segmentendpunkte, die nicht zum selben realen Objekt gehören, zu einem 3D-Punkt rekonstruiert werden, der außerhalb des Interaktionsvolumens liegt. Im Einzelnen geht der Algorithmus zur Bestimmung der Fingerspitze und zur Eliminierung von inkorrekten Segmenten folgendermaßen vor:

Algorithmus 3 : Bestimmung der Fingerspitze durch Segmentanalyse

Vorraussetzungen: Vektoren der Segmente $S_{i=1..n}$ und $S_{j=1..m}$; Position und Größe des realen Interaktionsvolumens

Ergebnis: 3D-Rekonstruktion der Fingerspitze

1: Sortiere die Segmentvektoren nach ihrer Zielrichtung, so dass das Segment mit dem größten maximalen y-Wert an erster Stelle im Vektor liegt.
2: **Für alle** i
3: **Für alle** j
4: Ermittle die Endpunkte P_i und P_j der Segmente i und j in Zielrichtung.
5: Berechne mittels des Stereokamerasystems aus den Segmentendpunkten P_i und P_j den korrespondierenden 3D-Punkt P_x.
6: **Wenn** (P_x innerhalb des vorgegebenen Interaktionsvolumens liegt) **dann**
7: **Gebe zurück** P_x als gültige Fingerspitze.
8: **Ende wenn**
9: **Nächstes** j
10: **Nächstes** i
11: **Gebe zurück** einen als ungültig markierten Punkt P_x.

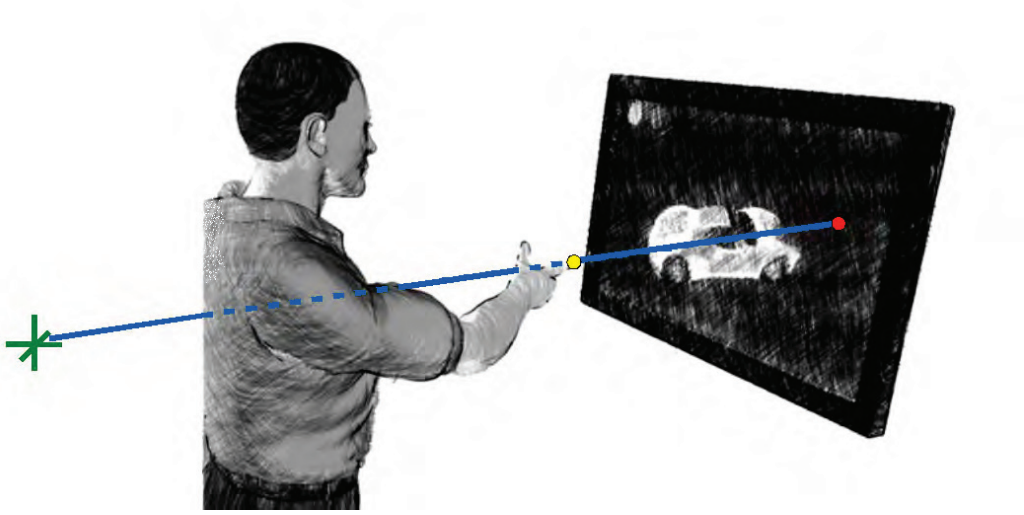

Abbildung 5.6: Projektion der Fingerspitze auf das Ausgabegerät durch Definition eines Referenzpunktes hinter dem Anwender.

Die durch den Kameraaufbau definierte Zielrichtung bestimmt für den Algorithmus sowohl die Reihenfolge in der sortierten Liste der Segmente als auch die Segmentendpunkte innerhalb der einzelnen Segmente. Der Segmentendpunkt wird als letzter Punkt des Segments in Zielrichtung definiert. Für den Fall, dass nicht ein einzelner, sondern mehrere Endpunkte eines Segments existieren, wird aus diesen Punkten das arithmetische Mittel als Segmentendpunkt gewählt. Zur einfacheren Berechenbarkeit wird die Zielrichtung entlang einer der Bildkoordinatenachsen gewählt. Wird kein gültiger Punkt als Repräsentant der Fingerspitze gefunden, weil alle möglichen Punktkorrespondenzen einen 3D-Punkt außerhalb der Interaktionsvolumens erzeugen, wird ein im System definierter, ungültiger Punkt zurückgegeben. Ungültige Punkte werden in dem in Abschnitt 5.2.3, Nachverarbeitungsschritte ignoriert und stellen damit sicher, dass kurzzeitig auftretende Störungen der Kamerabilder nicht zu einem unerwünschten Springen des Cursors führen.

5.2.2 Projektion der Fingerspitze

Als nächster Schritt des Verfahrens wird die zuvor ermittelte Fingerspitze der Zeigegeste auf den Ausgabebildschirm projiziert, um eine Interaktion zu ermöglichen. Dazu wird nach der Kamerakalibrierung und der Definition des Weltkoordinatensystems ein Referenzpunkt hinter dem Anwender festgelegt, der zusammen mit der Fingerspitze einen Strahl im Raum beschreibt. Der Schnittpunkt dieses Strahls mit der Fläche des Ausgabegerätes erzeugt dann den Punkt der zur Interaktion verwendet werden kann (siehe Abbildung 5.6). Es ist offensichtlich, dass die gewählte Position des Referenzpunktes einen erheblichen Einfluss auf den projizierten Punkt und damit auch auf das Interaktionsverhalten des Anwenders hat. Projektionsfehler und Ungenauigkeiten, die eine Interaktion negativ beeinflussen, können

beispielsweise dadurch entstehen, dass der Anwender von der vorgegebenen Position, die durch die Fußmarkierungen vorgegeben ist, abweicht. Zwar können solche Einflüsse dadurch verringert werden, dass der Referenzpunkt möglichst weit hinter dem Anwender definiert wird, allerdings sind dann auch weitaus größere Bewegungen der Hand notwendig, um den projizierten Interaktionspunkt zu bewegen. Dies wiederum erschwert es dem Anwender aber beispielsweise, die Ränder des Bildschirms mit der Zeigegeste zu erreichen. Eine geeignete Position für den Referenzpunkt kann heuristisch durch einfaches Befragen von vielen verschiedenen Anwendern des Systems ermittelt werden. Als geeignete Position im Raum für diesen Referenzpunkt hat sich ein Punkt in etwa 1,30 Meter Höhe ungefähr einen halben Meter hinter dem Anwender für die meisten Systemaufbauten als günstig erwiesen. Durch die Verwendung von Markierungen auf dem Boden als Indikator für die designierte Position des Anwenders für die Interaktion kann so eine ausreichende Genauigkeit für unterschiedliche Anwender des Systems erreicht werden. Dabei spielen Unterschiede in der Körpergröße keine entscheidende Rolle, ob eine intuitive Interaktion möglich ist. Auch die Frage, ob der Anwender Rechts- oder Linkshänder ist, spielt bei diesem Verfahren keine wesentliche Rolle. Bedingt durch die Tatsache, dass die menschliche Zeigegeste prinzipiell sehr ungenau ist und von verschiedenen Anwendern auch unterschiedlich verwendet wird, führt dazu, dass die meisten Anwender den Interaktionspunkt auf dem Bildschirm nicht dort erwarten, wo der Schnittpunkt des projizierten Strahls tatsächlich berechnet wurde. Daher ist es wichtig, dass die Anwendung selbst permanent eine visuelle Rückantwort liefert. Dies kann einfach erreicht werden, indem am Interaktionspunkt ein Zeiger (Cursor) dargestellt wird. Die Erfahrung mit dem System hat gezeigt, dass die meisten Menschen intuitiv in der Lage sind, sich auf die scheinbare Ungenauigkeit der Zeigegeste einzustellen und bereits nach wenigen Sekunden ein sicheres Gespür dafür bekommen, die gestenbasierte Interaktion richtig zu verwenden. Die Grundlage der Projektion selbst und die Berechnung des Schnittpunktes des Strahls mit der Ausgabefläche wurde bereits in Abschnitt 4.6 beschrieben. Das Ergebnis der Berechnung liefert den Schnittpunkt als 3D-Punkt im Weltkoordinatensystem. Um den Punkt für eine Interaktion verwenden zu können, ist nun noch eine geeignete Skalierung in der Projektionsfläche notwendig. Nach der extrinsischen Kamerakalibrierung und der Definition des Weltkoordinatensystems ist es demnach notwendig, die Position und Orientierung der Ausgabefläche anzugeben. Dazu müssen drei der Eckpunkte des Bildschirms manuell ausgemessen und dem System zugänglich gemacht werden. Da der Interaktionspunkt in den Einheiten des Weltkoordinatensystems vorliegt, wird mit Hilfe der Breite und Höhe des Bildschirms eine geeignete Skalierung in den Intervallen $x \in [0..1]$ und $y \in [0..1]$ mit $x, y \in R$ für das Gestenerkennungssystem gewählt. Da die Gestenerkennung strikt von der Anwendung, mit welcher der Anwender interagiert, getrennt ist und damit auch die Auflösung des Ausgabegerätes in Pixeln nicht bekannt ist, muss eine geeignete Rückskalierung auf Pixelkoordinaten in der Anwendung selbst erfolgen.

5.2.3 Nachverarbeitungsschritte

Die mögliche große Distanz des Anwenders zum Ausgabebildschirm benötigt eine weitere Unterstützung des Anwenders während der Interaktion. Es ist leicht vorstellbar, dass kleinste

Bewegungen der Fingerspitze im Raum durch die Projektion auf eine große Ausgabefläche zu einer visuellen Beeinträchtigung führen und Sprünge des dargestellten Cursors auf dem Bildschirm bewirken können. Diese kleinen Positionsänderungen im Raum können dabei entweder durch tatsächliche Bewegungen der Hand, aber auch durch Rekonstruktionsungenauigkeiten durch die Verwendung von diskreten Bildpunkten auftreten. Da die Kamerabilder in ihrer Auflösung begrenzt sind, kann bereits die Verschiebung der Fingerspitze in einem der Kamerabilder um ein einziges Pixel eine Translation des Cursors um mehrere Zentimeter auf dem Ausgabebildschirm verursachen. Daher ist es notwendig, geeignete Methoden zur künstlichen Stabilisierung der Handbewegungen bereitzustellen.

Glättung durch Mittelwertbildung

Die einfachste Art der Stabilisierung der Zeigerichtung ist eine Glättung des Interaktionspunktes durch Bildung des arithmetischen Mittelwerts über ein vorgegebenes Zeitintervall. Dazu wird der berechnete Schnittpunkt mit der Ausgabefläche nicht einfach als Interaktionspunkt angenommen, sondern zunächst in einem Vektor von 2D-Punkten der vorgegebenen Länge n gespeichert. Der nach außen gegebene Interaktionspunkt $p = (x_c, y_c)$ berechnet sich nun als arithmetische Mittelwerte der x- und y-Koordinaten

$$x_c = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad y_c = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{5.1}$$

Als ungültig markierte Punkte (vergleiche Abschnitt 5.2.1) werden dabei ignoriert und gehen nicht mit in die Berechnung ein. Die Länge des Vektors bestimmt damit zum einen die Intensität der Glättung, bewirkt auf der anderen Seite aber auch mit zunehmender Größe eine Verzögerung des Systems, was von vielen Anwenders als "Nachlaufen des Cursors" und damit als unerwünschten Effekt wahrgenommen wird. Für die Anzahl der zur Glättung verwendeten Punkte und damit für die Länge des Vektors muss also ein Kompromiss zwischen auftretenden Ungenauigkeiten und Sprüngen auf der einen Seite und der Reduzierung der Verzögerung auf der anderen Seite gefunden werden. Die Wahrnehmung dieser beiden unerwünschten Effekte hängt stark von der Geschwindigkeit der gerade verwendeten Handbewegung des Anwenders ab. Lässt er den Zeiger schneller über den Bildschirm gleiten, muss die Verzögerung minimiert werden, während die Reduktion der Positionsungenauigkeiten reduziert werden muss, wenn der Anwender für eine längere Zeit auf eine bestimmte Stelle zeigt und sich auf einen Punkt der Ausgabefläche konzentriert. Daher wird die Anzahl der zur Glättung verwendeten Punkte in Abhängigkeit zu der Varianz der Positionsänderung der Interaktionspunkte im Vektor gesetzt. Je schneller der Anwender die Hand bewegt, desto weniger Punkte gehen in die Mittelwertberechnung ein und reduzieren so die Verzögerung. Je langsamer die Bewegung des Interaktionspunktes wird, desto mehr Punkte gehen in die Berechnung ein und stabilisieren damit den Cursor stärker. Die maximale Anzahl der verwendbaren Punkte ergibt sich dabei aus der vorab definierten Länge des Vektors, der die Schnittpunkte mit der Ausgabefläche speichert.

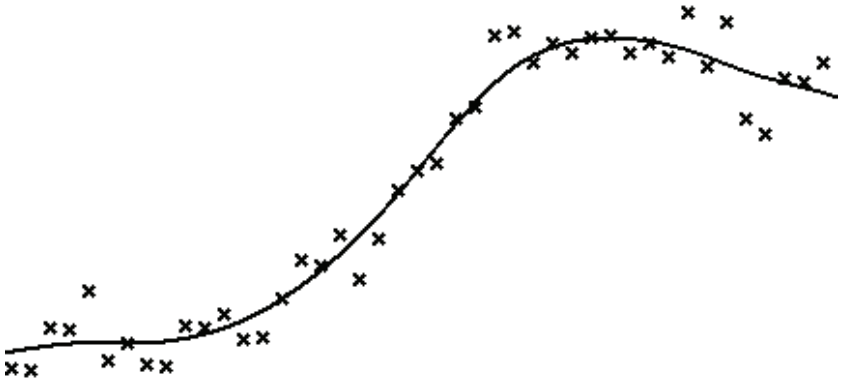

Abbildung 5.7: Beispiel einer Glättung mittels eines *Smoothing Splines* über 43 Werten.

Glättung durch Smoothing Splines

Eine weitere Möglichkeit der Glättung der Interaktionspunkte ist die Verwendung von glättenden Splines (engl.: *smoothing splines*) [Wah90, SEM00]. Eine Verwendung von Spline-Funktionen hat den Vorteil, dass die Berechnung durch stückweise Interpolation geschieht und dadurch eine Verzögerung des Systems vermieden werden kann. Allerdings hat die Verwendung von Splines den offensichtlichen Nachteil, dass durch die wenigen verwendeten Punkte während der Berechnung das Verfahren anfällig gegenüber Ausreißern ist. Smoothing Splines verwenden daher eine Mischung aus Spline-Interpolation und linearer Regression (siehe Abbildung 5.7). Dem Spline-Ansatz wird ein Strafterm $\lambda \geq 0$ hinzugefügt, der die gewünschte Glättung der Daten bewirken soll. Dieser Wert bestraft eine zu starke Gewichtung von Ausreißern mit zu großem Krümmungsverhalten. Während das Verfahren bei $\lambda \to 0$ sich der reinen Spline-Interpolation nähert, geht das Verfahren bei $\lambda \to \infty$ in die lineare Regression über. Bei n Wertepaaren (x_i, y_i) mit $i = 1..n$ berechnet sich das Smoothing Spline μ als

$$\sum_{i=1}^{n}(y_i - \mu(x_i))^2 + \lambda \int \mu''(x)^2 dx \tag{5.2}$$

Für die Verwendung zur Glättung der Interaktionspunkte kann der Strafterm λ erneut von der Varianz der im Vektor gegebenen Punkte abhängig gemacht werden. Bei nur kleinen Positionsänderungen über längere Zeit der Strafterm λ erhöht, um die Interaktion stärker zu glätten und damit zu stabilisieren, während bei schnelleren Zeigebewegungen und damit einhergehender stärkerer Varianz der Daten λ gesenkt wird.

5.3 Erweiterungen des Verfahrens

Im folgenden werden nun Erweiterungen beschrieben, die die Verwendbarkeit des Verfahrens für intuitiv bedienbare Anwendungen erhöht. Neben der Möglichkeit, ausschließlich mittels einer einzigen statischen Handpose Ereignisse auszulösen, um beispielsweise virtuelle Objekte zu selektieren, wird auch ein videobasierter Besuchersensor vorgestellt, der Anwendungen ermöglicht, automatisch auf einen neuen Nutzer des Systems zu reagieren.

5.3.1 Selektionsereignisse

Wie bereits in der Einleitung dieses Kapitel erläutert, sind die Einsatzmöglichkeiten einer Zeigegestenerkennung ohne die Möglichkeit zur Erzeugung von Selektionsereignissen stark eingeschränkt. Die Anzahl der möglichen Anwendungen, die ausschließlich durch die Translation eines zweidimensionalen Zeigers auf der Ausgabefläche zu bedienen sind, bleibt weit hinter den vorstellbaren Anwendungen, die eine positionsgenaue Auswahl zulassen, deutlich zurück. Für das hier vorgestellte Verfahren der Detektion und Projektion der Fingerspitze stellt sich insbesondere die Aufgabe, solche Selektionsereignisse ohne weitere Informationen über die Hand und ihre derzeitige Pose zu erzeugen. Daher werden im Folgenden zwei Methoden vorgestellt, welche die bereits für die Nachverarbeitungsschnitte der Interaktionsglättung angelegten Vektoren der Ergebnispunkte verwenden. Beide Verfahren machen sich dabei zu Nutze, dass für eine positionsgenaue Selektion eines Objektes der Zeiger auf einem beliebigen Weg an die gewünschte Position gebracht werden muss. Da der Weg zum Objekt jedoch nicht von Interesse ist, bewegt man den Cursor möglichst schnell dorthin. Des Weiteren kann angenommen werden, dass der Zeiger für eine Weile an der gewünschten Position verweilt. Dies ist auch bei einem klassischen Eingabegerät wie der Computermaus der Fall. Zunächst wird der Cursor in kurzer Zeit an die gewünschte Stelle gebracht und verweilt dort einen Moment, bis die Maustaste gedrückt wird. Da der Anwender nun eine Reaktion des Systems erwartet, wird in der ersten Zeit nach dem Selektionsereignis die Maus nicht weiter bewegt. Umgekehrt kann daraus gefolgt werden, dass es einem gewohnten Empfinden entspricht, dass bei schnellen Translationsänderungen des Zeigers keine Selektion stattfinden soll, das Verweilen des Zeigers an einem Ort wohl aber mit dem Ziel einer Systemreaktion verknüpft werden kann.

Selektion durch Regionenanalyse

Die erste in diesem Verfahren verwendete Methode zum Auslösen von Selektionsereignissen analysiert zuvor für eine Anwendung definierte Selektionsregionen. In vielen Anwendungen sind Position und Dimension der selektierbaren Elemente wie beispielsweise Schaltflächen oder Menüstrukturen beim Start der Anwendung bereits bekannt und auch im Verlauf der Anwendung unveränderlich. Außerdem unterliegen eventuelle Statuswechsel der Elemente fest vorgegebenen Regeln. Die Anwendung entscheidet selbst, ob zu einem bestimmten Zeitpunkt an der entsprechenden Stelle eines Elements ein eingehendes Ereignis akzeptiert

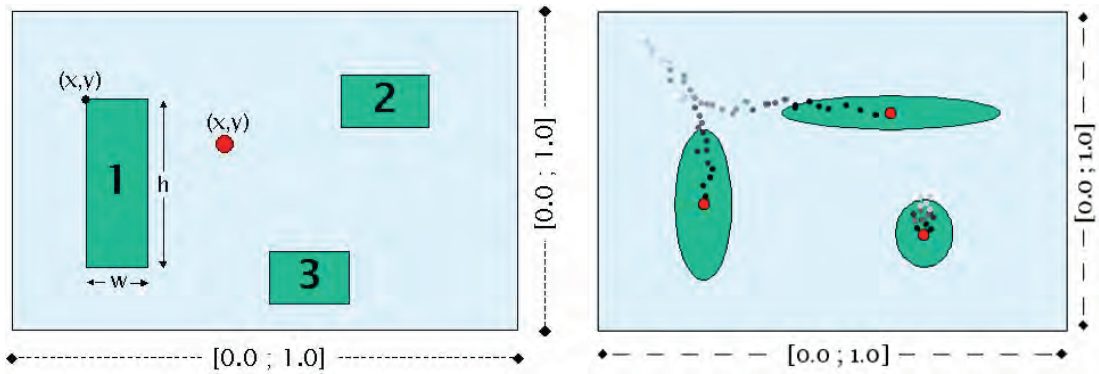

Abbildung 5.8: Vordefinierte selektierbare Regionen (links) und Geschwindigkeitsanalyse zur Selektion. Die Varianzen der Positionen der Interaktionspunkte bestimmen die Radien der Ellipsen (rechts). Unterschreiten die Radien eien vorgegebenen Schwellewrt, wird ein Selektionsereignis ausgelöst.

wird oder nicht. Dies macht sich die hier verwendete Methode zu Nutze und erwartet deshalb beim Start des Systems die notwendigen Parameter für alle in der Anwendung verwendeten Elemente. Da Elemente einer grafischen Benutzerschnittstelle fast ausschließlich rechteckige Strukturen besitzen, kann im Zeigegestenerkennungssystem ein Vektor von entsprechenden Zielregionen definiert werden, der zur Laufzeit des Systems bei jedem neuen Bildpaar zusätzlich zur Zeigerichtung und dem daraus resultierenden Interaktionspunkt analysiert wird (siehe Abbildung 5.8, links). Für jede Region gilt dabei, dass ein Selektionsereignis ausgelöst wird, wenn die Anzahl der Interaktionspunkte der letzten n Bildpaare, die innerhalb der Region liegen, einen vorgegebenen Schwellwert erreicht. Analog wird ein Ereignis zur Deselektion erzeugt, wenn der Schwellwert unterschritten wird. Dieses Verhalten einer Region entspricht somit dem Drücken und Loslassen einer Taste bei einer Computermaus (*Press-* und *Release-Events*). Die Tatsache, dass immer eine ausreichende Anzahl von Interaktionspunkten zur Analyse verwendet wird, um ein Ereignis auszulösen, stellt dabei sicher, dass der Anwender tatsächlich für eine gewisse Zeit mit dem Zeiger in der Zielregion verweilen muss, um ein Ereignis zu generieren. Je höher der Schwellwert dabei gewählt wird, desto robuster ist das Verfahren gegenüber versehentlich ausgelösten Ereignissen, erhöht aber auf der anderen Seite auch die Zeit, die der Anwender auf ein Objekt zeigen muss, um ein Objekt zu selektieren, beziehungsweise warten muss, bis die Deselektion tatsächlich erfolgt, nachdem er nicht mehr auf das Objekt zeigt.

Selektion durch Geschwindigkeitsanalyse

Nicht für alle Anwendungen gilt, dass die Positionen und Dimensionen der Elemente der grafischen Benutzerschnittstelle bereits beim Start der Anwendung bekannt sind und im Verlauf der Interaktion unverändert bleiben wie im vorigen Abschnitt angenommen. Ein gutes Beispiel für eine Klasse solcher Anwendungen sind echtzeitfähige virtuelle Welten. Interaktive

Abbildung 5.9: Selektion von Objekten in einer virtuellen 3D-Welt. Teilgeometrien des rotierbaren Flugzeugmodells sind mit Sensoren ausgestattet, über die die Transparenz der Außenhaut gesteuert wird.

Elemente werden beispielsweise in der *Virtual Reality Modeling Language* (VRML) durch eine Gruppe von Sensorknoten bereitgestellt. Diese Sensoren werden mit Teilgeometrien des Szenengraphen verknüpft und sind damit zu jeder Zeit der Anwendung dort verfügbar, wo sich die Geometrie auf dem Bildschirm befindet (siehe Abbildung 5.9). Die Veränderungen der Lage und Größe der selektierbaren Objekte entstehen durch die Möglichkeit, mit der virtuellen Welt zu interagieren und beispielsweise im 3D-Raum zu navigieren oder Objekte der Szene zu transformieren. Um also das Verfahren der Punktprojektion für die Interaktion mit virtuellen Welten verwendbar zu machen, wird eine Selektionsmethode benötigt, die in der Lage ist, ein Auswahlereignis an jeder beliebigen Position des Bildschirms zu erzeugen.

Die zweite in diesem Verfahren verwendete Methode zur Erzeugung von Selektionsereignissen beruht auf der Analyse der Geschwindigkeit des Interaktionszeigers. Wie bereits beschrieben kann davon ausgegangen werden, dass ein Anwender bei schnelleren Bewegungen der Zeigegeste kein Ereignis auslösen möchte, wohingegen das Zeigen auf eine bestimmte Position des Bildschirms für eine entsprechende Zeit eine Auswahl an dieser Stelle bewirken soll. Damit bleibt für das Verfahren die Aufgabe, die Geschwindigkeit der Interaktionspunkte auf der Ausgabefläche zu untersuchen. Fällt die Geschwindigkeit der Positionsänderungen unter einen vorgegebenen Schwellwert und erhöht sich innerhalb einer vorgegebenen Zeit nicht erneut, wird eine Selektionsereignis erzeugt. Auf der anderen Seite kann nach einem Selektionsereignis durch eine plötzliche Beschleunigung der Zeigegeste ein Deselektionsereignis ausgelöst werden. Für die Analyse der Geschwindigkeit werden daher die Varianzen der Positionen der Interaktionspunkte für eine gegebene Vektorlänge berechnet. Die Varianzen werden dabei getrennt nach den beiden Hauptachsen des normierten Ausgabegerätes bestimmt. Geometrisch können die Varianzen der Geschwindigkeit der Interaktionspunk-

te als Radien einer Ellipse aufgefasst werden (siehe Abbildung 5.8, rechts), deren Form sich während der Interaktion ständig ändert. Unterschreiten beide Radien den vorgegebenen Schwellwert, wird an der derzeitigen geglätteten (vergleiche Abschnitt 5.2.3) Position ein Selektionsereignis ausgelöst.

5.3.2 Besuchererkennung

Eine gerätefreie Interaktion, die ausschließlich durch die Verwendung einer Handgeste gesteuert wird, ist für viele Menschen etwas Neues und Ungewohntes. Wenn aber ein solches System an beispielsweise einem öffentlichen Ort ausgestellt ist, an dem auch viele technisch unversierte Anwender das System benutzen sollen, ist es notwendig, die Hemmschwelle zur Benutzung der neuen Technologie zu senken. Ein dafür entscheidender Anstoß kann es sein, wenn das System proaktiv auf den Anwender reagieren kann. Das setzt aber voraus, dass das System in der Lage ist, zu erkennen, ob sich ein potentieller neuer Nutzer in der Nähe des Systems befindet oder sogar bereits erste Versuche einer Interaktion unternimmt. In dem hier vorgestellten Verfahren wird ein videobasierter Besuchersensor verwendet, um zu entscheiden, ob sich ein neuer Anwender an der Position zur Interaktion mit dem System befindet. Entsprechend ist dieser Sensor auch in der Lage, zu erkennen, wenn ein Anwender die Interaktion nicht nur kurzzeitig unterbricht, sondern die Interaktionsposition endgültig verlässt. Dafür wird in einem der Kamerabilder eine Region definiert, die permanent die Unterschiede des Laufzeitbildes zu dem beim Start des Systems aufgenommenen Referenzbild analysiert. Die Position der Region im Bild wird so gewählt, dass der Anwender unabhängig von seiner Zeigegeste erkannt werden kann. Beispielsweise kann diese "Besucherregion" die am Boden angebrachten Markierungen und deren Umgebung beobachten (siehe auch Abbildung 5.3). Die Regionenanalyse beruht dabei wie bereits im Abschnitt 5.2.1 beschrieben aus der Auswertung des segmentierten Differenzbildes zwischen Referenzkantenbild und dem Kantenbild des aktuellen Kamerabildes. Überschreitet die Anzahl der Pixel in dem segmentierten Bereich einen vorgegebenen Schwellwert, hat sich ein neuer Anwender vor den Ausgabebildschirm begeben und es kann ein entsprechendes Ereignis generiert und an die Anwendung übermittelt werden, um eine entsprechende Reaktion zu bewirken. Um Fehlentscheidungen des Systems zu vermeiden, darf weder die Anwendung automatisch zurückgesetzt oder beendet werden, wenn sich der Anwender nur kurzzeitig von der Interaktionsposition entfernt, noch darf das System einen neuen Anwender erkennen, wenn eine Person beispielsweise über die Markierungen läuft, ohne vor dem Bildschirm stehen zu bleiben. Daher muss der Schwellwert zum Auslösen eines Ereignisses entsprechend "träge" eingestellt werden, um solche unerwünschten Ereignisse zu verhindern.

5.4 Zusammenfassung

In diesem Kapitel wurde ein Verfahren vorgestellt, dass eine intuitive Interaktion mittels einer Zeigegeste realisiert. Das Verfahren beruht auf der Detektion der Spitze des Zeigefingers auf Bildbasis, deren 3D-Rekonstruktion und Projektion mittels eines Referenzpunktes

auf die Ausgabefläche des Bildschirms. Durch eine geeignete Wahl des Kameraaufbaus und insbesondere auch der Orientierung der Kameras kann mittels einer Analyse von Differenzbildsegmenten und deren Segmentendpunkten in einer vorgegebenen Zielrichtung die Fingerspitze des Anwenders robust detektiert und rekonstruiert werden. Durch die Reduktion der Information über die Zeigegeste auf die Spitze des Zeigefingers benötigt das Verfahren keine modellbildende Trainingsphase und ermöglicht es jedem Anwender, sofort und intuitiv zu interagieren. Prinzipiell ist die hier vorgestellte Methodik zu Bestimmung der dreidimensionalen Position einer Zeigegeste auch auf andere statische Handgesten anwendbar. Betrachtet man aber die Handsilhouetten anderer Gesten in den Kamerabildern, zeigt sich, dass die Segmente anderer Handposen meist nicht so eindeutig von der Gesamtsilhouette trennbare Merkmale wie die Spitze des Zeigefingers aufweisen. Dadurch können leicht plötzlich auftretenden Sprünge des rekonstruierten Merkmals im Raum entstehen und stören dadurch den intuitiven Charakter der Interaktion. Zum anderen ist die Zahl der vorstellbaren Anwendungen, die mit anderen, einzelnen Handpose bedienbar sind, stark eingeschränkt.
Um störende Positionsänderungen der projizierten Punkte während der Interaktion zu vermeiden, verwendet das Verfahren zwei Methoden zur Glättung der Zeigerichtung durch Mittelwertbildung und durch glättende Spline-Funktionen. Darüber hinaus beinhaltet das Verfahren zwei unterschiedliche Methoden zur Erzeugung von Selektionsereignissen mittels der Analyse von vordefinierten Ereignisregionen und der Analyse der Positionsgeschwindigkeit der Interaktionspunkte. Damit ist der Anwender in der Lage, Elemente einer grafischen Benutzerschnittstelle ausschließlich durch die Verwendung der Zeigegestik zu bedienen. Eine videobasierte Methode zur Besuchererkennung ermöglicht es dem System, proaktiv auf einen neuen Anwender zu reagieren oder entsprechende Ereignisse zu erzeugen, wenn der Anwender die Interaktion beendet. Durch den bewussten Verzicht auf objekterkennende und damit rechenintensive Algorithmen beschränkt sich die relevante Bearbeitungszeit des Verfahrens lediglich auf die Durchführung der bildverarbeitenden Schritte der Extraktion von Kanten, Berechnung der Differenzbilder, Segmentierung und die Analyse der Bildsegmente. Dadurch kann die Echtzeitfähigkeit des Verfahrens mit bis zu 30 Bildern pro Sekunde gewährleistet werden und erfüllt die Voraussetzungen für eine einfache und intuitive Interaktion zwischen Mensch und Computer.

Kapitel 6

Interaktion durch Merkmalbasierte Gesten-Klassifikation

In den beiden vorigen Kapiteln wurden zwei Verfahren vorgestellt, die in der Lage sind, eine Zeigegeste in Echtzeit zu erkennen und zu verfolgen und so eine intuitive Interaktion zwischen Mensch und Computer zu ermöglichen. Beide Verfahren sind in der Lage, die Zeigegeste unter den aufgestellten Voraussetzungen für eine intuitive Interaktion vollständig automatisch zu detektieren. Ungelöst bleibt bisher allerdings noch das genannte Problem der Merkmalsextraktion und Klassifizierung in Echtzeit. Viele Anwendungen, die durch Gestenerkennung gesteuert werden sollen, sind nur schwer intuitiv durch die Verwendung von nur einer Geste bedienbar. Um mehr als nur eine einzige Handgeste während der Interaktion mit dem Computer verwenden zu können, wird deshalb in diesem Kapitel ein Verfahren entwickelt, dass in der Lage ist, verschiedene Gesten in Echtzeit zu unterscheiden. Dieses Verfahren basiert auf dem Training und der Analyse von zweidimensionalen Bildmerkmalen und deren Klassifikation. Nach der Bestimmung der verwendeten Geste werden die zur Interaktion notwendigen Parameter der Geste unter Verwendung eines Stereokamerasystems im Raum bestimmt und für die Interaktion bereit gestellt. Neben den notwendigen bildverarbeitenden Schritten zur Merkmalsextraktion erläutert dieses Kapitel auch die verwendeten Methoden zur Modellbildung und zur Unterscheidung der Gesten zur Laufzeit des Systems. Das hier vorgestellte Verfahren gliedert sich in die folgenden Schritte:

1. Segmentierung der Handgeste in den Bildern des Stereokamerasystems.

   Dieser Schritt des Verfahrens trennt die in den Kamerabildern sichtbare Hand vom Hintergrund der Szene und ermöglicht mittels eine Analyse und Verfeinerung der Segmente eine robuste Identifikation der Hand auf Bildbasis. Die so ermittelten Segmente dienen als Grundlage für die folgende Merkmalsextraktion der Geste.

2. Merkmalsextraktion der Geste auf Bildbasis.

   Dieser Schritt des Verfahrens bestimmt die für die Klassifikation und Klassierung der Gesten notwendigen Parameter. Für eine robuste und echtzeitfähige Merkmalsbestimmung erfolgt dieser Schritt auf Bildbasis unter Verwendung der Ergebnisse der

Segmentierung. Die so ermittelten Merkmalsvektoren der einzelnen Bilder werden zu einem einzigen Merkmalsvektor kombiniert und für die Klassifikation und Klassierung verwendet.

3. Klassifikation der Gesten durch geeignete Modellbildung.

   In diesem Schritt des Verfahrens wird aus einer Vielzahl von Trainingsdaten automatisch die Klassifikation der vordefinierten Gesten ermittelt. Dabei wird besonderer Wert darauf gelegt, dass die Trainingsphase für den Anwender zum einem möglichst kurz und zum anderen einfach und verständlich gehalten wird.

4. Klassierung eines Merkmalsvektors zur Bestimmung der verwendeten Handgeste.

   Neue, aus den Kamerabildern zur Laufzeit ermittelte Merkmalsvektoren werden anhand des zuvor errechneten Modells einer Klasse des Merkmalsraums zugeordnet. Eine Methode zur Eliminierung von Ausreißern verhindert dabei einen unerwünschten kurzzeitigen Wechsel der Geste.

5. Berechnung der zur Interaktion relevanten 3D-Parameter.

   Unter Verwendung der Informationen des Stereokamerasystems wird die zur intuitiven Interaktion notwendige Position der Geste im Raum ermittelt und zusammen mit dem Ergebnis der Klassierung der Anwendung bereit gestellt.

6.1 Einleitung

Die beiden bisher vorgestellten Verfahren verwenden eine Zeigegeste, um die Interaktion der Mensch-Maschine-Schnittstelle zu realisieren. Durch die Erweiterungen der Gestenerkennung und -verfolgung durch die Möglichkeit, Selektionsereignisse zu erzeugen, eröffnet sich eine Vielzahl von Interaktionsmöglichkeiten. Diese reichen von der reinen zweidimensionalen Bewegung eines Objektes auf dem Ausgabebildschirm bis zur Untersuchung eines dreidimensionalen Objekts mittels Rotation durch die Verwendung von ereignisauslösenden Schaltflächen (vergleiche auch Kapitel 9). Insbesondere durch die deiktische Art eignet sich die Erkennung der Zeigegeste für multimodale Anwendungen, die Sprach- und Gestenerkennung miteinander verknüpfen. Dennoch stößt die ausschließliche Verwendung der zeigenden Hand als Eingabemodalität in einigen Anwendungsklassen an ihre Grenzen. Insbesondere bei der Interaktion mit virtuellen dreidimensionalen Welten werden für eine sinnvolle Schnittstelle zwischen Anwender und Anwendung zusätzliche Informationen benötigt. Neben der Navigation in dreidimensionalen Räumen ist die Positionierung von virtuellen Objekten eine klassische Aufgabe. Mit den bereits vorgestellten Verfahren ist eine Positionierung im Raum durchaus möglich: Der Anwender zeigt auf das Objekt, um es so für eine Verschiebung zu markieren und bewegt es dann mittels der Zeigegeste im Raum an die gewünschte Stelle. Bei einem erneuten Selektionsereignis kann das Objekt dann wieder von der Geste gelöst und damit an der neuen Stelle positioniert werden. Nichtsdestotrotz ist diese Art der Objektverschiebung im Raum nicht besonders intuitiv. Dem Anwender muss zum einem erklärt werden, wie diese Technik funktioniert, zum anderen benötigt er in der Regel eine gewisse

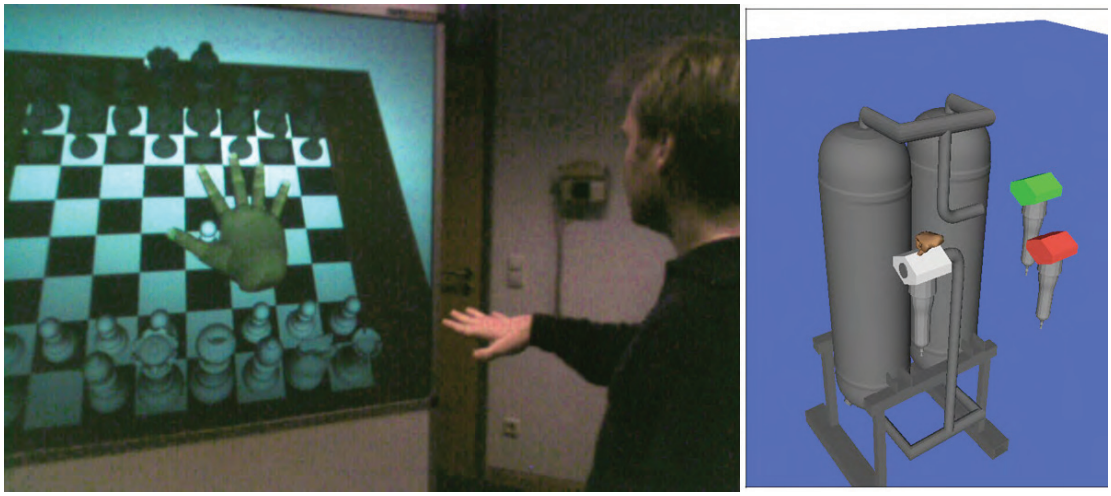

Abbildung 6.1: Anwendungen der Gestenklassifikation. Verwendet wird die Unterscheidung der offenen und der geschlossenen Hand, um Objekte im Raum zu bewegen. Links virtuelles Schachspiel (vergleiche Kapitel 9), rechts industrielle Anwendung zur Plazierung von Filterelementen.

Zeit des Übens, bis er die Technik beherrscht. Deutlich einfacher und intuitiver ist es, wenn der Anwender in die Lage versetzt wird, ein Objekt im Raum zu greifen, an eine andere Stelle zu bewegen und dort wieder loszulassen. Damit eröffnen sich eine Vielzahl von Anwendungen zur einfachen Interaktion mit 3D-Welten (siehe Abbildung 6.1). Von einer intuitiven Interaktion mit virtuellen 3D-Welten muss deshalb gefordert werden, eine Methode bereitzustellen, die in der Lage ist, eine geschlossenen Hand von einer geöffneten Hand als Synonyme für das Greifen und Loslassen von virtuellen Objekten zu unterscheiden.

Das folgende Verfahren erweitert daher die Interaktion mittels Zeigegeste um die Möglichkeit, unterschiedliche Gesten zu klassifizieren und im Raum zu verfolgen. Um dem Anwender einen einfachen Umgang mit der 3D-Welt zu ermöglichen, soll das System zu jeder Zeit eine visuelle Rückantwort bereit stellen, die dem Anwender nicht nur die derzeitige Position im Raum, sondern auch die entsprechende Geste anzeigt. Dies kann beispielsweise durch die Verwendung eines Modells der Hand in verschiedenen modellierten Handposen erreicht werden (siehe Abbildung 6.1). Das Verfahren beruht auf der Klassifikation und Klassierung von Merkmalen der in den Kamerabildern segmentierten Hand. Für die Klassifikation, also der Berechnung der für eine Unterscheidung der Gesten notwendigen Parameter, werden die Merkmale in einer kurzen Trainingsphase aufgezeichnet. Aus diesen Trainingsdaten wird dann das Klassifikationsmodell berechnet. Praktisch bedeutet dies, dass der Anwender, wenn er das System zum ersten Mal verwenden möchte, die zur Verfügung stehenden Handgesten für jeweils etwa zehn Sekunden benutzen muss. In dieser Zeit lernt das Verfahren automatisch die Eigenschaften und Eigenheiten des jeweiligen Anwenders anhand der extrahierten Merkmale zu unterscheiden. Bei diesem Schritt lernt also das System, wie der Anwender eine Geste versteht und verwendet. Für eine einfache Interaktion und die Akzeptanz des Verfahrens ist diese "Lernrichtung" wichtig. Das System passt sich hier dem Anwender an

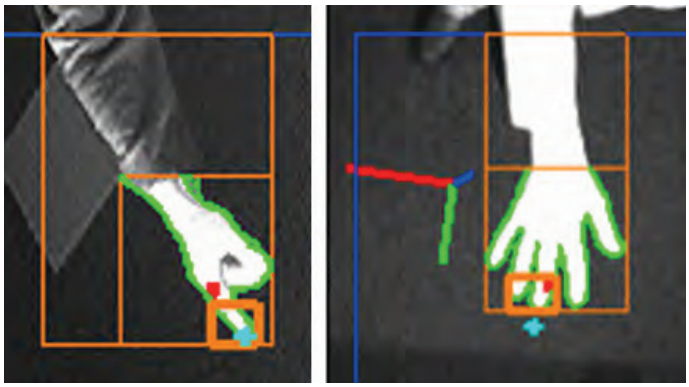

Abbildung 6.2: Segmentierung der Handgeste in drei Schritten, dargestellt durch orange Rechtecke. Überlagert ist außerdem die Begrenzungslinie des endgültig als Hand angenommenen Segments.

und nicht wie häufig üblich, andersherum. Nach der Trainingsphase kann der Anwender die angelernten Gesten für die Interaktion verwenden. In diesem Klassierungsschritt werden die Merkmalsvektoren neuer Kamerabilder ermittelt und anhand des Modells einer der erlernten Gesten zugeordnet.

6.2 Segmentierung der Hand

Der erste Schritt des Verfahrens zur Bestimmung der Merkmalsvektoren ist eine Trennung der Hand vom Hintergrund der Bilder durch eine schwellwertbasierte Segmentierung. Da für das System keine idealen Umgebungsbedingungen vorausgesetzt werden dürfen, kann es bei diesem Schritt vorkommen, dass in den Kamerabildern mehr als ein einziges Segment erzeugt wird. Oft entstehen durch beispielsweise Änderungen der Lichtverhältnisse zusätzlich zu dem Segment, das die Hand des Anwenders beschreibt, weitere Segmente, die nicht für die Merkmalsbestimmung der Hand verwendet werden sollen. Die Bestimmung, welche der Segmente nun für die folgende Merkmalsbestimmung herangezogen werden (im Folgenden auch als *Gestensegmente* bezeichnet), verläuft analog zum der im vorigen Kapitel 5 beschriebenen Methode: Durch die geeignete Wahl der Position und Orientierung der Kameras kann eine Zielrichtung innerhalb eines Segments für eine Suche nach dem Segmentendpunkt, also dem Bildpunkt, an dem ein Segment sein umschließendes Rechteck (*bounding box*) berührt, ermittelt werden. Eine Triangulierung der möglichen Punktkorrespondenzen identifiziert Segmente, die außerhalb des definierten Interaktionsvolumens des Anwenders liegen und verworfen werden können. Existiert mehr als nur ein 3D-Punkt innerhalb des Volumens, werden diejenigen Segmente gewählt, die den zum Ausgabegerät am nächsten liegenden 3D-Punkt erzeugen.
Die Segmentierung selbst erfolgt erneut auf Basis von Differenzbildern zwischen Referenzbildern, die bei Start des Systems aufgezeichnet wurden und den zur Laufzeit gültigen Kame-

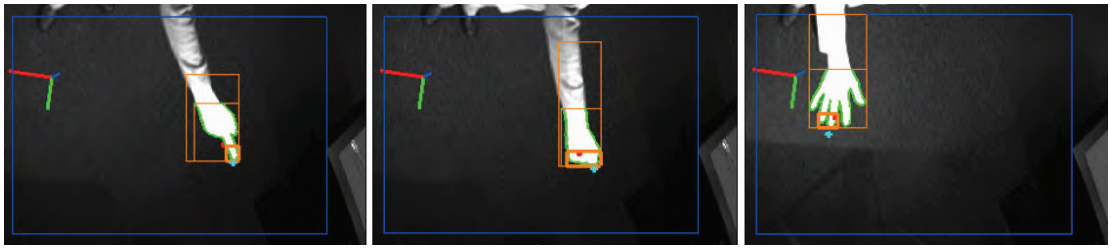

Abbildung 6.3: Segmentierungsergebnisse für drei verschiedenen Gesten. Neben den segmentumschließenden Rechtecken (orange) sind unter anderem auch die offenen Umrandungen der Segmente dargestellt (grün).

rabildern. Im Gegensatz zum Verfahren des vorigen Kapitels werden allerdings die Differenzbilder nicht aus den Kantenbildern, sondern aus den Originalkamerabildern selbst bestimmt. Grund dafür ist, dass für die folgende Merkmalsextraktion nicht nur die äußere Umrandung der Hand sondern die gesamt Fläche der Hand verwendet werden soll. Diese Art der Segmentierung ist etwas anfälliger gegenüber ungünstigen Beleuchtungsbedingungen. Aus diesem Grund kann das Interaktionsvolumen zusätzlich mit Infrarotscheinwerfern beleuchtet werden. Beispielsweise realisiert durch Kameraobjektive mit eingearbeiteten Infrarot-Leuchtdioden wird die Art der Beleuchtung von den meisten Anwender gar nicht wahrgenommen, zumindest aber als nicht störend empfunden. Durch die deutlich kürzere Strecke zwischen Kamera und Hand im Gegensatz zu der Strecke zwischen Kamera und Hintergrund der Szene wird die Hand des Anwenders sehr viel stärker beleuchtet und erscheint dadurch in den Kamerabildern auch heller als der Szenenhintergrund (siehe Abbildung 6.3). Dadurch kann eine binarisierende Segmentierung der Bilder einfach und robust durchgeführt werden.

Im Verfahren der Punktprojektion war mit der Fingerspitze ausschließlich ein einzelner Punkt des Segments für die Gestenerkennung relevant. In dem hier vorgestellten Verfahren ist es aber notwendig, die Hand vollständig als Segment zu beschreiben. Durch die rein schwellwertbasierte Segmentierung kommt es aber häufig zu einer Übersegmentierung, bei der die Hand und Teile des Unterarms in einem einzelnen Segment liegen. Ein weiterer Schritt der Segmentierung ist deshalb die Trennung der Hand vom Unterarm. Durch die vorgegebene Richtung der Interaktion zum Ausgabebildschirm hin und durch den gewählten Kameraaufbau kann hier eine rein bildbasierte Segmentreduktion durchgeführt werden, bei der anhand einer vorgegebenen Segmenthöhe die relevante Region des ursprünglichen Segments abgeschnitten wird. Die Abbildungen 6.2 und 6.3 zeigen das Segmentierungsergebnis an verschieden Gesten. Zu sehen sind die umschließenden Rechtecke (engl.: *bounding boxes*) der jeweiligen Segmente dargestellt durch orange Rechtecke. Das jeweils größte Segment beschreibt das ursprüngliche, der Hand zugeordnete Segment. Das Rechteck der mittleren Größe zeigt das Ergebnis der Trennung von Hand und Unterarm. Das kleinste der jeweils drei Rechtecke beschreibt die Region um die jeweiligen Segmentendpunkte. Für die folgende Berechnung der Segmentmerkmale wird bereits während der Segmentierung die Umrandung des Segments aufgezeichnet. In den Abbildungen 6.2 und 6.3 sind diese Konturlinien durch grüne Polygonzüge den Kamerabildern überlagert.

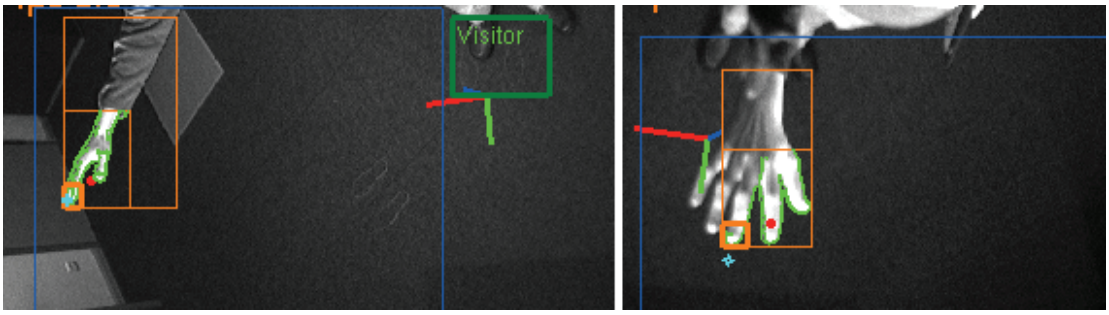

Abbildung 6.4: Fehlerhafter Schwellwert führt zu Fehlsegmentierung der Handgeste.

6.3 Merkmalsextraktion

Mit der korrekten Bestimmung der Gestensegmente stehen genügend Informationen für die Merkmalsberechnung zur Verfügung. Als für die Klassifikation verwendbare Merkmale werden sowohl Parameter verwendet, die direkt als Ergebnis der Segmentierung vorliegen, als auch Merkmale, die aus diesen Basisparametern berechnet werden können. Als Basisparameter stehen nach dem Segmentierungsschritt die folgenden Parameter zur Verfügung:

- Verhältnis zwischen Breite und Höhe des umschließenden Rechtecks des Segments
- Verhältnis zwischen der Länge der Segmentkontur und der Fläche des Segments
- Lage des nach Pixelintensitäten gewichteten Schwerpunktes des Segments
- Verhältnis zwischen Anzahl der Pixel des Segments und des umschließenden Rechtecks

Neben diesen Basisparametern werden noch die folgenden Parameter berechnet:

- Kompaktheit des Segments

  Die Kompaktheit eines Segments ist definiert als das Verhältnis zwischen der quadrierten Länge der Umrandung und Anzahl der Pixel des Segments:

$$Kompaktheit = \frac{(Laenge\ der\ Segmentumrandung)^2}{Flaeche\ des\ Segments} \tag{6.1}$$

  Durch die Art der Segmentierung und der daraus folgenden Ordnung der Randpunkte kann die Länge der Umrandung direkt berechnet werden. Dabei wird die Achter-Nachbarschaft zur Bestimmung der Länge verwendet, bei der diagonale Pixelnachbarn den euklidischen Abstand $\sqrt{2}$ aufweisen.

- Krümmung der Segmentumrandung (engl.: *curvature*)

  Die Krümmung der Umrandung des Segments wird berechnet als das Verhältnis zwischen der Anzahl der Pixel der Umrandung, an denen sich die Richtung signifikant ändert und der Anzahl der Pixel der Segmentumrandung. Die Richtungsänderungen können direkt aus der *chain code*-Kodierung (auch als Freeman-Kodierung [Fre61] bezeichnet) der Segmentumrandung ermittelt werden.

- Zentrale Momente bis zum Grad 2.

  Durch die Invarianz gegenüber der Translation eines Segments können die zentralen Momente [Hu62]

  $$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \tag{6.2}$$

  bereits als beschreibende Merkmale des Segments aufgefasst werden [Flu06].

- Ausrichtung des Segments

  Über die Berechnung von $\mu'_{11} = \mu_{11}/\mu_{00}$, $\mu'_{20} = \mu_{20}/\mu_{00}$ und $\mu'_{02} = \mu_{02}/\mu_{00}$ können außerdem Aussagen über die Ausrichtung des Segments gemacht werden. Der Winkel

  $$\theta = \frac{1}{2}\, tan^{-1}\left(\frac{2\,\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \tag{6.3}$$

  stellt somit die Richtung der längeren Seite eines minimalen, das Segment umschließenden Rechtecks (*oriented bounding box*) dar.

- Exzentrizität (Länglichkeit) des Segments

  Die Eigenwerte

  $$\lambda_i = \frac{\mu'_{20} - \mu'_{02}}{2} \pm \frac{\sqrt{4\,{\mu'_{11}}^2 + (\mu'_{20} - \mu'_{02})^2}}{2} \tag{6.4}$$

  der Kovarianzmatrix

  $$C = \begin{pmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{pmatrix} \tag{6.5}$$

  beschreiben mit

  $$\sqrt{1 - \frac{\lambda_2}{\lambda_1}} \tag{6.6}$$

  ein Maß für die Länglichkeit des Segments.

Die Auswahl der zur Klassifizierung und Klassierung verwendeten Segmentmerkmale erfolgt nach zwei Gesichtspunkten. Zum einen werden solche Merkmale gewählt, von denen erwartet werden kann, dass ein geeignetes Verfahren sie zur Unterscheidung der Handgesten verwenden kann. So ist beispielsweise davon auszugehen, dass sich das Verhältnis von Umrandungslänge eines Segments zur Fläche des Segments für die geschlossenen und die geöffnete Hand deutlich unterscheidet. Gleiches gilt für diese beiden Gesten offensichtlich auch für die Krümmung der Segmentumrandung und die Kompaktheit. Zum anderen wird

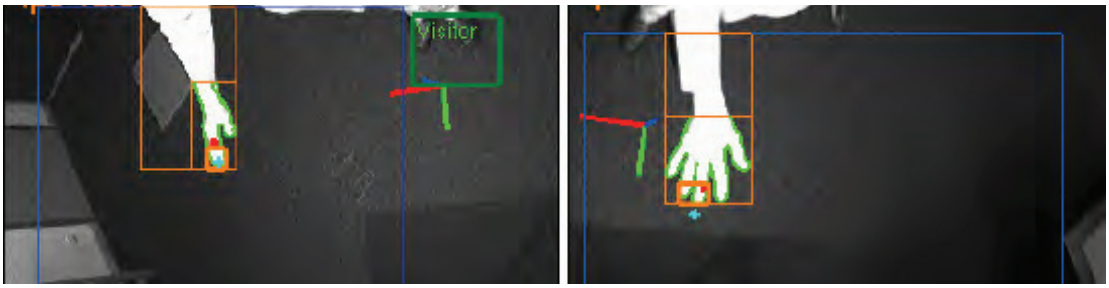

Abbildung 6.5: Kamerasichten erzeugen unterschiedliche Merkmalsausprägungen. Insbesondere die Orientierung der Hand führt zu signifikant unterschiedlichen Ergebnissen in gleichzeitig aufgenommenen Bildern des Stereokamerasystems.

die Auswahl der Merkmale aber durch ihre Berechenbarkeit begrenzt. Die maximalen Segmentsehnen parallel zu den Hauptachsen des Segments ist beispielsweise ein bekanntes und oft verwendetes segmentbeschreibendes Merkmal, dessen Berechenbarkeit allerdings einen hohen Rechenaufwand erfordert. Um die Echtzeitfähigkeit des Verfahrens während der Klassierung neuer Handgestensegmente sicherzustellen, muss daher auf Merkmale verzichtet werden, deren Berechnung keine interaktiven Bildwiederholraten garantieren können.
Wichtig ist, dass an dieser Stelle des Verfahrens keine explizite Entscheidung getroffen werden muss, welche der berechneten Merkmale für die Klassifikation eingesetzt werden. Solange die Berechenbarkeit in Echtzeit gewährleistet ist, können alle vorhandenen Parameter in die Merkmalsvektoren aufgenommen und der Klassifizierung bereitgestellt werden. Erst der zur Klassifizierung verwendete Algorithmus (vergleiche Abschnitt 6.4) berechnet, welche Merkmale und Kombinationen von Merkmalen eine bestmögliche Unterscheidung der Gesten anhand den Trainingsdaten zulassen.

6.3.1 Aufbau des Merkmalsvektors

Die beschriebenen Merkmale der Handgeste werden in den folgenden Schritten sowohl für die Modellbildung, also für die Klassifikation der Gesten, als auch für die Erkennung einer Geste zur Laufzeit, also für die Klassierung der Gesten verwendet. Nach der Berechnung der einzelnen Merkmale der Segmente in beiden Kamerabildern werden alle extrahierten Merkmale in einem Merkmalsvektor abgelegt. Die vorgegebene Größe dieses Vektors ergibt sich aus der Anzahl der bestimmten Merkmale und der Anzahl der verwendeten Kamerabilder. Die Reihenfolge der Merkmale innerhalb des Vektors ist dabei beliebig aber eindeutig zu wählen, damit eine Vergleichbarkeit zweier Vektoren gewährleistet ist. Allerdings führt die Bestimmung der Merkmale der Handgeste auf 2D-Bildbasis bei der Verwendung eines Stereokamerasystems mit zwei Kameras zu einem Entscheidungsproblem.

Die Segmente einer Handgeste und damit auch die extrahierten Merkmalen der Segmente von zwei zeitgleich aufgenommenen Bildern des Stereokamerasystems unterscheiden sich zum Teil deutlich in den beiden Kamerasichten. Grund für diese Unterscheide ist zumeist die

Orientierung der Hand des Anwenders in der Szene. Abbildung 6.5 zeigt dies an einem Beispiel: Beide Bilder zeigen die Kamerasichten auf die offene Hand des Anwenders zur gleichen Zeit. Die Segmente und ihre Merkmale sind dabei unterschiedlich. Durch die Lage der Hand im Raum sind im rechten Kamerabild alle Finger und der Daumen der Hand klar sichtbar, während im linken Bild die Hand als eine kompakt Fläche zu sehen ist und sich lediglich der Daumen von dieser Fläche trennen lässt. Diese Tatsache hat offensichtlich auch Auswirkungen auf die Merkmale der beiden Segmente und führt zu deutlich unterschiedlichen Ergebnissen bezüglich der Länge der Segmentumrandung, des Seitenverhältnisses der Segmente und des Krümmungsverhaltens der Segmentumrandung. Es ist aber davon auszugehen, dass der Anwender die gleiche Geste an einer anderen Stelle vor dem Ausgabebildschirm oder mit einer anderen Rotation der Hand den beiden Kameras zeigen kann, so dass die Werte der extrahierten Merkmale für die beiden Bilder vertauscht sind. Dies führt dazu, dass die Merkmalsvektoren dann nicht mehr vergleichbar sind. Eine einfache aber wie sich in den folgenden Abschnitten dieses Kapitels zeigen wird effektive Lösung für dieses Problem ist es, bei der Erstellung des Merkmalsvektors die Merkmalspaare der Größe nach zu sortieren. Nicht die Kamera, sondern die Größe eines Merkmals bestimmt also die Position innerhalb des Vektors. Für die $i = 1..m$ Merkmale x_i aus zwei Kamerabildern mit $x_i^g > x_i^k$ ergibt sich so bei vordefinierter Reihenfolge der Merkmale ein Merkmalsvektor

$$\vec{x} = \{x_1^g, x_1^k, ..., x_m^g, x_m^k\}, \tag{6.7}$$

der für die Klassifikation und Klassierung der Handgesten verwendet werden kann.

6.4 Klassifikation und Klassierung der Handgesten

Mit der Möglichkeit, aus einem Bildpaar des Stereokamerasystems die relevanten Segmente der Handgeste zu ermitteln und deren Merkmale in Echtzeit zu bestimmen, folgt in einem Trainingsschritt des Verfahrens eine Klassifikation von bekannten Merkmalsvektoren. Dieser Schritt der Modellbildung führt die Merkmalsvektoren mit dem Modellwissen zusammen. Voraussetzung dafür ist also, dass für eine gegebene Anzahl von Merkmalsvektoren die zugehörige Handgeste bereits bekannt ist. Die Modellbildung wird während einer Lernphase des Systems durchgeführt, wenn ein neuer Anwender die Gestenerkennung als Eingabemodalität verwenden möchte. Auch bei diesem Verfahren gilt, dass es zwar prinzipiell möglich ist, das Modell aus Trainingsdaten von mehreren unterschiedlichen Anwendern zu bestimmen und im Folgenden bei neuen Anwendern auf die Lernphase zu verzichten. Nichtsdestotrotz gilt, dass bei einer individuellen Modellbildung die Erkennungsrate deutlich ansteigt und so einen robusteren Umgang mit dem System gewährleistet. Im Gegensatz zu dem in Kapitel 4 vorgestellten Verfahren ist hier allerdings keine aufwendige manuelle Bereitstellung der Trainingsdaten notwendig. Durch die Möglichkeit, die Merkmalsvektoren in dem kalibrierten Stereokamerasystem automatisch zu ermitteln, kann demnach dieser Schritt vollautomatisch erfolgen. Der Anwender wird während der Trainingsphase gebeten, die vorgegebenen Handgesten für jeweils einige Sekunden vor dem Ausgabebildschirm auszuführen. Dadurch sind zu allen Merkmalsvektoren die zugehörigen Handgesten bekannt und können so klassifiziert

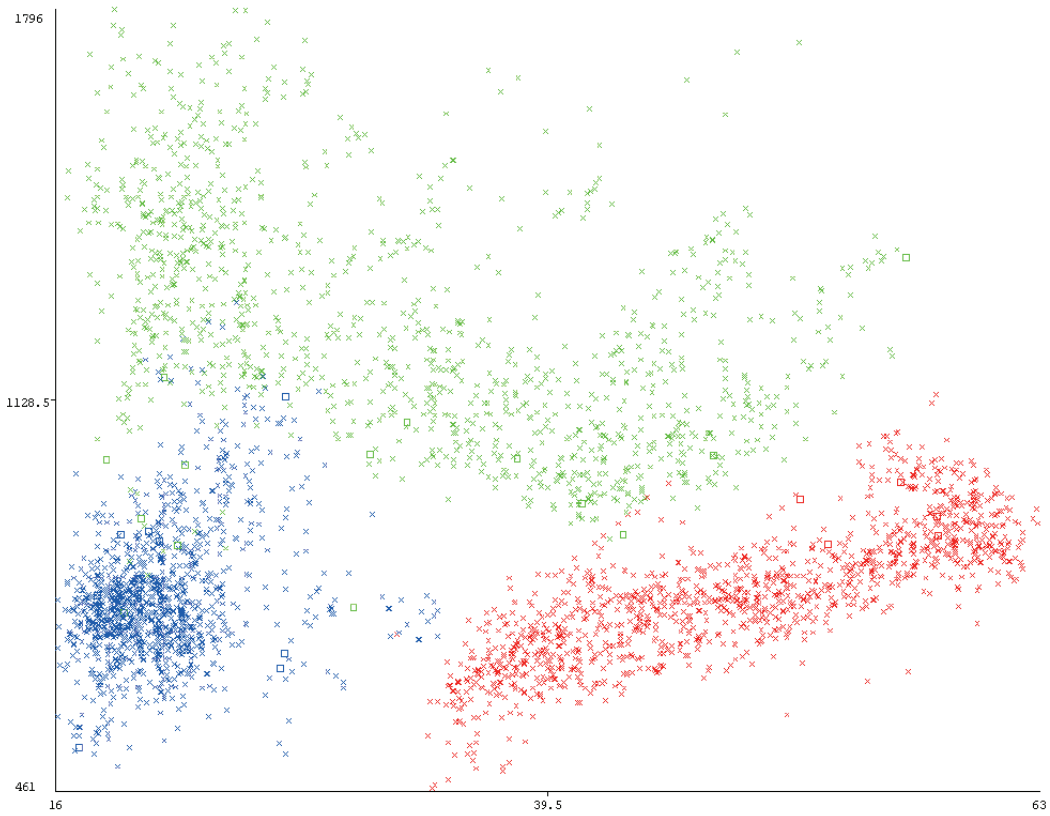

Abbildung 6.6: Verteilung von zwei Merkmalen einer Trainingssitzung. Blau stellt die Zeigegeste, Rot die geschlossene Hand und Grün die geöffnete Hand dar. Alle Kreuze wurden in einer Kreuzvalidierung korrekt klassiert, die Quadrate sind Fehlklassierungen.

werden.

In der Mustererkennung werden als Eingabe für die Klassifikation häufig Hidden Markov Modelle [Fin03, SP95, EAHAM08, MA07], künstliche neuronale Netze [RW08, Kje97, FTJL08] oder Klassifizierungsverfahren wie Support-Vector-Maschinen [SS01, Vap99, LGS08, Mal08b, CMNP08a] verwendet. In dem hier entwickelten Verfahren wird der Naïve Bayes-Klassifikator verwendet, um unterschiedliche Handgesten zu unterscheiden. Der oft zu *Data Mining*-Zwecken verwendete Klassifikator [WF05] geht von der Tatsache aus, dass alle Attribute des Merkmalsvektors zum einen gleich wichtig sind und zum anderen in Bezug auf ihren Klassenwert statistisch unabhängig sind. Zwar wird insbesondere die Annahme der Unabhängigkeit in der Praxis häufig verletzt, allerdings liefert das Verfahren oft trotzdem eine sehr hohe Erkennungsrate. Insbesondere durch die schnelle Berechenbarkeit sowohl während der Klassifizierung als auch der Klassierung eignet sich die Methode für interaktive Anwendungen mit dem Anspruch auf Echtzeitfähigkeit.

Der Naïve Bayes-Klassifikator beruht auf dem aus der Wahrscheinlichkeitsrechnung bekannten Bayes-Theorem über bedingte Wahrscheinlichkeiten. Das Klassifikationsproblem in der hier behandelten Gestenerkennung stellt sich im Fall der Naïve-Bayes-Klassifikation als die Wahrscheinlichkeit, dass eine Handgeste, repräsentiert durch einen Merkmalsvektor $\vec{x} = (x_1, ..., x_m)$ im m-dimensionalen Merkmalsraum, in die Klasse der Geste c_j einzuordnen ist. Gesucht ist also die bedingte Wahrscheinlichkeit

$$P(c_j|\vec{x}) = P(c_j)\frac{P(\vec{x}|c_j)}{p(\vec{x})} \tag{6.8}$$

oder anders ausgedrückt die Wahrscheinlichkeit, dass der Merkmalsvektor $\vec{x}$ durch die Geste c_j in den Kamerabildern ermittelt wurde. Die Wahrscheinlichkeit $P(c_j)$ lässt sich bei n Trainingsdokumenten leicht berechnen als

$$P(c_j) = \frac{|Gesten\ in\ c_j|}{n}. \tag{6.9}$$

Durch die große mögliche Anzahl verschiedener $\vec{x}$ ist die Berechnung von $P(\vec{x}|c_j)$ schwierig. Eine übliche Strategie ist es deshalb, anzunehmen, dass $P(\vec{x}|c_j)$ für alle c_j als

$$P(\vec{x}|c_j) = \prod_{i=1}^{m} P(x_i|c_j) \tag{6.10}$$

geschrieben werden kann. Dabei kann $P(x_i|c_j)$ geschätzt werden als

$$P(x_i|c_j) = \frac{n_c}{n_j} \tag{6.11}$$

wobei n_c die Anzahl der Vorkommen des Merkmals x_i in den zu c_j gehörenden Gesten und n_j die Anzahl der Merkmale in den zu c_j gehörenden Gesten ist. Damit ergibt sich weiterhin folgende Form zur Berechnung von $P(c_j|\vec{x})$

$$P(c_j|\vec{x}) = P(c_j)\frac{\prod_{i=1}^{m} P(x_i|c_j)}{P(\vec{x})} \tag{6.12}$$

Für die Klassierung eines neuen Merkmalsvektors $\vec{x}$ werden die Wahrscheinlichkeiten $P(c_j|\vec{x})$ für alle gegebenen Handgesten des Modells berechnet und die Geste mit der höchsten Wahrscheinlichkeit unter $\vec{x}$ als Ergebnis gewählt.

Für das hier beschriebene Verfahren zur Unterscheidung von Zeigegeste, geschlossener und geöffneter Hand werden in der Trainingsphase zur Modellbildung die vom Anwender ausgeführten Gesten verwendet. Bei einer Dauer von etwa zwanzig Sekunden pro Geste bei einer Wiederholrate von zwanzig Bildern pro Sekunde ergeben sich pro Geste etwa 400 Merkmalsvektoren, die in das Modell eingehen. Die Klassifikation der Gesten mit dem Naïven Bayes-Klassifikator dauert weniger als fünfzehn Sekunden[1]. Zusammen mit der Dauer für die

[1]Verwendet wird die freie und offene Software "Weka 3: Data Mining Software in Java", http://www.cs.waikato.ac.nz/ml/weka/, [WF05]

Aufnahme der Trainingsgesten durch den Anwender ergibt sich eine Gesamtzeit von weniger als zwei Minuten, bis der Anwender das System nach dem Start zur Interaktion verwenden kann.

6.4.1 Validierung der Klassifizierung

Die Überprüfung der Methode zur Unterscheidung von Zeigegeste, offener und geschlossener Hand erfolgt anhand von Testdaten, die während einer Testsitzung während einer zweiten Lernphase aufgenommen werden, da hier sichergestellt werden kann, dass die korrekte Geste ausgeführt wird und die Ergebnisse so vergleichbar sind. Während die Merkmalsvektoren der Trainingsdaten zur Modellbildung verwendet wird, werden die Daten der Testsitzung zur Ermittlung der Erkennungsrate verwendet. Die folgende Tabelle 6.1 zeigt die Ergebnisse der Klassierung in einer Matrix. Insgesamt werden 1323 Bildpaare zur Validierung des Modells verwendet. Bei 69 fehlerhaft erkannten Gesten ergibt sich eine Erkennungsrate von 94.8% (siehe 6.2).

Tabelle 6.1: Klassierungsmatrix einer von den Trainingsdaten unabhängigen Testsitzung für den Naïven Bayes-Klassifikator.

Vergleich durchgeführte und vom System erkannte Geste			
	Zeigegeste	**Geschlossene Hand**	**Geöffnete Hand**
Zeigegeste	449	6	33
Geschlossene Hand	3	426	1
Geöffnete Hand	24	2	379

Zusätzlich kann die Erkennungsrate der beiden Sitzungen unabhängig voneinander mittels einer klassischen Kreuzvalidierung [Koh95] überprüft werden. Dafür wird aus dem Datensatz der einen Sitzung eine Hälfte der Merkmalsvektoren zufällig ausgewählt und als Trainingsdaten zur Modellbildung eingesetzt, die andere Hälfte wird als Testdaten zur Validierung verwendet. Die Kreuzvalidierung wird für beide aufgenommenen Datensätze durchgeführt und erreicht hier Erkennungsraten von 95,1% beziehungsweise 97,6% (siehe Tabelle 6.2).

Tabelle 6.2: Erkennungsrate für drei verschiedenen Handgesten.

Trainingsdaten	**Testdaten**	**Erkennungsrate**
Trainingssitzung	Kreuzvalidierung	95.1%
Testsitzung	Kreuzvalidierung	97.6%
Trainingssitzung	Testsitzung	94.8%

Sowohl durch die Wahl des Kameraaufbaus als auch durch eine individuell unterschiedliche Verwendung der Gesten entstehenden für eine Geste während der Interaktion unterschiedliche Merkmalsvektoren. Der Grund dafür ist, dass sich während der Interaktion durch

eine Änderung der Position und der Orientierung der Hand im Raum auch das Erscheinungsbild der Gestensegmente in den beiden Kamerabildern ändert und daraus folgend auch die Merkmalsvektoren variieren. Ein einfaches Beispiel für diesen Sachverhalt ist die Verwendung einer Zeigegeste. Je nachdem, auf welchen Punkt des Ausgabebildschirms der Anwender zeigt, ändert er dabei neben der Position auch die Rotation der Hand im Raum. Es ist leicht vorstellbar, dass sich bei einer Drehung der Hand in Richtung einer der Kameras, im Segment immer weniger vom Zeigefinger zu erkennen ist, je weiter die Hand der Kamera zugewandt wird, sich der Zeigefinger also scheinbar "verkürzt".
Wird während der Trainingsphase des Systems zur Modellbildung nicht die gesamte mögliche Interaktionsfläche verwendet, sind eben diese Merkmalsvektoren nicht an der Modellbildung beteiligt und führen mit höherer Wahrscheinlichkeit zu Fehlentscheidungen während der Klassierung neuer Gesten. Abbildung 6.7 zeigt beispielhaft einen möglichen Grund, warum nicht alle Gesten korrekt erkannt werden können. In dem dargestellten Fall der Klassierung der offenen Hand wurde während der Trainingsphase die rechte obere Ecke des Bildschirms ausgelassen, wodurch es bei der späteren Interaktion an dieser Stelle zu vermehrt auftretenden Fehlentscheidungen des Systems kommt.

6.4.2 Nachverarbeitungsschritte

Mit einer Erkennungsrate von etwa 95% erreicht das Verfahren eine für die Interaktion ausreichend hohe Erkennungsrate. Nichtsdestotrotz bedeutet dies auch, dass bei einer Wiederholrate des Systems von etwa zwanzig Bildern pro Sekunde im Schnitt einmal pro Sekunde eine Geste fehlerhaft ermittelt wird. Wie bereits erwähnt, ist es für eine intuitive Interaktion wichtig, dass der Anwender zu jeder Zeit eine visuelle Rückantwort des Systems erhält. Für die Unterscheidung der drei hier verwendeten Gesten bedeutet dies, dass dem Anwender eine virtuelle Repräsentation der Hand in der jeweiligen Ausprägung der Pose auf dem Ausgabebildschirm angezeigt wird (siehe Abbildung 6.1). Fehlentscheidungen können also zu einem kurzzeitigen Wechsel der Pose der dargestellten virtuellen Hand und führen und stören dadurch die intuitive Interaktion durch den Anwender. Aus diesem Grund werden neu ermittelte Gesten einer Nachverarbeitung unterzogen. Ähnlich der im Abschnitt 5.3.1 beschriebenen Methode zur Selektion von Objekten mittels Zeigegeste wird für die Entscheidung ein Vektor vorgegebener Länge verwendet, in dem die letzten n Posen der Hand gespeichert werden. Nicht die für ein Bildpaar bestimmte Geste wird vom System nach außen gegeben, sondern die Geste mit der größten Häufigkeit innerhalb des Vektors. Damit kann ein kurzzeitiges Umschalten von einer zu einer anderen Geste vermieden werden. Nachteil dieses Vorgehens ist allerdings, dass dadurch auch eine Verzögerung bei einem tatsächlich auftretenden und damit gewünschten Wechsel der Geste erzeugt wird. Bei einer Vektorlänge von sieben Gesten und einer durchschnittlichen Wiederholrate des Systems von zwanzig Bildern pro Sekunde ergibt sich eine Verzögerung von vier Bildpaaren, bis die vom System nach außen gegebene Geste der realen Handgeste entspricht. Das entspricht einer Verzögerungszeit von ca. 200 Millisekunden, die von den meisten Anwendern als nicht störend wahrgenommen werden.

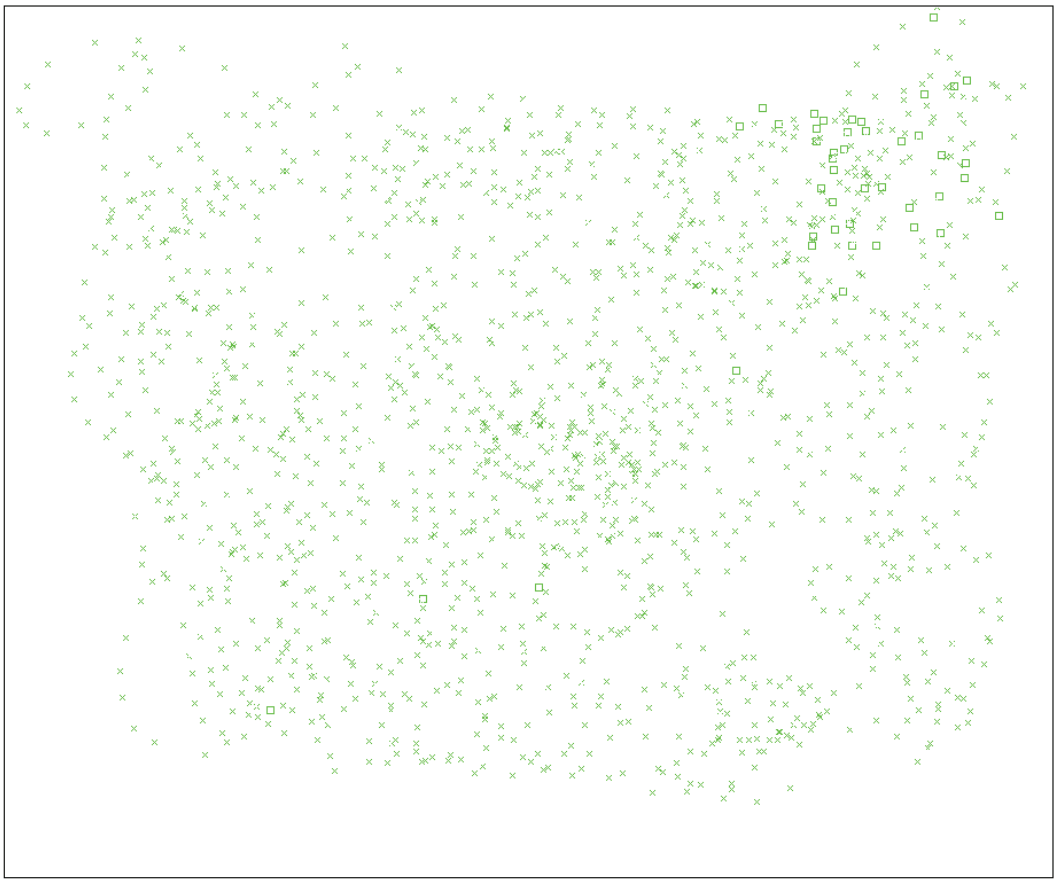

Abbildung 6.7: Projektion von klassifizierten Datensätzen der geöffneten Hand auf den Ausgabebildschirm. In der rechten oberen Ecke des Bildschirms treten vermehrt Klassifizierungsfehler (dargestellt durch Quadrate) aufgrund fehlender Trainingsdaten auf.

6.5 Berechnung der 3D-Parameter

Mit den klassifizierten Gesten lassen sich nun in einem letzten Schritt des Verfahrens die für eine intuitive Interaktion relevanten Parameter bestimmen. Neben der Pose selbst muss auch die Position der Hand im Raum bestimmt werden, um eine Interaktion mit einer virtuellen 3D-Welt zu ermöglichen. Unter Verwendung des kalibrierten Stereokamerasystems wird daher mit der bereits im Abschnitt 5.2.1 des Kapitels 5 beschriebenen Methode der 3D-Rekonstruktion der Segmentendpunkte die Position der Hand im Raum bestimmt. Es werden also erneut auf Bildbasis die Pixel ermittelt, an denen in vorgegebener Suchrichtung die Segmente ihre umschließenden Rechtecke berühren. Die Triangulierung des so entstehenden Punktepaares wird als 3D-Position der Geste verwendet. Einfacher und damit scheinbar logischer wäre es, die bereits während der Segmentierung berechneten Schwerpunkte der Segmente zur Triangulierung zu verwenden. Allerdings führt diese Art der Positionsbestimmung insbesondere bei einem Wechsel der verwendeten Geste zu plötzlichen und als störend

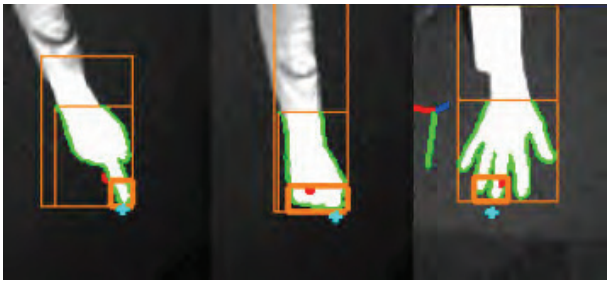

Abbildung 6.8: Position der Geste im Raum durch Rekonstruktion der Handsegmentendpunkt. Neben den Segmentierungsergebnissen und der Umrandung der Geste ist jeweils auch die geglättete Projektion der 3D-Position der Geste als cyan Punkt dargestellt.

empfundenen Änderungen der 3D-Position, da sich auch in den Kamerabildern die Gestensegmente innerhalb sehr kurzer Zeit deutlich ändern. Dennoch kann es bei der Berechnung der 3D-Postion durch sowohl leichte Bewegungen der Hand als auch durch Ungenauigkeiten der bildverarbeitenden Schritte zu einer unerwünschten Zitterbewegung der visuellen Rückantwort kommen, die eine intuitive Interaktion stört. Daher werden auch in diesem Verfahren die glättenden Nachverarbeitungsschritte wie in Abschnitt 5.2.3 des vorigen Kapitels beschrieben, angewendet. Im Gegensatz zu den zweidimensionalen Ergebnispunkten der projizierten Fingerspitze werden hier allerdings die 3D-Positionen der Geste in einem Vektor gespeichert und zur Stabilisierung der Handposition geglättet.
Als visuelle Rückantwort kann beispielsweise innerhalb einer Anwendung, die durch das System gesteuert wird, ein beliebiges virtuelles 3D-Modell der menschlichen Hand verwendet werden, für das die verwendeten Gesten vorab in verschiedenen Posen modelliert wurden (siehe auch Abbildung 9.11). Durch die Tatsache, dass in dem hier vorgestellten Verfahren allerdings die Orientierung der Hand nicht berechnet werden kann, beschränkt sich die visuelle Rückantwort des Systems dann auf eine Echtzeitdarstellung der verwendeten Geste und ihrer Position im Raum.

6.6 Zusammenfassung

In diesem Kapitel wurde ein Verfahren zur Unterscheidung von verschiedenen Handgesten in Echtzeit entwickelt. Nachdem die Verfahren der beiden vorigen Kapitel bereits das Problem der vollständig automatischen Suche nach der Handgeste gelöst haben, adressiert das hier vorgestellte Verfahren zusätzlich die Problematik der Merkmalsextraktion und Klassifikation, die für ein Gestenerkennungssystem erforderlich sind, das für eine intuitive Interaktion zwischen Mensch und Computer eingesetzt werden soll. Dabei stehen die vorgestellten Verfahren aber nicht für sich alleine, sondern können durchaus kombiniert und gemeinsam eingesetzt werden. So ist beispielsweise die Klassifikation der geschlossenen und offenen Hand und einer Zeigegeste sinnvoll, um für eine Interaktion mit dreidimensionalen virtuellen Welten ein möglichst großes Spektrum an Interaktionsmöglichkeiten zur Verfügung zu stellen. In diesem Fall können die Verfahren der *Merkmalbasierten Gesten-Klassifikation* und der *Interaktion*

durch Punktprojektion kombiniert verwendet werden. Zunächst erreicht die Klassifikation eine Entscheidung, welche der bereit gestellten Gesten derzeit verwendet wird. Für den Fall, dass hier die Zeigegeste als Interaktionsform ermittelt wurde, wird das Verfahren der Punktprojektion zugeschaltet, um neben der Postion der Hand im Raum auch die Zeigerichtung und damit den Interaktionspunkt auf dem Ausgabebildschirm zu berechnen. Für den Fall, dass die offene oder die geschlossene Hand als derzeitige Geste erkennt wird, können die ermittelten Parameter verwendet werden, um virtuelle Objekt im Raum aufzunehmen und an einer anderen Stelle des Raums wieder abzulegen.
Aus technischer Sicht beruht das in diesem Kapitel beschriebene Verfahren auf der Einordnung von Merkmalsvektoren durch einen Naïven Bayes-Klassifikator. Während einer initialen Lernphase des Systems werden Merkmalsvektoren mit Informationen über die einzelnen Gesten gespeichert und zur Modellbildung verwendet. Diese Lernphase verlangt lediglich, dass der Anwender die einzelnen Gesten für einige Sekunden vor dem Ausgabebildschirm ausführt. Die gesamte Lernphase beträgt inklusive der Modellbildung selbst in der Regel weniger als zwei Minuten und ermöglicht so auch neuen Anwendern des Systems, schnell mit der eigentlichen Interaktion zu beginnen. Um die Echtzeitfähigkeit des Verfahrens zu gewährleisten, werden die Merkmale zunächst im Zweidimensionalen, also auf Bildbasis bestimmt. Dazu werden unter Verwendung der Informationen des Stereokamerasystems die Segmente der Geste in den Kamerabildern bestimmt und relevante Parameter der Segmente berechnet. Das aus dem Ansatz der zweidimensionalen Merkmalsextraktion resultierende Problem der Vergleichbarkeit von Merkmalsvektoren wird dadurch gelöst, dass die einzelnen Merkmale der Größe nach sortiert und in einer vorgegebenen Reihenfolge im Merkmalsvektor abgelegt werden. Die Klassierung der drei verwendeten Gesten (Zeigegeste, offene und geschlossene Hand) erreicht während der Klassierung neuer Merkmalsvektoren eine Erkennungsrate von etwa 95%. Die Rate kann durch eine Eliminierung von Ausreißern durch eine Häufigkeitsanalyse der Ergebnisse weiter erhöht werden, ohne dass es zu nennenswerten Verzögerungen des Systems kommt. Neben der Pose selbst wird unter Verwendung des Stereokamerasystems auch die Position der Handgeste im Raum bestimmt. Mit den im vorigen Kapitel bereits beschriebenen, glättenden Nachverarbeitungsschritten werden eventuelle Zitterbewegungen der Hand reduziert und ermöglichen so eine einfach und intuitive Interaktion mit virtuellen dreidimensionalen Welten.
Mit dem Verfahren der *Merkmalbasierten Gesten-Klassifikation* ist bereits ein großer Teil der in der wissenschaftlichen Literatur noch nicht ausreichend behandelten Probleme für eine intuitive Interaktion durch videobasierte Gestenerkennung gelöst, da insgesamt sowohl für das Problem der automatischen Initialsuche einer Geste als auch für das Problem der Merkmalsbestimmmung und Klassifikation Lösungen gefunden werden konnten, die eine intuitive Verwendung von Handgesten ermöglichen. Allerdings bleibt eine Klasse von Anwendungen, für welche die bisher in dieser Arbeit entwickelten Verfahren nicht ausreichend sind, um eine intuitive Interaktion zu realisieren. Für Anwendungen, welche die Erkennung einer hohen Anzahl von ähnlichen Handposen voraussetzen, ist das Verfahren der Klassifikation durch Merkmalsvektoren nicht geeignet. Da bereits leichte Änderungen der Position und der Orientierung der Hand im Raum in Bezug auf die Kamerabilder zu unterschiedlichen Merkmalswerten führen, können nur Gesten automatisch voneinander getrennt werden, die auch bei einer hohen Varianz der Merkmalswerte signifikant unterscheidbar sind. Sollen aber

beispielsweise dynamische Prozesse einer Handgeste erkannt werden, die sich durch eine kontinuierliche Änderung der Gelenkwinkel der Hand auszeichnen, ist dies offensichtlich nicht der Fall. Daher beschäftigt sich das Verfahren der *Interaktion durch Momenten-Analyse* im folgenden Kapitel mit dem Problem, dynamische Prozesse einer Handgeste so zu erkennen und zu beschreiben, dass sie unter den Voraussetzungen für eine intuitive Interaktion verwendbar sind.

Kapitel 7

Interaktion durch Momenten-Analyse

Nachdem in den vorigen drei Kapiteln Verfahren zur intuitiven Gestenerkennung entwickelt wurden, die ausschließlich auf der Erkennung und Verfolgung von einzelnen, statischen Handgesten beruhen, wird in diesem Kapitel ein Verfahren vorgestellt, dass die Dynamik von Handgesten mit allen Freiheitsgraden der menschlichen Hand untersucht und erkennt. Um die Voraussetzungen und Anforderungen an eine intuitive Interaktion zu erfüllen, wird dabei die Dynamik einer vorgegebenen Geste auf eine diskrete Anzahl von statischen Einzelposen reduziert. Diese Reduktion des vorgegebenen Suchraums ist insbesondere für die Forderung der Echtzeitfähigkeit notwendig, da ein allgemeingültiges Verfahren zur Erkennung von Handposen in allen Freiheitsgraden der menschlichen Hand in Echtzeit in der wissenschaftlichen Literatur nicht beschrieben ist. In dem hier entwickelten Verfahren wird die Echtzeitfähigkeit des Systems durch die Kombination von zwei verschiedene Ansätzen erreicht:

1. Durch die Diskretisierung des dynamischen Prozesses einer Geste auf eine vorgegebene Anzahl von statischen Einzelposen der Hand kann der Suchraum für das Verfahren deutlich eingeschränkt werden, ohne dass der visuelle Eindruck einer kontinuierlichen Bewegung für den Anwender eingeschränkt wird.

2. Durch die Verwendung der Grafikprozessoren des Rechners (GPU) als Co-Prozessoren für das Systems werden die notwendigen Berechnungszeiten der bildverarbeitenden Schritte des Verfahrens deutlich gesenkt. Dieser Schritt ermöglicht eine um ein Vielfaches höhere Verarbeitung von Bildinformationen gegenüber der alleinigen Verwendung des Hauptprozessors (CPU).

Das hier vorgestellte Verfahren arbeitet in zwei aufeinanderfolgenden Schritten. Nach der Lokalisation und Segmentierung der Hand in den Bildern des kalibrierten Stereokamerasystems wird die bestmöglich passende Handpose über eine Analyse von Pseudo-Zernike-Momenten der Gestensegmente in einem vorab berechneten Datensatz von künstlich erzeugten Handsilhouetten ermittelt. Die Erzeugung dieser Vergleichs-Silhouetten wird einmalig nach der Kalibrierung des Stereokamerasystems durch die Projektion eines generischen X3D-Handmodells

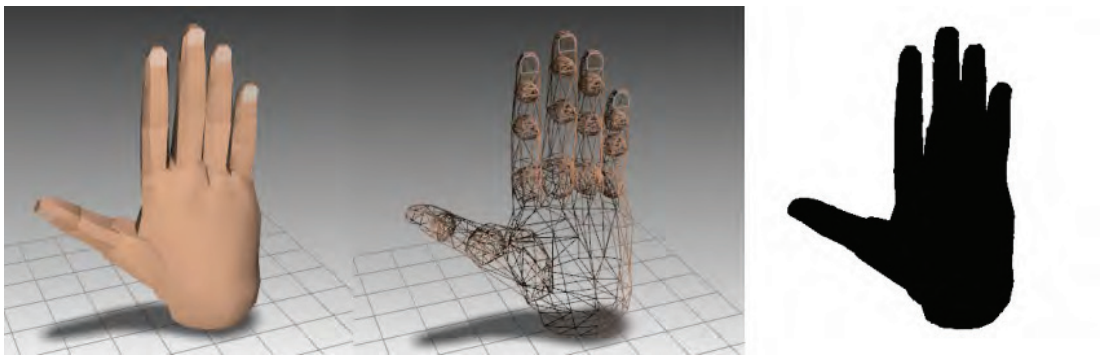

Abbildung 7.1: Zur Silhouettenerzeugung verwendetes X3D-Handmodell mit knapp 1600 Polygonen (von links: Flat Shading, Drahtgittermodell und invertiert binarisierte Silhouette).

unter Berücksichtigung der intrinsischen und extrinsischen Kameraparameter erzielt (siehe Abbildung 7.1). Da die Pseudo-Zernike-Momente invariant bezüglich Translation und Rotation sind, wird in einem zweiten Schritt des Verfahrens die tatsächliche Position und Orientierung der Hand im Raum mittels des bereits in Kapitel 4 vorgestellten nichtlinearen Optimierungsverfahrens *Simulated Annealing* bestimmt. Das gesamte Verfahren wird anhand der für die intuitive Interaktion wichtigen dynamischer Gesten, dem Öffnen und Schließen des Pinzettengriffs, also dem Zugreifen mit Hilfe von Daumen und Zeigefinger, überprüft.

7.1 Einleitung

Die im vorigen Kapitel verwendeten Handgesten der geschlossenen und der vollständig geöffneten Hand als Synonyme für das Zugreifen und Loslassen von virtuellen Objekten sind einfache und von den meisten Anwendern intuitiv verständliche Gesten, um Objekte im dreidimensionalen virtuellen Raum zu bewegen. Dennoch müssen diese Gesten als grobe Näherungen für die eigentlichen Gesten des Anwenders verstanden werden. Insbesondere kleine Objekte werden im Realen nicht mit der kompletten Hand, sondern oft mit dem Pinzettengriff ausschließlich mit Daumen und Zeigefinger gegriffen. Für den Pinzettengriff spielt die Rotation der Hand im Raum offensichtlich eine entscheidende Rolle, da das Zugreifen und Loslassen sich durch die Position der Daumen- und Fingerkuppe in Bezug auf das Objekt bestimmt. Um in einem solchen dynamischen Prozess den exakten zeitlichen Moment zu erfassen, an dem das Objekt aufgenommen oder losgelassen wird, also die Daumen- und Fingerkuppe die Oberfläche des Objektes berühren oder verlassen, müssen neben feinen Abstufungen der Handorientierung auch einzelne Gelenkwinkel des Handmodells präzise ermittelt werden. Einfache Segmentbeschreibungen wie in Kapitel 6 verwendet, sind für diesen Zweck nicht mehr ausreichend. Für das in diesem Kapitel beschriebene Verfahren werden aus diesem Grund höhere statistische Momente verwendet, die eine feine Unterteilung unterschiedlicher Gestensegmente zulassen.

7.2 Pseudo-Zernike-Momente

In der Bildverarbeitung werden statistische Momente häufig verwendet, um Objektsegmente zu beschreiben und um bereits bekannte Objekte in einem segmentierten Bild zu suchen und zu erkennen. Die Anwendungen reichen dabei von der Erkennung von Schriftzeichen [ZS06] über die Identifizierung von Produktverpackungen [KK05] und der Klassifizierung von gezeichneten Skizzen [HN04] bis hin zur Unterscheidung von verschiedenen Gesichtsmimiken [HL00]. Einfache zentrale Momente bis zur Ordnung 2 wurden bereits im vorigen Kapitel zur Merkmalsextraktion von Gestensegmenten und deren Klassifikation verwendet. Die in dem hier beschriebenen Verfahren verwendeten Pseudo-Zernike-Momente werden ebenfalls zur Merkmalsbestimmung der segmentierten Handsilhouetten verwendet. Allgemein kann ein Moment der Ordnung $(p+q)$ für eine stetige Funktion $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ in der Form

$$M_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi_{pq}(x,y) f(x,y)\, dx\, dy \qquad \text{mit} \qquad p,q = 0,1,2,... \tag{7.1}$$

beschrieben werden, wobei $\phi_{pq}(x,y)$ als die Basisfunktion der Momente aufgefasst wird. In der Bildverarbeitung können für $f(x,y)$ die Pixel eines binarisierten Bildes P_{xy} verwendet werden. Durch diese Diskretisierung erhält man

$$M_{pq} = \sum_x \sum_y \phi_{pq}(x,y) P_{xy} \qquad \text{mit} \qquad p,q = 0,1,2,.... \tag{7.2}$$

Abhängig von der Wahl der Basisfunktion unterscheidet man orthogonale und nicht-orthogonale Momente. Orthogonalität für zwei stetige Funktionen $y_n : A \rightarrow \mathbb{R}$ und $y_m : A \rightarrow \mathbb{R}$ mit $n, m \in \mathbb{N}, A \subseteq \mathbb{R}$ ist durch

$$y_n \perp y_m \Leftrightarrow \int_{x \in A} y_n(x) y_m(x)\, dx = 0 \tag{7.3}$$

gegeben. Im diskreten Fall der Verarbeitung von Segmentinformationen kann das Integral erneut durch durch eine entsprechende Summe ersetzt werden.
The und Chin [TC88] haben mit geometrischen, Legendre-, Zernike-, Pseudo-Zernike-, komplexen und Rotationsmomenten sechs verschiedene Arten von orthogonalen und nicht orthogonalen Momenten verglichen und konnten nachweisen, dass Zernike- und insbesondere Pseudo-Zernike-Momente in Hinblick auf Rauschempfindlichkeit, Redundanz und Repräsentationskapazität die besten Ergebnisse liefern.
Zur Berechnung von Zernike-Momenten werden mit den Zernike-Polynomen [Zer34] als Basisfunktionen komplexe, orthogonale Polynome, die senkrecht auf dem Einheitskreis $x^2+y^2 \leq 1$ verlaufen, verwendet. Die Zernike-Momente entstehen durch die Projektion der Bildfunktion auf diese orthogonalen Basisfunktionen. Pseudo-Zernike-Momente sind ähnlich wie die Zernike-Momente definiert und basieren auf einer von Bhatia und Wolf [BW54] vorgeschlagenen Abwandlung der ursprünglichen Zernike-Polynome. Die Pseudo-Zernike-Basisfunktion der Ordnung p und Wiederholung q ist gegeben durch:

$$W_{pq}(r \cdot cos\theta, r \cdot sin\theta) = R_{pq} \cdot e^{iq\theta} \tag{7.4}$$

Abbildung 7.2: Diskretisierung des Pinzettengriffs durch statische Einzelposen. Zu sehen sind sechs der insgesamt sechzehn Posen, die für die Rekonstruktion verwendet werden.

mit $p \geq 0, q \leq p$ und $q \in \mathbb{N}$. Die radialen Polynome können dabei mit

$$R_{pq} = \sum_{s=0}^{p-|q|} (-1)^s \frac{(2 \cdot p + 1 - s)!}{s!(p - |q| - s)!(p + |q| + 1 - s)!} r^{p-s} \tag{7.5}$$

berechnet werden. Des Weiteren gilt:

$$\langle W_{nm}, W_{pq} \rangle = \frac{\pi}{n+1} \delta_{np} \delta_{mq} \tag{7.6}$$

Damit sind die Pseudo-Zernike-Momente definiert als

$$P_{pq} = \frac{\pi}{n+1} \langle f, W_{pq} \rangle = \frac{\pi}{n+1} \int \int f(x,y) W^*_{pq}(x,y)\, dx\, dy \tag{7.7}$$

Für ein diskret vorliegendes Bild oder ein binäres Bildsegment können die Pseudo-Zernike-Momente der Ordnung p und Wiederholung q somit durch

$$P_{pq} = \frac{\pi}{n+1} \sum_x \sum_y P_{xy} W^*_{pq}(x,y) \tag{7.8}$$

berechnet werden. Effiziente Berechnungsmöglichkeiten von Pseudo-Zernike-Momenten insbesondere für den Einsatz in der Bildverarbeitung und der Computer Vision sind in [LC03, CRM03, PBKM07] beschrieben worden und bilden die Grundlage für die Umsetzung in dem hier vorgestellten Verfahren zur Gestenerkennung.

7.3 Verfahren der Momentenanalyse

Um den dynamischen Prozess eine Geste wie dem Öffnen oder Schließen des Pinzettengriffs zu rekonstruieren, wird zunächst der Suchraum des Verfahrens eingeschränkt. Theoretisch ist in dem kinematischen Modell der menschlichen Hand mit 26 Freiheitsgraden eine unendlich große Anzahl von unterschiedlichen Handposen denkbar. Um den Rechenaufwand für der Erkennungsprozess zu verkleinern und damit die Echtzeitfähigkeit des Verfahrens zu gewährleisten, wird der Prozess der dynamischen Geste in eine fest vorgegebene Anzahl von

Abbildung 7.3: Silhouetten einer einzelnen Pose mit unterschiedlichen Rotationen. Zu sehen sind neun verschiedene Rotationen der Hand um die z-Achse in 10°-Schritten.

statischen Einzelposen diskretisiert. Für den Pinzettengriff beispielsweise wird eine Aufteilung in sechzehn Einzelposen verwendet, so dass der visuelle Eindruck für den Anwender nicht gestört wird, er also während der Interaktion keine Sprünge zwischen zwei Einzelposen als störend wahrnimmt (siehe Abbildung 7.2). Die Bestimmung der jeweiligen Handpose beruht auf dem Vergleich der Pseudo-Zernike-Momente von realen und künstlich erzeugten Silhouetten der Hand. Zu Erzeugung der realen Handsilhouetten wird für die eingehenden Kamerabilder zunächst wie bereits in Kapitel 6 beschrieben unter Ausnutzung der Parameter des kalibrierten Stereokamerasystems die relevanten Segmente der Handpose ermittelt. Zur Erzeugung der künstlichen Silhouetten werden in einer virtuellen Umgebung Kopien der realen Kameras erzeugt und ein generisches X3D-Modell der menschlichen Hand im virtuellen Raum positioniert. Da die Pseudo-Zernike-Momente invariant bezüglich der Translation sind, ist in diesem Schritt nur eine sehr grobe Bestimmung der 3D-Position der Hand notwendig. Ein einfach zu ermittelnder Wert ist hier die Triangulierung der Schwerpunkte der beiden Gestensegmente. Durch Projektion des Modells auf die Bildebene, also einem Rendering-Schritt der virtuellen Szene entsteht so eine künstliche Silhouette der Handpose. Für das Rendering der virtuellen Szene wird in diesem Verfahren OpenSG [RVB02] eingesetzt.

Durch die Tatsache, dass für die realen und künstlich erzeugten Silhouetten die gleichen Kameraparameter verwendet werden, müssen die realen und künstlich generierten Silhouetten nahezu identisch sein. Allerdings spielt hier die Rotation der Hand eine entscheidende Rolle. Da bei fest vorgegebenen Kameraparametern bereits kleine Änderungen der Orientierung der Hand zu deutlich unterschiedlichen Silhouetten führen können, müssen für jede der vordefinierten Einzelposen eine Vielzahl unterschiedlicher Silhouetten durch Änderung der Rotation der Hand im virtuellen Raum erzeugt werden. Für die Interaktion mit einer Anwendung, die auf einem Ausgabegerät vor dem Nutzer des Systems dargestellt wird, ist die grundsätzliche Orientierung der Hand bereits vorgegeben. Für die Rotation des Handgelenks gelten zudem biomechanische Einschränkungen, die eine Drehung der Hand um die drei Hauptachsen nur um ungefähr 180°(x-Achse), 90° (y-Achse) und 180° (z-Achse) zulassen. Werden Rotationen als Punkte auf der Oberfläche einer Kugel aufgefasst, erzeugen diese Einschränkungen demnach ein Kugelkeil (sphärisches Zweieck) mit einem Öffnungswinkel von 90°. Für das hier vorgeschlagene Verfahren werden die einzelnen Silhouetten in 10°-Schritten generiert (siehe Abbildung 7.3). Pro Einzelpose ergeben sich demnach $19 * 19 * 10 = 3610$ oder insgesamt $3610 * 16 = 57760$ Silhouettenpaare, für welche die Pseudo-Zernike-Momente berechnet werden. Nach Abschnitt 7.2 werden für Polynome der Ordnung 4 pro Segment $4 + 3 + 2 + 1 = 10$ Momente berechnet.

In einem ersten Schritt des Verfahrens werden nach einer neuen Kalibrierung des Stereokamerasystems die entsprechenden Silhouetten erzeugt und deren Pseudo-Zernike-Momente berechnet. Für jede neue Ausprägung einer Pose (bestimmt durch eine geänderte Rotation der virtuellen Hand) wird eine Vektor erzeugt, der folgende Parameter enthält:

- Alle Pseudo-Zernike-Momente beider Gestensegmente
- Verwendete Rotationswinkel des Handmodells
- Alle internen Gelenkrotationen des verwendeten Handmodells in der aktuellen Pose
- Verweis auf die gespeicherten Bilder der Silhouetten

Diese Vektoren bilden nun die Sätze einer Vergleichsdatenbank, in der zur Laufzeit des Systems der am besten passende Eintrag gesucht wird. Als Maß für die Suche werden alle Momente des Segmentpaares als Vektor aufgefasst. Demnach ist der Datensatz zu wählen, dessen euklidische Distanz zum analogen Vektor, der aus den beiden Kamerabildsegmenten berechnet wurde, minimal ist. Da davon ausgegangen werden kann, dass sich bereits das erste Moment eines Segmentpaares zwischen realer und virtuell erzeugter Silhouette zumindest stark ähnelt, werden die Datensätze nach der Erzeugung nach der Größe des ersten Moments sortiert und abgespeichert.
Nichtsdestotrotz ist für die Bestimmung des bestmöglichen Datensatzes zur Laufzeit der Erkennung nicht das erste Moment sondern die euklidische Distanz für die Lösungsfindung ausschlaggebend. Da nicht alle Einträge der Datenbank überprüft werden können, ohne den Echtzeitanspruch der intuitiven Interaktion zu gefährden, wird nach der Berechnung der Momente in einem neuen Kamerabildpaar zunächst der Datenbankeintrag, der dem ersten Moment am ähnlichsten ist, ermittelt. Von hier aus werden für eine vorgegebene Nachbarschaft (von beispielsweise 5000 Datensätzen) die euklidischen Abstände zwischen den Momentvektoren berechnet und somit lokal das bestpassende Segmentpaar ermittelt. Die diesem Datensatz zugeordneten internen Gelenkrotationen werden nun als erkannte Handpose angenommen und können zur Animation an die darstellende Anwendung übertragen werden.

7.4 Bestimmung der Handrotation

Das Rekonstruktionsergebnis der Momentenanalyse bildet gut die innere Skelettstruktur der Handpose wieder. Allerdings kann es bei der Rotation der Hand im Raum bei dem gewählten Verfahren zu Problemen kommen. Da die Pseudo-Zernike-Momente auch invariant bezüglich der Rotation sind, kommt es häufiger vor, dass die Orientierung der Hand im Raum deutlich von der tatsächlichen Rotation abweicht. Ein Beispiel, dass sich die gerenderten Silhouetten in ihren Form nur wenig voneinander unterscheiden und somit auch ähnliche Momente liefern ist in Abbildung 7.3 zu sehen. Durch die Verwendung eines Systems mit zwei Kameras können leicht Fälle auftreten, in denen Daumen und Zeigefinger in der zweiten Kamera verdeckt und somit in den Silhouetten nicht erkennbar sind. Dies kann zu einer deutlich von

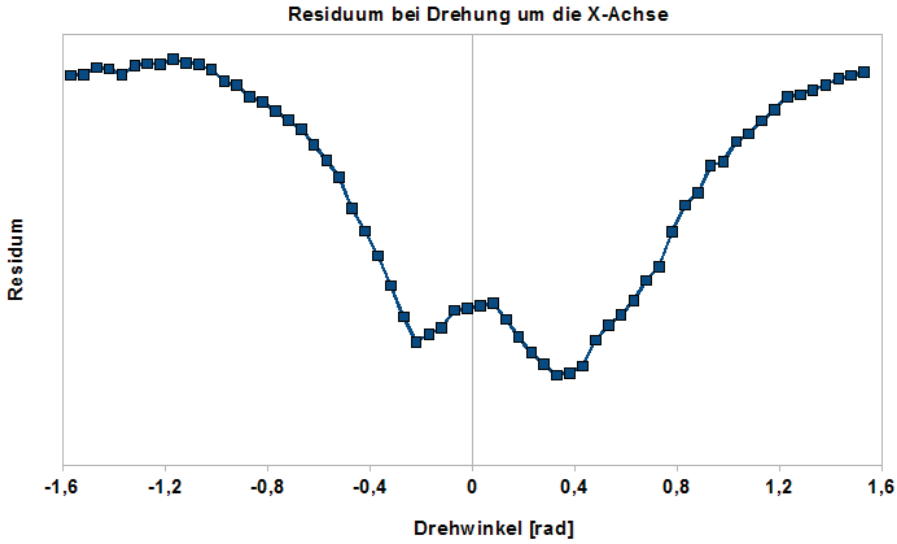

Abbildung 7.4: Residuumsfunktion bei Rotation einer statischen Pose um die x-Achse. Das Diagramm zeigt die Residuumswerte (y-Achse) für die Rotation des Handmodells um die x-Achse, während die anderen fünf Parameter fest bleiben.

der realen Pose abweichenden Rotation der Hand führen. Für ein präzises Zugreifen eines virtuellen Objektes ist aber offensichtlich die Position der Daumen- und Fingerkuppe und damit der exakten Orientierung der Hand im Raum notwendig. Prinzipiell kann die Verdeckungsproblematik durch die Verwendung eines Stereokamerasystems mit mehr als zwei Kameras gelöst werden. Allerdings erhöht sich bei der Verwendung von mehr als zwei Kameras auch der Rechenaufwand, verlangsamt das System damit deutlich und gefährdet somit den Echtzeitanspruch für die intuitive Interaktion. Aus diesem Grund wird nach ermittelter Pose durch die Momentenanalyse ein weiterer Schritt eingeführt, der mit bereits feststehenden internen Gelenkwinkel die Rotation der Hand im Raum bestimmt.

Die Anpassung der Orientierung wird mittels des bereits in Kapitel 4 vorgestellten nichtlinearen Optimierungsverfahrens *Simulated Annealing* durchgeführt. Dabei werden die realen und künstlich erzeugten Silhouetten der gefundenen Handpose auf Bildbasis miteinander verglichen. Die Verwendung von *Simulated Annealing* zur Rekonstruktion einer Pose durch Silhouettenvergleich konnte bereits in [MS07, Mal07] für die Bestimmung der Körperpose in monokularen Bildströmen erfolgreich nachgewiesen werden. Allerdings ist die ausschließliche Verwendung dieses Optimierungsverfahren zu rechenaufwendig, um verwertbare Ergebnisse in Echtzeit gewinnen zu können. In dem hier vorgestellten Verfahren ist durch den vorangegangenen Schritt der Momentenanalyse der Suchraum für die endgültige Pose bereits auf ein Minimum reduziert, so dass für die Anpassung nur noch wenige Optimierungsschritte notwendig sind.

Für die Bestimmung der Orientierung der Hand im Raum werden in jedem Optimierungsschritt die realen Handsilhouetten mit den künstlich erzeugten Silhouetten bei geänderter Position und Orientierung des gerenderten 3D-Modells als Differenzbild miteinander vergli-

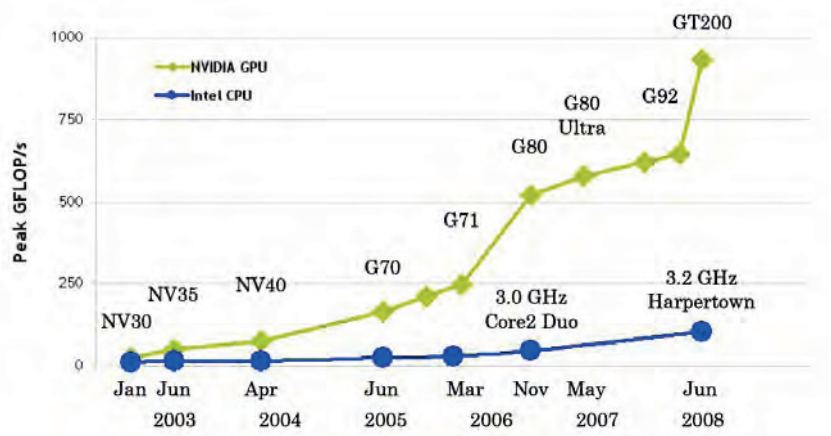

Abbildung 7.5: Vergleich der Leistungsentwicklung für CPU und GPU aus [Cor08].

chen. Die zu optimierenden Unbekannten der Zielfunktion E ergeben sich also als Vektor

$$x_i = (t_x, t_y, t_z, \alpha_x, \alpha_y, \alpha_z)^T. \tag{7.9}$$

aus der Translation und Rotation der Handpose im Raum. Dabei wird auch hier der Suchraum auf geeignete Intervalle reduziert. Abhängig von der Kamerakalibrierung kann beispielsweise ein Suchvolumen von $\pm 10cm$ ausgehend von der initialen Schätzung der Position der Hand im Raum gewählt werden. Um die Silhouetten miteinander vergleichen zu können, werden die gerenderten Silhouettenbilder des virtuellen Handmodells auf die gleiche Größe der realen Silhouetten skaliert. Als Änderungsschritte im Variablenvektor von x_{i-1} nach x_i wird zunächst zufällig eine der sechs Variablen des Vektors ausgewählt und dann eine zufällige Änderung dieses Wertes im vorgegebenen Intervall festgelegt. Somit ergibt sich für die unter x_i gerenderten Silhouetten beider Kameras das Residuum

$$E(x_i) = \sum_{Kamera} \sum_{x,y} |R_{x,y} - K_{x,y}|\,, \tag{7.10}$$

wobei $R_{x,y}$ ein Pixelwert der realen und $K_{x,y}$ ein Pixelwert der künstlichen Silhouette bezeichnet.
Für die Festlegung des Auskühlschemas wird die Residuumsfunktion für einige statische Posen untersucht. Dabei zeigt sich, dass die Zielfunktion im vorgegebenen Suchraum nur wenige lokale Minima aufweist (siehe Abbildung 7.4). Diese Tatsache erlaubt die Wahl eines schnellen Auskühlschema, das mit nur wenigen Optimierungsschritten bereits ein für die Interaktion verwendbares Ergebnis liefert.

7.5 Verfahrensbeschleunigung durch Parallelisierung

Wie bereits in Kapitel 4 gezeigt, ist die Verwendung von *Simulated Annealing* in der Objekterkennung sehr rechenintensiv und ermöglicht nur bedingt die Verwendung für ein interak-

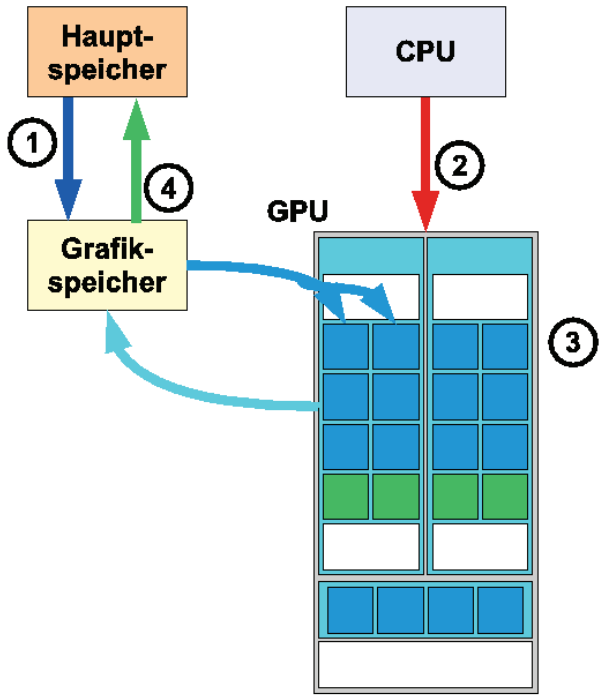

Abbildung 7.6: Arbeitsablauf einer Parallelverarbeitung mit CUDA.

tives System. Zudem verbraucht der Verfahrensschritt der Momentenanalyse bereits einen hohen Anteil der zur Verfügung stehenden Rechenkapazität. Um trotzdem zu einem System mit interaktiven Bildwiederholraten zu gelangen, müssen deshalb weitere Einsparungen in Bezug auf die geforderte Rechenleistung erreicht werden. In dem hier entwickelten Verfahren wird dies durch die Verwendung der Grafikprozessoren (GPU) zur Entlastung des Hauptprozessors (CPU) erreicht. Diese im Englischen *General Purpose Computation on Graphics Processing Unit* (GPGPU) genannte Technik erlaubt es, Berechnungen auf den Multiprozessoren der Grafikkarte parallel auszuführen, die nichts mit der eigentlichen Computergrafik zu tun haben. Neben der reinen Aufteilung der Rechenarbeit auf mehrere Prozessoren, macht man sich dabei außerdem zunutze, dass die Rechenleistung moderner Grafikkarten in den letzten Jahren deutlich schneller gewachsen ist, als die Rechenleistung der Hauptprozessoren (siehe Abbildung 7.5). Während beispielsweise ein mit 3,6 GHz getakteter Intel Pentium 4 Hauptprozessor mit vier Prozessorkernen eine Rechenleistung von 7,2 GFlops erreicht[1], kommt eine Nvidia GeForce 9800 GT Grafikkarte mit 112 Streamprozessoren auf eine theoretische Rechenleistung von bis zu 508 GFlops[2]. Eine Möglichkeit, die Grafikkarte mit in die Berechnung einzubeziehen, ist die vom amerikanischen Grafikkartenhersteller Nvidia entwickelte *Compute Unified Device Architecture* (CUDA) [Hal08].

CUDA bietet mit einem frei verfügbaren SDK [Cor08] die Möglichkeit, parallelisierbare Auf-

[1] Quelle: http://www.intel.com/support/processors/
[2] Quelle: http://de.wikipedia.org/wiki/Nvidia-Geforce-9-Serie

gaben auf der Grafikhardware durchzuführen. Dafür werden generell folgende Arbeitsschritte durchgeführt (siehe auch Abbildung 7.6):

1. Kopieren der Daten vom Hauptspeicher in den Speicher der Grafikkarte.
2. Übermittlung der Rechenanweisung.
3. Ausführung der Rechenanweisungen in parallelen Prozessen.
4. Kopieren der Ergebnisse von der Grafikkarte in den Hauptspeichers des Rechners.

Für das in diesem Kapitel vorgestellte Verfahren bieten sich neben der parallelen Berechnung der euklidischen Abstände der Momentvektoren insbesondere auch alle bildverarbeitenden Schritte wie Segmentierung, Berechnung von Differenzbildern, Bildskalierung und die Berechnung der Pseudo-Zernike-Momente an, da hier jedes Pixel in einem eigenen Prozess (Thread) verarbeitet werden kann. Für diejenigen bildverarbeitenden Prozesse, die im Anschluss an einen Rendering-Schritt des virtuellen Handmodells durchgeführt werden, entfällt sogar der Schritt des Kopierens der Bilddaten in den Speicher der Grafikkarte, da die entsprechenden CUDA-Anweisungen direkt die auf der Grafikkarte erzeugten Bilddaten verwenden können.

Tabelle 7.1: Vergleich der bildverarbeitenden Schritte zwischen CPU und GPU.

Schritt	CPU	GPU
Segmentierung	9 ms	0.8 ms
Differenzbild	8 ms	0.7 ms
Bildskalierung auf 1/4 der Originalgröße	14 ms	1.1 ms
Pseudo-Zernike-Momente der Ordnung 4	30 ms	2.1 ms

Tabelle 7.1 vergleicht die für das Verfahren notwendigen bildverarbeitenden Schritte zwischen CPU und GPU. Dargestellt ist jeweils der Mittelwert aus zwanzig Durchläufen bei unterschiedlichen Bildern der Größe 640*480, bzw. binären Segmenten mit einer Größe von ungefähr 100*100 Pixeln des umschließenden Rechtecks. Die Verarbeitungszeiten mittels CUDA wurden gemessen auf einer Nvidia GeForce 9800 GT Grafikkarte.

7.6 Ergebnisse

Das in diesem Kapitel entwickelte Verfahren wurde anhand der dynamischen Geste des Pinzettengriffs mit einer Diskretisierung in sechzehn statischen Einzelposen überprüft. Dazu wurde als Hardware ein Standard-PC mit einem Intel Core 2 Quad Prozessor mit 2,3 Ghz, 4 GB RAM und einer Nvidia GeForce 9800 GT Grafikkarte verwendet. Für die Erzeugung der vordefinierten Silhouetten- und Momente-Datenbank wurden folgende Werte ermittelt: OpenSG erreicht beim Rendern des im Abschnitt 7.1 vorgestellten virtuellen Modells der Hand mit knapp 1600 Polygonen eine Wiederholrate von bis zu 200 Bildern pro Sekunde. Für die Erzeugung von 57760 Silhouetten pro Kamera benötigt das System nach der Kalibrierung der Kameras demnach ungefähr zehn Minuten. Durch die benötigte Zeit für

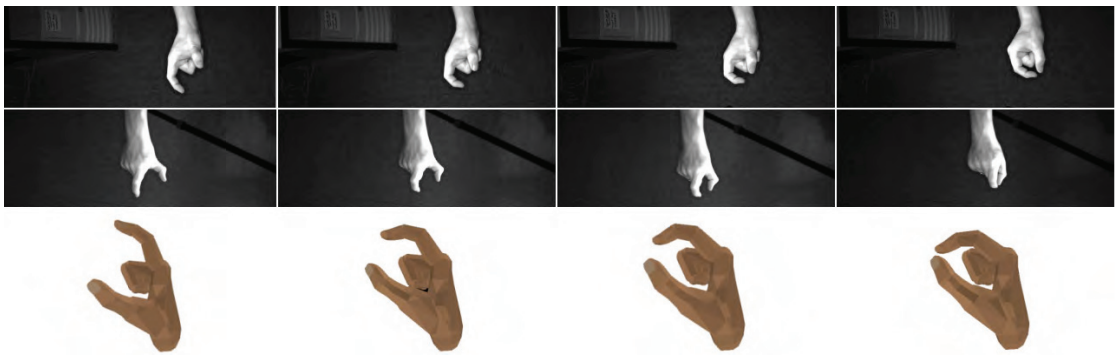

Abbildung 7.7: Pinzettengriff in vier verschiedenen Posen. Die oberen Reihen zeigen Ausschnitte der Kamerabilder, die untere Reihe die ermittelten Handposen.

die Berechnung der Pseudo-Zernike-Momente (insgesamt ca. vier Minuten) und durch die notwendige Sortierung der Vektoren nach ihrem ersten Moment ergibt ein reale Zeit für die Erzeugung der Vergleichsdatenbank von bis zu zwanzig Minuten, die nach einer neuen Kalibrierung benötigt werden, bis das System für die Erkennung bereit ist.
Das System erreicht zwar auf der gewählten Hardware nicht die in der Einleitung dieser Arbeit genannte Geschwindigkeit von 20 Bilder pro Sekunde, die für eine Echtzeitanwendung gefordert werden, erreicht aber mit durchschnittlich 13 Bildern pro Sekunde eine Bildwiederholrate, die bereits für eine interaktive Anwendung geeignet ist. Zudem ist davon auszugehen, dass durch den Einsatz von CUDA zur Verfahrensbeschleunigung bereits die Verwendung einer leistungsstärkeren Grafikkarte zu deutlich kürzeren Verarbeitungszeiten und damit höheren Bildwiederholraten führt.
Eine Bestimmung der korrekten Erkennungsrate ist für das vorgestellte Verfahren praktisch nicht möglich, da es kein Verfahren gibt, das die tatsächlichen Gelenkwinkel der menschlichen Hand anhand von Videobildern bestimmen kann. Die Verwendung von zusätzlicher Hardware wie beispielsweise einem Datenhandschuh liefert zwar präzise Gelenkwinkel, um die reale Pose der Hand zu bestimmen, ändert aber auch das Erscheinungsbild der Hand in den Kamerabildern und führen so zu abweichenden Silhouetten. Auch die Verwendung von künstlich erzeugten virtuellen Silhouetten zum Testen des Verfahren schließt sich aus, da in diesem Fall weder Kalibrierungsfehler noch Unterschiede in den Handgeometrien oder Störungen der Kameras wie beispielsweise Rauschen berücksichtigt werden. Aus diesem Grund wurde zur Bestimmung der Erkennungsrate folgendes Verfahren angewendet: Anstelle eines Live-Videostroms des Kamerasystems wurde die Erkennung anhand von zuvor aufgenommenen Videosequenzen überprüft. Die Videosequenzen zeigen das Zugreifen mittels des Pinzettengriffs. Bei der Aufnahme der Sequenzen wurde insbesondere darauf geachtet, dass sich für alle aufeinanderfolgenden Bilder der Sequenz Daumen und Zeigefinger weiter schließen. Bei einem nicht-systematischen Fehler des Verfahren muss davon ausgegangen werden, dass bei der Hälfte der falschen Entscheidungen Daumen und Zeigefinger zu stark geschlossen und bei der anderen Hälfte der Fehlentscheidungen zu weit geöffnet erkannt werden. Als Kriterium für den Grad der Öffnung zwischen Daumen und Zeigefinger kann der Abstand zwischen

Daumen- und Fingerkuppe verwendet werden. Da für die aufgenommenen Sequenzen bekannt ist, dass sich in jedem aufeinanderfolgenden Bildpaar Daumen und Zeigefinger weiter schließen, beschreibt die Anzahl der erkannten Fehlrekonstruktionen, bei denen sich der Pinzettengriff weiter öffnet, statistisch die Hälfte der tatsächlichen Fehlinterpretationen. Für den Test wurden drei verschiedene Sequenzen mit insgesamt 240 Bildpaaren einer Kalibrierung verwendet. Mit diesem Test ergibt sich bei 34 fehlerhaft erkannten Bildpaaren (7%) für das Verfahren somit eine Erkennungsrate von 86%. Im Vergleich mit den drei anderen in dieser Arbeit entwickelten Algorithmen bleibt das Verfahren der Momentenanalyse in Bezug auf die Verwendbarkeit für eine intuitive Interaktion noch zurück. Zum einen erfüllt das Verfahren die in der Einleitung dieser Arbeit aufgestellte Bedingung nach der Echtzeitfähigkeit mit einer Wiederholrate von mindestens 20 Bildern pro Sekunde noch nicht. Zum anderen wird ein uneingeschränkter Einsatz für eine intuitive Interaktion dadurch erschwert, dass das Verfahren deutlich anfälliger gegenüber Änderungen der Umgebungssituation ist. Insbesondere eine stärkere Änderung der Lichtverhältnisse kann eine Über- oder Untersegmentierung der Handpose verursachen. Ändert sich aber das Erscheinungsbild der Silhouette, sind die daraus resultierenden Momente nicht mehr mit den Momenten der künstlich erzeugten Silhouetten vergleichbar. Dies führt dann zwangsläufig dazu, dass eine Rekonstruktion der Geste versagt. Bei der Verwendung des Verfahrens muss deshalb eine konstante Beleuchtungssituation der Umgebung sichergestellt sein.

7.7 Zusammenfassung

In diesem Kapitel wurde ein Verfahren entwickelt, dass den dynamischen Prozess einer Handgeste rekonstruiert und so eine einfache und intuitive Interaktion beispielsweise beim Zugreifen und Loslassen eines virtuellen dreidimensionalen Objektes mit dem Pinzettengriff ermöglicht. Das Verfahren basiert auf der Analyse der Pseudo-Zernike-Momente von Handsilhouetten. Durch die Diskretisierung des dynamischen Prozesses in eine vorgegebene Anzahl von statischen Einzelposen wird der Suchraum für das Verfahren deutlich eingeschränkt. Die Momente der realen Handsilhouetten werden mit einer vorab erzeugten Tabelle aus künstlich erzeugten Handsilhouetten verglichen. Um die Invarianz der Pseudo-Zernike-Momente gegenüber der Rotation auszugleichen, wird nach der Bestimmung der internen Gelenkwinkel der Hand ein nichtlinearer Optimierungsschritt angewendet, der auf Bildbasis reale Silhouettenbilder mit künstlich erzeugten Silhouettenbildern vergleicht. Die Differenz zweier Silhouetten bestimmt dabei das Residuum der zu minimierenden Zielfunktion. Zur Beschleunigung des Verfahrens wird ein Großteil der Berechnung parallelisiert auf der Grafikhardware ausgeführt. Das Verfahren erfüllt zwar nicht die der Einleitung dieser Arbeit aufgestellte Anforderung an die Echtzeitfähigkeit, erreicht mit durchschnittlich 13 Bildern pro Sekunde eine Bildwiederholrate, die bereits für eine interaktive Anwendung geeignet ist.

Kapitel 8

Ergebnisse

Nachdem in den vorigen Kapiteln vier neue Verfahren zur intuitiven Interaktion durch videobasierte Gestenerkennung entwickelt und untersucht wurde, sollen die Verfahren nun im Folgenden verglichen und bewertet werden. Neben einer Gegenüberstellung der Verfahren in Hinblick auf die technischen Randbedingungen und der Überprüfung der für intuitive Interaktion geforderten Bedingungen der Algorithmen, wird auch die Verwendbarkeit der verschiedenen Verfahren in Bezug auf mögliche Anwendungen beschrieben, indem gezeigt wird, welche Handgesten für verschiedenen Anforderungen an die intuitive Interaktion geeignet sind und dadurch die Auswahl des zu verwendenden Verfahrens bestimmt werden kann. Der zweite Teil dieses Kapitels untersucht in einer Usability-Studie [MSS07] explizit die Verwendbarkeit des Verfahrens der *Interaktion durch Punktprojektion* und evaluiert dessen Akzeptanz durch eine Befragung von mehr als 80 Anwendern. Um die Bedienbarkeit einer Anwendung und damit die Verwendbarkeit des Verfahrens für eine intuitive Interaktion zu überprüfen, wird einem Nutzer des Systems eine einfache Aufgabe gestellt, die ohne weitere Erklärungen gelöst werden muss und überprüft, ob er die Aufgabe lösen kann. Am Anschluss wird auch das subjektive Empfinden durch eine Bewertung der Lösung der Aufgabe durch den Anwender überprüft.

8.1 Vergleich und Bewertung der Verfahren

Im einleitenden Kapitel dieser Arbeit wurden drei Hauptanforderungen an Verfahren aufgestellt, die erfüllt werden müssen, damit eine Interaktion zwischen Mensch und Computer einfach und intuitiv stattfinden kann: Zum einen dürfen die Verfahren keine direkten technischen Hilfsmittel fordern, die der Anwender tragen oder in der Hand halten muss. Zum anderen müssen die Verfahren ohne oder zumindest nur mit minimalem Trainingsaufwand verwendbar sein, damit der Anwender in die Lage versetzt wird, ohne Verzögerung mit der Anwendung zu interagieren. Außerdem müssen die Verfahren echtzeitfähig sein, also sowohl mit einer Wiederholrate von mindestens 20 Bildern pro Sekunde als auch mit einer Verzögerung von weniger als 200 Millisekunden arbeiten. Die folgende Tabelle 8.1 stellt die vier entwickelten Verfahren in einem direkten Vergleich in Bezug auf diese drei Forderungen gegenüber.

Tabelle 8.1: Vergleich der Verfahren in Bezug auf die aufgestellten Forderungen an Verfahren zur intuitiven Interaktion.

	Interaktion durch Aktive Formen	**Interaktion durch Punktprojektion**	**Interaktion durch Merkmalbasierte Klassifikation**	**Interaktion durch Momenten-Analyse**
Technische Hilfsmittel	keine	keine	keine	keine
Trainingsaufwand	keiner	keiner	2 min bei individueller Modellbildung[1]	keiner
Echtzeitfähigkeit	ja (20 fps)	ja (30 fps)	ja (30 fps)	nein (13 fps)[2]

[1] Eine individuelle Modellbildung durch ein personenbezogenes Training ist zwar nicht erforderlich, wenn ein bereits vorab erstelltes Modell aus Trainingsdaten von mehreren Anwendern verwendet wird, erhöht aber die Erkennungsrate des Systems (vergleiche Kapitel 6).

[2] Es ist aber zu erwarten, dass allein durch die Entwicklung der Hardware auf dem Gebiet der Grafikprozessoren bereits nach einem Jahr die geforderten 20 Bilder pro Sekunde erreicht werden (vergleiche Kapitel 7).

Intuitive Interaktion zwischen Mensch und Computer sollte immer benutzerorientiert und damit auch anwendungsbezogen sein. Ziel für eine einfache und intuitive Bedienbarkeit eines Computersystems muss es deshalb sein, dass der Anwender sich nicht auf die Eingabemodalität und deren Technik konzentrieren muss, sondern dass er sich direkt die Aufgaben der Anwendung bearbeiten kann. Die bedeutet insbesondere auch, dass die Wahl des Verfahrens nicht durch technische Restriktionen oder durch die Umgebungsbedingungen getroffen werden darf, sondern dass aus der Anwendung selbst die Anforderungen an das Verfahren zur Gestenerkennung abgeleitet werden müssen. Aus diesem Grund werden im Folgenden die Besonderheiten der einzelnen Verfahren zusammengefasst und typische Aufgaben aufgezeigt, für die die einzelnen Verfahren besonders geeignet sind.

- Interaktion durch Rekonstruktion von Aktive Formen

  Das Verfahren ist in der Lage, eine vom System vorgegebene Zeigegeste zu erkennen und zu verfolgen. Die Tatsache, dass sich der Anwender strikt an die trainierte Geste halten muss, kann sich eine Anwendung zu nutze machen, um explizit die Zeigegeste von anderen Gesten, die ein Anwender nicht benutzen soll, abzugrenzen.

- Interaktion durch Punktprojektion

  Das Verfahren erkennt und verfolgt beliebige individuelle Interpretationen einer Zeigegeste. Daher ist das Verfahren immer dann besonders gut einsetzbar, wenn eine

Vielzahl unterschiedlicher Nutzer das System ohne Lernphase und ohne vorherige Erklärung mit der Interaktion beginnen sollen.

- Interaktion durch Merkmalbasierte Gesten-Klassifikation

  Das Verfahren ist in der Lage, mehrere unterschiedliche Handposen in Echtzeit voneinander zu trennen und die Position der Hand im Raum zu bestimmen. Damit ist dieses Verfahren insbesondere für eine Interaktion mit virtuellen dreidimensionalen Welten geeignet, in denen beispielsweise der Anwender Objekte im Virtuellen bewegen kann.

- Interaktion durch Momentenanalyse

  Das Verfahren ist in der Lage, den dynamischen Prozess im kinematischen Modell einer Geste nachzubilden. Damit ist dieses Verfahren besonders für Anwendungen geeignet, in denen nicht nur die Position der Hand im Raum bestimmt werden muss, sondern auch Aussagen über entweder die internen Gelenkwinkel des Handskeletts oder über die Position der Fingerspitzen und der Daumenspitze getroffen werden müssen.

Die folgende Tabelle 8.2 stellt die für die Auswahl des geeigneten Verfahrens notwendigen Parameter für alle vier Verfahren gegenüber und ermöglicht so den direkten Vergleich in Hinblick auf Art und Anzahl der detektierbaren Gesten sowie auf die Besonderheiten der Verfahren.

Tabelle 8.2: Vergleich der Verfahren in Hinblick auf die detektierbaren Gesten.

	Interaktion durch Aktive Formen	**Interaktion durch Punktprojektion**	**Interaktion durch Merkmalbasierte Klassifikation**	**Interaktion durch Momenten-Analyse**
Gestenart	statisch	statisch	statisch	dynamisch
Anzahl Gesten	1	1	≥ 3	1
Getestete Gesten	Zeigegeste	Zeigegeste	Zeigegeste, offene und geschlossene Hand	Greifbewegung durch Pinzettengriff
Besonderheiten	Art der Geste wird vom System vorgegeben	Individuelle Interpretationen der Geste möglich	Individuelle Gesten durch Trainingsphase möglich	Art der Geste wird vom System vorgegeben

Im Folgenden sollen noch die Einsatzmöglichkeiten der Verfahren für unterschiedliche Aufgabenstellungen von möglichen Anwendungen betrachtet werden, für die eine videoba-

sierte Gestenerkennung eingesetzt werden kann. Da oft mehr als nur eins der Verfahren technisch in der Lage ist, zur Steuerung einer Anwendung eingesetzt zu werden, soll offensichtlich das Verfahren tatsächlich zum Einsatz kommen, das aufgrund seiner Besonderheiten eine intuitive Interaktion am besten unterstützt.

- Steuerung eine Anwendung durch Bedienung einer grafischen Benutzerschnittstelle

  Programme, die über eine klassische grafische Benutzerschnittstelle mit Schaltflächen, Menüs und ähnlichen zweidimensionalen Elementen bedient werden, verwenden grundsätzlich einen Cursor als visuelle Rückantwort des Systems. Dieser Cursor wird dabei kontinuierlich in Echtzeit über den Ausgabebildschirm bewegt. Die zweidimensionale Bewegung dieses Zeigers kann typischerweise durch die Verwendung einer Zeigegeste realisiert werden. Damit bietet sich in diesem Fall sowohl das Verfahren der *Interaktion durch Rekonstruktion von Aktiven Formen* als auch das Verfahren der *Interaktion durch Punktprojektion* an. Die Entscheidung, welches der beiden Verfahren zu Einsatz kommen soll, muss über die Umgebungsbedingungen der Anwendung entschiedenen werden, insbesondere, indem untersucht wird, wie das System reagieren soll, wenn nicht explizit die Zeigegeste durch den Anwender benutzt wird.

- Navigation in einer dreidimensionalen Welt

  Die Bewegung eines Anwenders innerhalb einer dreidimensionalen Welt ist eine klassische Aufgabe für Anwendungen der Virtuellen Realität. Bei einer Navigation, die typischerweise mittels technischer Hilfsmittel wie einer Computermaus oder aber auch Datenhandschuhen realisiert wird, stehen klassischerweise zwei unterschiedliche Ansätze zur Verfügung. Zum einen kann die Navigation durch sogenannte Billboards, also planare virtuelle Objekte, die immer an der gleichen Position des Bildschirms dem Anwender zugewandt dargestellt werden, realisiert werden. Billboards, die zur Navigation verwendet werden, zeigen in der Regel Symbole, die dem Anwender die Art der Navigation vorgeben. Typischerweise werden hier texturierte Flächen verwendet, auf denen Pfeile die Richtung der Navigation in der 3D-Welt anzeigen. Für eine Navigation mittels Gestenerkennung bieten sich hier erneut die Verfahren der *Interaktion durch Rekonstruktion von Aktiven Formen* und der *Interaktion durch Punktprojektion* an, da diese interaktiven Flächen einfach durch eine Zeigegeste bedient werden können. Zum anderen werden oft Datenhandschuhe zur Navigation eingesetzt. Die 3D-Position des Handschuhs und damit der Hand des Anwenders bestimmt die Navigationsrichtung. Bewegt der Anwender beispielsweise die Hand nach rechts, steuert er auch in der virtuellen Welt in die entsprechende Richtung. Eine Bewegung der Hand nach vorne oder nach hinten beschleunigt oder verlangsamt die Navigation. Eine Unterscheidung der Geste entscheidet dabei grundsätzlich, ob der Anwender sich in der Welt bewegen möchte oder nicht. Analog zum Datenhandschuh kann hier das Verfahren der *Interaktion durch Merkmalbasierte Klassifikation* verwendet werden, um eine Navigation ohne technische Hilfsmittel zu realisieren.

- Neupositionierung von virtuellen Objekten in einer dreidimensionalen Welt

  Dreidimensionale Objekte im Raum zu verschieben und zu bewegen ist eine oft auf-

Abbildung 8.1: Demografische Verteilung der Testanwender (aus [MSS07]).

tretende Aufgabe in Anwendungen der Virtuellen Realität. Für eine intuitive Lösung dieser Aufgabenstellung ist eine Unterscheidung von Handgesten erforderlich, damit ein Zugreifen und Loslassen des Objekts realisiert werden kann. Hier können die Verfahren der *Interaktion durch Merkmalbasierte Klassifikation* und der *Interaktion durch Momentenanalyse* eingesetzt werden, um eine gestenbasierte Inetraktion zu ermöglichen.

- Untersuchung von virtuellen Objekten in einer dreidimensionalen Welt

  Ein typisches Merkmal der Untersuchung von dreidimensionalen Objekten ist die Tatsache, dass der Anwender das Objekt in die Hand nehmen und durch Rotation der Hand von unterschiedlichen Blickwinkeln betrachten kann. Zunächst ist hier erneut eine Unterscheidung von Handgesten notwendig, um zu detektieren, wann das Objekt aufgenommen bzw. wieder abgelegt wird. Da während der Betrachtung des Objekts selbst neben der Position auch die Orientierung der Hand ermittelt werden muss, kann das Verfahren der *Interaktion durch Momentenanalyse* eingesetzt werden, um diese Aufgabe ohne technische Hilfsmittel zu lösen.

8.2 Usability-Studie

Im folgenden Abschnitt wird anhand einer Usability-Studie gezeigt, dass das Verfahren der *Interaktion durch Punktprojektion* für eine intuitive Interaktion verwendbar ist. Dabei wird zum einen die Lösung einer dem Anwender gestellten Aufgabe untersucht, um anhand eines objektiven Kriteriums die Bedienbarkeit der Anwendung mittels Gestenerkennung auswerten zu können. Zum anderen wird über eine Nutzerbefragung das subjektive Empfinden des Anwenders bei einer Interaktion ohne technische Hilfsmittel ermittelt und ausgewertet.

In der hier vorgestellte Usability-Studie [MSS07] konnte das Verfahren von insgesamt 81 Anwendern getestet werden. Um die Ergebnisse der Studie vergleichbar zu machen, wurden jedem neuen Anwender zunächst einige demografische Fragen gestellt. Neben Angaben über das Alter und Geschlecht, die Ausbildung und den Beruf (siehe Abbildung 8.1) wurde insbe-

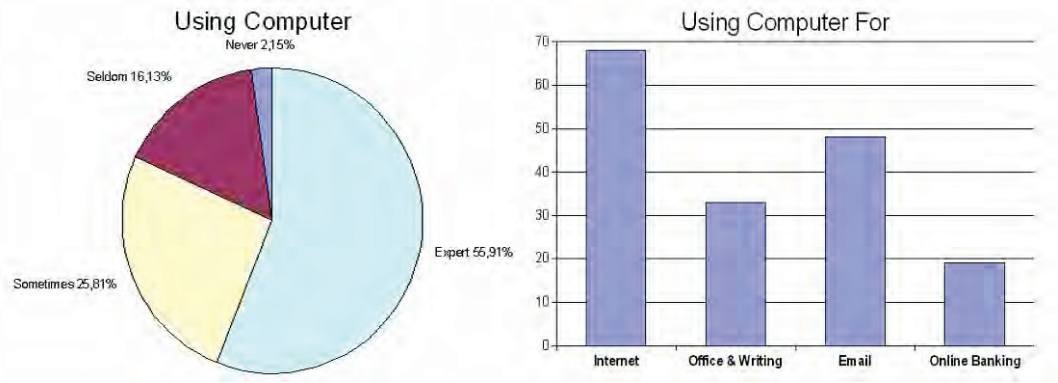

Abbildung 8.2: Bisherige Verwendung eines Computers der Testnutzer (aus [MSS07]).

sondere auch der tägliche Umgang mit einem Computer abgefragt. Neben den Fragen, ob und für welche Zwecke der Computer normalerweise eingesetzt wird (siehe Abbildung 8.2), wurde auch nach bereits verwendeten Eingabemodalitäten gefragt, um sicherzustellen, dass der Anwender noch nicht durch eine gerätefreie Eingabe mit einem Computer vorbelastet war. Die Evaluierung wurde an zwei unterschiedlichen Orten durchgeführt. Zum einen wurde zum Testen ein Laboraufbau verwendet, zum anderen konnte das System in einem Museum und damit an einem öffentlichen Ort aufgebaut und getestet werden. Durch diese Tatsache weicht zwar die demografische Verteilung der Studie von der durchschnittlichen Verteilung in der Bevölkerung ab, dennoch stehen ausreichend Informationen zur Verfügung, um auch Ergebnisse von technisch unversierten Anwendern zu erhalten, die nie oder nur selten mit einem Computer arbeiten. Wie in Abbildung 8.1, links zu sehen ist, waren zwei Drittel der Anwender Männer. Der hohe Anteil an Jugendlichen in der Altersklasse zwischen 14 und 21 Jahren ist damit zu erklären, dass im Museum auch Schulklassen für die Teilnahme an der Evaluierung gewonnen werden konnten.
Mit 56% bezeichneten sich die meisten Anwender als Computerexperten, die einen Computer täglich oder zumindest mehrmals in der Woche benutzen. Ein Viertel (26%) gab an, den Computer hin und wieder zu benutzen, während 16% einen Rechner nur selten verwenden. Zwei Nutzer (2%) gaben an, keine Erfahrungen im Umgang mit einem Computer zu haben. Zudem wurden die Anwender gefragt, für welche Aufgaben sie den Computer normalerweise einsetzen. Es ist nicht überraschend, dass die meisten Computernutzer im Internet surfen und viele den Computer nutzen, um Emails zu schreiben. Ungefähr die Hälfte der Computernutzer ist vertraut mit Büroanwendungen, um beispielsweise Briefe am Rechner zu schreiben. Ein Drittel verwendet den Computer für Online-Bankgeschäfte (siehe Abbildung 8.2).
Als zur Evaluierung des Verfahrens verwendete Anwendung wurde dem Ort des Tests entsprechend eine Anwendung aus dem musealen Kontext gewählt. In dieser Anwendung ist der Museumsbesucher in der Lage, mittels einer Zeigegeste eine virtuelle Lupe über ein digitalisiertes Bild wandern zu lassen, um Details des Bildes hervorzuheben (siehe auch Abbildung 9.1). Die Anwendung der Untersuchung des *Heuwagen-Triptychons* von Hieronymus Bosch wird im folgenden Kapitel 9 näher beschrieben.

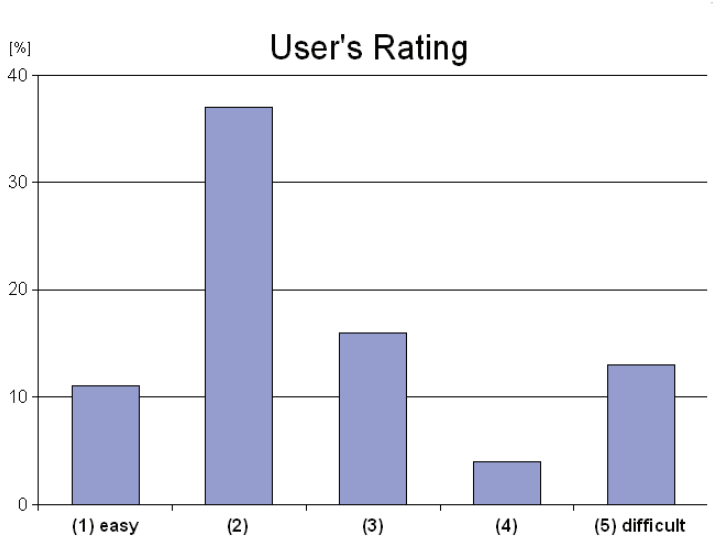

Abbildung 8.3: Ergebnis der Anwenderbefragung (aus [MSS07]). Jeder Anwender konnte die Interaktion mittels Zeigegeste mit einer Note zwischen 1 und 5 bewerten.

Für eine korrekte Evaluierung der intuitiven Interaktion ist ist wichtig, dass der Anwender die Zeigegeste ohne vorige Erklärung benutzt und dass ausgeschlossen werden kann, dass er vor der eigenen Anwendung bei einem anderen Nutzer zusehen kann, wie das System auf eine Geste reagiert. Aus diesem Grund wurden zu Beginn des Tests keine ausführlichen Erklärungen verwendet. Vor dem Stellen der Aufgabe wurde lediglich der Hinweis gegeben, dass sich das System allein durch Zeigen auf den Bildschirm steuern lässt. Die Aufgabe, die jeder Anwender zu erfüllen hatten, war denkbar einfach: Das auf dem Bildschirm dargestellte Gemälde zeigt in der Bildmitte unübersehbar einen Heuwagen. Die Aufforderung an den Nutzer war "*Bitte zeigen Sie mir ein Rad des Heuwagens!*". Gemessen wurde dann die Zeit, die verstreicht, bis der Anwender die Lupe über einem der beiden Räder positioniert hatte. Mehr als 92% der Anwender waren in der Lage, ohne Zeitverzögerung innerhalb der ersten fünf Sekunden die virtuelle Lupe an der entsprechende Stelle des Ausgabebildschirms zu positionieren. Lediglich 8% hatten Probleme, die Lupe umgehend mittels einer Zeigegeste zu bewegen. Der häufigste Grund dafür war, dass der Anwender nicht direkt auf den Bildschirm zeigte, sondern am Bildschirm vorbei nach rechts oder links deutete, um die dargestellte Lupe mit der Geste als Anweisung für den Computer nach rechts oder links bewegen zu lassen. Dieses Verhalten ist ein gutes Beispiel für unser von Computermaus und Tastatur geprägtes Denken, bei dem der Computer eine Anweisung nicht durch direkte Interaktion, sondern durch einen externen Befehl erhält. Nichtsdestotrotz waren bis auf zwei Anwender alle Nutzer in der Lage, die Aufgabe zu erfüllen, nachdem noch einmal explizit darauf hingewiesen wurde, dass die Lupe sich immer an genau der Stelle auf dem Bildschirm befindet, auf die der Anwender deutet. Neben dieser durch einen Experten kontrollierten, objektiven Überprüfung, ob die Aufgabe gelöst werden konnte, wurde danach auch das subjektive Empfinden des Anwenders bezüglich der Eingabemodalität abgefragt. In einem Fragebogen

konnte der Anwender das Interaktionsverhalten mit einer Note zwischen 1 für *"einfach"* bis 5 für *"schwierig"* bewerten. Insgesamt bewerteten vier von fünf Anwendern (79%) die Interaktion mit den Noten 1 (*"einfach"*) bis 3 (*"neutral"*). Mehr als jeder dritte Anwender vergab die Note 2, während nur jeder Achte die Interaktion mit *"schwierig"* bewertete (siehe Abbildung 8.3).

8.3 Zusammenfassung

In diesem Kapitel wurden die in dieser Arbeit entwickelten vier Verfahren zur intuitiven Interaktion durch videobasierte Gestenerkennung miteinander verglichen und bewertet. Neben einer Gegenüberstellung der verwendbaren Gesten und der technischen Restriktionen der einzelnen Verfahren, wurden auch die Möglichkeiten betrachtet, wie für eine vorhandene Anwendung aufgrund der Vorgaben für die Steuerung der Anwendung das entsprechende Verfahren, das eingesetzt werden sollte, um eine intuitive Interaktion durch videobasierte Gestenerkennung zu realisieren, ausgewählt werden kann. Aufgrund dieser Ergebnisse werden nun im folgenden Kapitel Anwendungen beschrieben, in denen die Verwendbarkeit der Verfahren im praktischen Einsatz nachgewiesen werden konnte.

Kapitel 9

Anwendungen

In diesem Kapitel werden einige Anwendungen beschrieben, in denen die in dieser Arbeit beschriebenen Verfahren zur videobasierten Gestenerkennung Verwendung gefunden haben. Im Laufe der letzten Jahre sind über dreißig verschiedene Anwendungen entstanden, die zum Teil ausschließlich als Demonstratoren unter Laborbedingungen gezeigt wurden, zum Teil aber auch einer breiten Öffentlichkeit vorgestellt werden konnten. Sei es beispielsweise einem Fachpublikum für einen kurzen Zeitraum von einer Woche als Teil eines Kunstwerk des russisch-deutschen Bildhauers Sergej Dott[1] auf der Hannover Messe 2005 auf dem Messestand der Fraunhofer-Gesellschaft[2] oder für insgesamt über ein Jahr während der Sonderausstellung Computer.Medizin[3] des Heinz-Nixdorf-MuseumsForum[4] in Paderborn wie in Abschnitt 9.4 beschrieben.

9.1 Anwendungsentwicklung

Allen Anwendungen gemeinsam ist die Tatsache, dass die Anwendungsentwicklung selbst auf einer strikten Trennung zwischen Eingabemodalitäten und der eigentlichen Inhaltsausführung beruht. Damit ist gewährleistet, dass die Realisierung einer Anwendung in allen Phasen des Entwicklungszyklus auch ohne die Verwendung des Gestenerkennungssystems durchführbar ist und zu jeder Zeit die Gestenerkennung durch beispielsweise eine Standard-Maus ersetzt werden kann. Multimodale Anwendungen, die Gesten- und Spracherkennung miteinander verbinden, können jederzeit durch die Verwendung von Maus und Tastatur auf korrektes Verhalten überprüft werden. Für die daraus resultierende, notwendige Kommunikation zwischen Eingabemodalität und Rendering stehen zwei unterschiedliche Möglichkeiten zur Verfügung. Zum einen können für Anwendungen, die ausschließlich eine Zeigegeste verwenden, Mausereignisse direkt auf Betriebssystemebene erzeugt werden. Damit wird es möglich, beliebige Anwendungen zu steuern, die mit der 2D-Cursor-Position und einem einfachen Mausklick bedient werden können. So ist beispielsweise auch die Steuerung

[1] http://www.sergejdott.de/
[2] http://www.fraunhofer.de/presse/presseinformationen/2005/04/Presseinformation08042005.jsp
[3] http://www.computer-medizin.de/
[4] http://www.hnf.de

von Präsentationen oder die Ausführung von Flash-basierten Webinhalten mittels Gestenerkennung möglich. Zum anderen können Ereignisse der Gestenerkennung aber auch zur Bedienung von 3D-Anwendungen genutzt werden, indem Gestenerkennung und Rendering-Software über eine Netzwerkschnittstelle miteinander kommunizieren. Als Standard für die Anbindung von Peripheriegeräten an 3D-Anwendungsprogramme hat sich heute das *Virtual Reality Peripheral Network* (VRPN) [THS+01] durchgesetzt, das auch von den meisten kommerziellen Tracking-Systemen unterstützt wird. Die im Folgenden beschriebenen Anwendungen verwenden das auf dem Szenengraphsystem OpenSG [RVB02] basierende instant**reality**-Rahmenwerk [BDR04]. instant**reality** ist ein flexibles und offenes Framework speziell für Echtzeit-Mixed-Reality-Systeme. Das Basis-Renderingmodul instantPlayer unterstützt dabei VRML- (Virtual Reality Modeling Language) und X3D-Anwendungen für Echtzeitvisualisierungen auf unterschiedlichsten Ausgabegeräten von Apples iPhones bis hin zu großen Projektionswänden. Dei Verwendung von X3D/VRML als Programmiersprache hat gegenüber proprietären Sprachen entscheidende Vorteile für die Anwendungsentwicklung:

- Für einen effizienten Entwicklungszyklus ist es wichtig, umfangreiche Entwicklungswerkzeuge zur Modellierung von statischen und dynamischen Szenen zur Verfügung zu haben. Für X3D/VRML-Szenen stehen dem Anwendungsentwickler eine große Anzahl unterschiedlichster Werkzeuge zum Modellierung, Optimierung und Konvertierung zur Verfügung.

- Die Schnittstellen sind durch einen Industrie-unabhängigen ISO-Standard definiert[5].

- Durch die vorgegebene Plattformunabhängigkeit ist die Entwicklung und das Testen der Anwendungen auf unterschiedlichsten Systemen gewährleistet und kann auf Standard-PCs durchgeführt werden.

- VRML, X3D und JavaScript sind sehr viel einfacher zu erlernen als "Low Level"-Schnittstellen, die von einigen VR-Plattformen bereit gestellt werden.

- Es sind eine Vielzahl von Büchern, Anleitungen und bereits entwickelten Beispielen verfügbar.

9.2 Untersuchung digitalisierter Gemälde

Die Untersuchung digitalisierter Gemälde an einem Computersystem ist prädestiniert für die Verwendung einer Zeigegeste als Eingabemodalität für die Interaktion. Insbesondere bei der Verwendung eines solchen digitalen Museumsexponats in der für Gemälde ursprünglichen Umgebung eines Museums liegen vier Vorteile auf der Hand:

- Digitale Kopien von Original-Leinwänden sind durch die heute verwendete digitale Fototechnik oft sehr hochauflösend und benötigen daher auch eine entsprechend große Darstellungsfläche zur Visualisierung.

[5] http://www.web3d.org/x3d/specifications/

Abbildung 9.1: Interaktive Erkundung des Heuwagen-Triptychons von Hieronymus Bosch mittels einer virtuellen Lupe, die durch eine Zeigegeste bewegt werden kann[MDS05a].

- In einem Museum ist es nicht nur verboten, das Originalgemälde zu berühren, oft wird auch ein zu dichtes Annähern an die Leinwand durch entsprechende Absperrungen verhindert. Zwar ist die direkte Interaktion mit dem virtuellen Exponat erwünscht, der durch das Zeigegesten-Erkennungssystem erforderliche Abstand zur Darstellung dennoch wünschenswert, um die Unterschiede zwischen Originalgemälde und virtuellem Exponat zu verkleinern.
- Insbesondere in Museen ist der Anteil technisch unversierter Menschen, die einen Computer nicht bedienen können oder möchten besonders hoch. Hier kann die videobasierte Gestenerkennung Abhilfe schaffen, da es trotz der Interaktion mit einem technischen System keine sichtbaren technischen Geräte gibt. Die zur Gestenverfolgung notwendigen Kameras sind weit über dem Nutzer und somit außerhalb des direkten Sichtfeldes angebracht und der Ausgabebildschirm ist das einzige technische Gerät, dass für den Anwender offensichtlich ist. Ein solcher Aufbau erleichtert die Hemmschwelle mit dem Exponat zu interagieren.
- Die Darstellung von zweidimensionalem Inhalt ermöglicht ein sofortiges, intuitives Verständnis der Steuerung.

9.2.1 Hieronymus Bosch - Triptychon Der Heuwagen

Die erste Anwendung zur Erkundung digitalisierter Gemälde zeigt das Triptychon "Der Heuwagen" des niederländischen Malers Hieronymus Bosch (um 1450-1516). In der digitalen Version sind der linke (*Der Garten Eden*) und der rechte Innenflügel (*Die Hölle*), sowie der

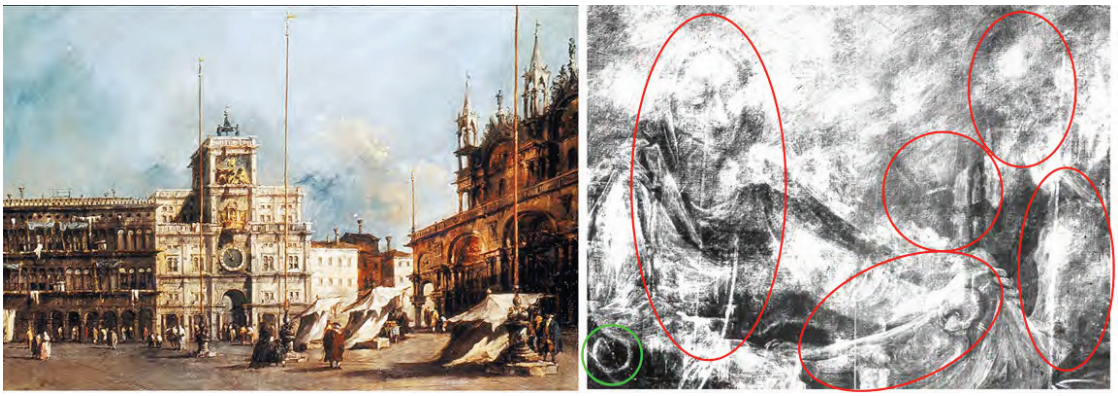

Abbildung 9.2: Francesco Guardi (1712-1793), Der Markusplatz in Venedig, Fotografie des Originalgemäldes, links und Ausschnitt der Röntgenaufnahme der Leinwand mit markierten Details einer Krippenszene, rechts [SFJSL04].

Mittelflügel (*Der Heuwagen*) zu sehen. Der Maler des ausklingenden Mittelalters ist bekannt für seine Gemälde mit einer Fülle von Details und faszinierenden Darstellungen dämonischer Figuren und Fabelwesen (siehe auch Abbildung 9.1). In den Museen, in denen Gemälde von Hieronymus Bosch zu sehen sind, führt dieser Detailreichtum immer wieder dazu, dass Besucher unerlaubterweise sehr dicht an die Bilder herantreten, um diese Details aus der Nähe zu betrachten. Hier bietet ein virtuelles Exponat Abhilfe, bei dem der Museumsbesucher das Werk aus entsprechendem Abstand betrachten und mit ihm interagieren kann und trotzdem die Möglichkeit besteht, interessante Detail hervorzuheben. Dabei wird die Zeigegestenerkennung zur Steuerung einer virtuellen Lupenfunktion verwendet, die er in Echtzeit über die virtuelle Leinwand gleiten lassen kann [MDS05b]. Um dem Anwender zu ermöglichen, auch für längere Zeit auf einem Detail zu verharren, ist für diese Anwendung neben der standardmäßig verwendeten Glättung der projizierten Zeigerichtung durch Mittelwertbildung (siehe Abschnitt 5.2.3) ein zweiter Nachverarbeitungsschritt notwendig, der es ermöglicht auch bei leichten, unvermeidbaren Bewegungen der Hand die virtuelle Lupe stabil auf einer bestimmten Position auf dem Bildschirm zu halten. Dieser Bewegungsglättung durch *smoothing splines* wie im Abschnitt 5.2.3 beschrieben wird automatisch zugeschaltet, wenn sich die Position der Zeigegeste innerhalb einer vordefinierten Zeit von etwa einer halben Sekunde nicht signifikant ändert.

9.2.2 Francesco Guardi - Markusplatz in Venedig

Eine zweite Anwendung zur Erkundung digitalisierter Gemälde zeigt ein Bild des italienischen Malers Francesco Guardi (1712-1793). Das Originalgemälde, auf dem der Markusplatz in Venedig zu sehen ist, hängt in der Gemäldegalerie der Akademie der bildenden Künste (AFA) in Wien[6]. Eine Untersuchung der Leinwand [SFJSL04] brachte ans Licht, dass Guardi für dieses Bild eine ältere, bereits bemalte Leinwand verwendet hatte, wie es im Rokoko

[6] http://www.akbild.ac.at/

Abbildung 9.3: Interaktion mit einem Gemälde von Francesco Guardi. Der Anwender steuert einen virtuellen Röntgenstrahl, um die im Hintergrund liegende Schicht des Gemäldes freizulegen [MDS05a].

durchaus üblich war. Eine Röntgenaufnahme des Gemäldes zeigt Details einer weihnachtlichen Krippenszene eines unbekannten Künstlers. Durch die Verwendung von Bleiweiß in den Ölfarben der damaligen Zeit zeigen sich beispielsweise die Konturen von Maria, die Jesus in einem Tuch in ihren Armen hält, zwei Hirten und der Kopf des Ochsen und des Esels (siehe Abbildung 9.2). Die klassischen Wege, diese Zusatzinformationen dem Museumsbesucher zugänglich zu machen, sind zum einen der Abdruck des Röntgenbildes in einem Museumsführer in Buchform oder durch eine Schautafel neben der Original mit erklärenden Texten und der Darstellung der Aufnahme. Auch die Verwendung eines heute im Museum üblichen Audioführers schließt sich durch den hohen visuellen Anteil der Zusatzinformationen aus. Im Rahmen des von der Europäischen Gemeinschaft geförderten Forschungsprojektes art-E-fact [GSH+04] wurde daher eine VRML-Anwendung entwickelt, in der beide Bilder in digitaler Form in zwei hintereinander liegenden Schichten dargestellt werden. Über eine JavaScript-Knoten im Szenengraphen wird an der Stelle, auf die der Anwender zeigt, der Blick durch das Originalbild der vorderen Schicht auf das dahinter liegende Röntgenbild freigegeben (siehe Abbildung 9.3). Die eigentlichen Zusatzinformationen über die einzelnen Details im Röntgenbild ist dabei über das automatische Abspielen von Sounddateien realisiert, sobald der Museumsbesucher für eine vorgegebene Zeit von mehr als einer halben Sekunde auf einem gefundenen Detail verharrt. Die Bereiche der interessanten Stellen im Bild sind dabei als vordefinierte Interaktionsregionen vordefiniert.

Abbildung 9.4: Sudoku-Spiel gesteuert durch multimodale Verknüpfung von Gesten- und Spracherkennung. Ein Spielstein ist durch die Zeigegeste aktiviert und mit der Zahl 1 gefüllt.

9.3 Multimodales Sudoku

Als Beispiel für die multimodale Verknüpfung von Zeigegesten- und Spracherkennung wurde im Rahmen dieser eine Sudoku-Anwendung entwickelt, in der ein vom Computer generiertes Zahlenrätsel gelöst werden muss.
Sudoku ist ein weltweit beliebtes Logikrätsel ähnlich einem Kreuzworträtsel. Ziel des Spiels in der heute üblichen Version ist es, ein 9×9-Gitter mit den Ziffern 1 bis 9 so zu füllen, dass jede Ziffer in einer Spalte, in einer Zeile und in einem Block (3×3-Unterquadrat) nur einmal vorkommt. Je nach Schwierigkeitsgrad des Rätsels werden mehr oder weniger der insgesamt 81 Felder bereits korrekt gefüllt vorgegeben, während die freien Felder durch den Spieler gefüllt werden müssen.
Sudoku-Spiele sind in den unterschiedlichsten Formen und Spielarten verfügbar: Angefangen vom klassischen Rätsel auf Papier, wie es beispielsweise in Zeitungen abgedruckt wird, über online spielbare Varianten im Internet bis hin zu eigenständigen Computersystemen für den mobilen Einsatz. Die hier beschriebene Sudoku-Anwendung verwendet eine große Projektionsfläche, auf der das Spielgitter dargestellt wird (siehe Abbildung 9.4). Der Spieler steht vor der Projektionsfläche und wird mit einem Mikrofon ausgestattet, um die entsprechenden

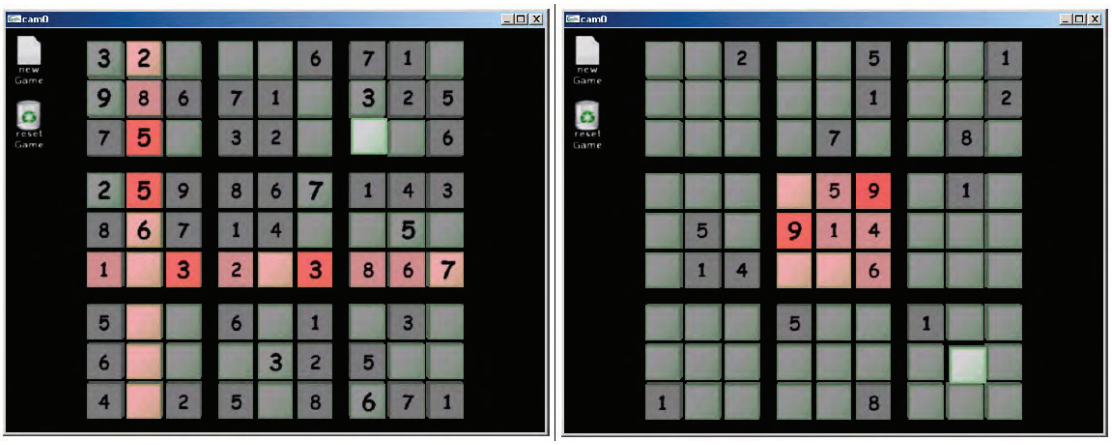

Abbildung 9.5: Fehlervisualisierung beim Sudoku-Spiel: Verletzung der Reihen- und Spaltenregel, links und Verletzung der Blockregel, rechts.

Zahlen über einen Sprachbefehl an den Rechner zu übermitteln. Um zu verhindern, dass der Spieler die gewünschte Position des Feldes für die neue Zahl umständlich durch Angabe von Zeilen und Spaltennummer definieren muss, deutet er einfach auf ein noch freies Feld, während er die Zahl nennt. Die Anwendung bietet drei verschiedene Schwierigkeitsgrade, bei denen entweder 30, 45 oder 60 der insgesamt 81 Felder beim Start eine neuen Spiels bereits gefüllt sind.

9.3.1 Eingabemodalitäten

Zu Beginn eines Spiels steht der Anwender mit einem Mikrofon auf einer durch Markierungen am Boden vordefinierten Position vor der Ausgabeleinwand. Zunächst wählt er einen Schwierigkeitsgrad durch Zeigen auf das entsprechende Menüsymbol oder durch Sprechen eines der Wörter "*easy*", "*medium*" oder "*hard*", wodurch entsprechend ein neues Rätsel erzeugt und auf dem Ausgabegerät angezeigt wird. Durch Zeigen auf ein leeres Feld und gleichzeitiges Nennen der entsprechenden Zahl, die in das Feld eingetragen werden soll, füllt er Zug um Zug das Gitter. Die verwendete Spracherkennung ist Kommando-basiert, untersucht also permanent den über das Mikrofon eingehenden Audiostrom und reagiert auf in einem Regelwerk abgelegte Wörter, Phrasen oder Sätze. Dies ermöglicht es, dass sich der Spieler trotz aktiver Spracherkennung mit anderen Personen unterhält und beispielsweise die aktuelle Spielsituation diskutiert. Um zu vermeiden, dass dadurch Zahlen versehentlich in das Gitter eingetragen werden, darf der Spieler die Zahlen nicht einfach sagen, sondern muss sie durch den Wortzusatz "*number*" versehen. Nur wenn er eines der Kommandos von "*number 1*" bis "*number 9*" gibt, während durch Zeigen auf das Spielfeld ein Feld markiert ist, wird die entsprechende Zahl in das Feld eingetragen. Hat der Spieler sich geirrt und eine falsche Zahl eingegeben, werden die entsprechenden Regelverletzungen durch rot markierte Reihen, Spalten oder Blöcke gekennzeichnet (siehe Abbildung 9.5). Um Felder wieder zu

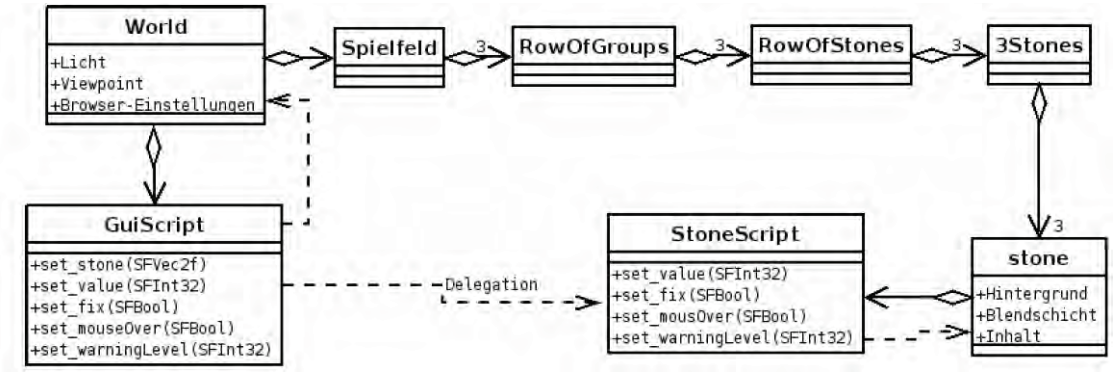

Abbildung 9.6: Klassen- und Knoten-Design der JavaScript-Implementierung des Sudoku-Spiels.

löschen, hat der Spieler die Möglichkeit, eines der Kommandos "*delete that*", "*delete this*", "*remove that*" or "*remove this*" zu geben. Wird währenddessen durch Zeigen ein Spielstein markiert, wird das entsprechende Feld wieder gelöscht. Das Spiel endet automatisch mit einer Meldung, wenn das Rätsel vollständig und korrekt gelöst wurde und ermöglicht dann ein neues Spiel in einem der drei Schwierigkeitsgrade zu starten.

9.3.2 Implementierung

Für die Implementierung und Darstellung des Sudoku-Spiels wurde eine dreidimensionale Oberfläche mit plastisch wirkenden Spielsteinen als Gitterelemente gewählt, obwohl die Spielidee selbst durch eine zweidimensionale Ausgabe realisierbar ist. Jeder einzelne der Spielsteine beispielsweise besteht aus drei verschiedenen einzelnen Objekten, einer Basisgeometrie (*background layer*), einem Beleuchtungselement (*illumination layer*), um selektierte Felder hervorzuheben und einem Texturelement (*content layer*), das einen schnellen Wechsel der Ziffer erlaubt . Durch die Verwendung einer 3D-Szene hebt sich die Anwendung von den klassischen, bekannten Sudoku-Varianten ab, und unterstützt damit auch die Akzeptanz der Anwender, neuartige Interaktionsformen wie die Kombination aus Zeigegestenerkennung und Spracherkennung bei einem Computerspiel auszuprobieren.
Die Regeln des Spiels sind als JavaScript-Knoten realisiert und somit beim Starten der 3D-Welt direkt im Szenengraphen verfügbar. Die Anwendung trennt sich in unterschiedliche Script-Knoten für die eigentliche Spiellogik und für die Erzeugung interner Ereignisse zur Eingabeverarbeitung und zur Visualisierung. Diese strikte Trennung von Logik, Eingabeverarbeitung und Visualisierung ermöglicht eine schnelle Änderung der Visualisierungskomponenten (beispielsweise einen Austausch der Geometrien) oder den Austausch der Eingabemodalitäten (beispielsweise die Verwendung einer Tastatur anstelle der Spracheingabe), ohne dabei die Elemente der Spielsteuerung ändern zu müssen.
Das Sequenzdiagramm (siehe Abbildung 9.6) für die Erzeugung und den Ablauf eines Sudoku-Spiels ist in fünf Einzelschritte unterteilt, deren Ablauf entweder durch längeres Deuten auf eine in der linken oberen Ecke des Bildschirms permanent eingeblendete Schalt-

fläche oder durch das Sprachkommando "*start new game*" aktiviert wird:

1. Interne Erzeugung eines vollständig gelösten Feldes für die Überprüfung der einzelnen Spielzüge zur Laufzeit des Spiels.
2. Erzeugung des korrespondierenden Spielfeldes mit den zu Spielstart nach außen sichtbaren ausgefüllten Feldern.
3. Visualisierung des Spielfeldes (inkl. der Interaktionsmethoden zur multimodalen Verarbeitung von Gesten- und Spracherkennung).
4. Überprüfung der vorgegebenen Lösung nach der Verarbeitung jedes einzelnen Spielzugs.
5. Methoden zur Fehlererkennung und Fehlerbehandlung wie bereits im Abschnitt 9.3.1 beschrieben.

Die Schritte drei bis fünf laufen dabei iterativ ab, wobei das System blockierend auf neue und sinnvolle Eingabe der Gesten- und Spracherkennung wartet. Eingehende Ereignisse werden in einem eigenen Skript-Knoten verarbeitet, auf zeitliche und logische Konsistenz überprüft und miteinander zu einem internen Ereignis zur Ablaufsteuerung verknüpft.
Für die Visualisierung der neu erzeugten Ereignisse werden bei jedem Durchlauf alle Felder des Spiels erneut ausgelesen, wenn ein neues internes Ereignis erzeugt wurde. Die unterschiedlichen Visualisierungsereignisse werden dann durch den Aufruf von fünf verschiedenen Methoden zur Steuerung der Benutzerschnittstelle realisiert, die in der folgenden Tabelle erläutert werden:

`VizStoneSet(x,y)`	Aktiviert die Überprüfung auf Fehleingaben oder notwendige Korrekturen für einen Stein an der Position `(x,y)`.
`setStoneValue(x,y,value)`	Setzt einen neuen Wert eines Steines an der Position `(x,y)` und aktualisiert die entsprechenden Texturen.
`fixStone(x,y,bool)`	Markiert den Stein an der Position `(x,y)` als unveränderbar, wenn `bool` auf *wahr* gesetzt ist, andernfalls wird der Stein als durch Benutzereingaben während des Spiels veränderbar angenommen.
`setStoneErrorLevel(x,y,level)`	Setzt die Fehlervisualisierung eines Steins an der Position `(x,y)` auf `level`. Zulässige Werte sind dabei 0:OK, 1:Warnung und 2:Fehler.
`setMouseOver(x,y,bool)`	Setzt alle Visualisierungs-Attribute eines Steins an der Position `(x,y)` . Die eventuelle Hervorhebung des Steins wird erreicht durch eine vorgegebene Translation und Farbänderung des Objektes.

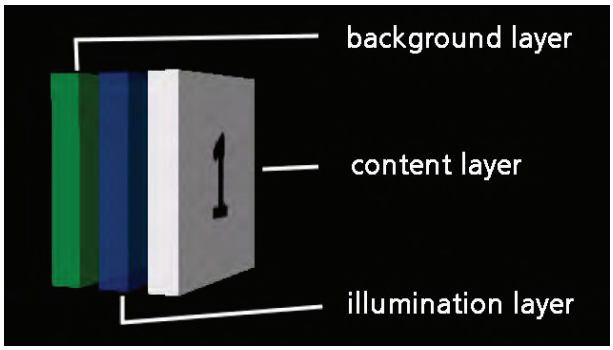

Abbildung 9.7: 3D-Vorlage eines Sudoku-Feldes.

Während für die Laufzeitdarstellung der Anwendung aus den bereits erwähnten Gründen der instantPlayer des instant**Reality**-Rahmenwerks verwendet wird, wurde zum Modellieren des Szenengraphen und der visuellen Elemente des Spiels wie beispielsweise die Spielsteine und ihre einzelnen Schichten (siehe Abbildung 9.7) White Dune [7] verwendet. White Dune ist eine freie und Open-Source-Software zur Erstellung und zur Bearbeitung von 3D-Modellen in VRML97 entwickelt an der Universität Stuttgart und hat sich auf die Entwicklung von Echtzeit-Szenengraphen spezialisiert.

9.4 Anatomisches Theater

Eine Anwendung, die eine Interaktion mit dreidimensionalem Szenen-Inhalt ausschließlich mittels videobasierter Erkennung einer Zeigegeste realisiert, ist das "Anatomische Theater" der Sonderausstellung *Computer.Medizin*[8] des Heinz-Nixdorf-Museumsforum in Paderborn. Aus dem Ausstellungstext: "*Die Ausstellung richtet sich sowohl an den interessierten Laien als auch an Mitarbeiter im Gesundheitswesen. Sie zeigt anhand spektakulärer Exponate den Nutzen und die Grenzen des Computers in der Medizin auf. Zahlreiche interaktive Exponate sorgen für eine hohe Attraktivität.*"
Die Sonderausstellung wurde vom 25. Oktober 2006 bis 1. Mai 2007 im Heinz-Nixdorf MuseumsForum in Paderborn und danach als Wanderausstellung vom 30. September 2007 bis 31. März 2008 im Ausstellungszentrum der Deutsche Arbeitsschutzausstellung DASA in Dortmund[9] gezeigt.
Das Exponat im Anatomischen Theater ermöglicht es dem Museumsbesucher, das Innere des menschlichen Körpers zu erkunden. Aus der Exponatbeschreibung: "*Ende des 16. Jahrhunderts entstanden die ersten Anatomischen Theater. Als Tempel der Sterblichkeit boten sie der interessierten Öffentlichkeit einen ebenso informativen wie faszinierenden Einblick in das menschliche Innere. In der medizinischen Ausbildung ersetzen heute moderne computer-*

[7] http://vrml.cip.ica.uni-stuttgart.de/dune/
[8] http://www.computer-medizin.de/
[9] http://www.dasa-dortmund.de/

Abbildung 9.8: Untersuchung von verschiedenen Körperschichten in einem virtuellen Museumsexponat des Heinz-Nixdorf-Museumsforum in Paderborn für die Sonderausstellung *Computer.Medizin*.

erzeugte Bilder und Programme vielfach die klassischen Anatomieatlanten. Sie ermöglichen dreidimensionale virtuelle Reisen durch den Körper, die zu einem tieferen Verständnis der Struktur und Vorgänge des Körpers beitragen. Diese Abbildungen wären mit dem menschlichen Auge allein nie zu sehen. Wie bei einer echten Sektion können Sie Körperteile aufklappen, Strukturen entfernen, Organe drehen oder Gewebe anfärben. Und das - im Gegensatz zur realen Situation - immer und immer wieder."
Die Gesten-basierte Erkundung des Körpers im Anatomischen Theater der Ausstellung ist eines von drei Leitexponaten: Durch Interaktion mittels einer Zeigegeste können die Besucher Haut, Nerven- und Kreislaufsystem sowie das Skelett des menschliche Körpers auf einem Großbildschirm betrachten. Die Anwendung zeigt das virtuelle Modell eines menschlichen Körpers in sieben verschiedenen Schichten:

- Haut
- Muskeln
- Innere Organe

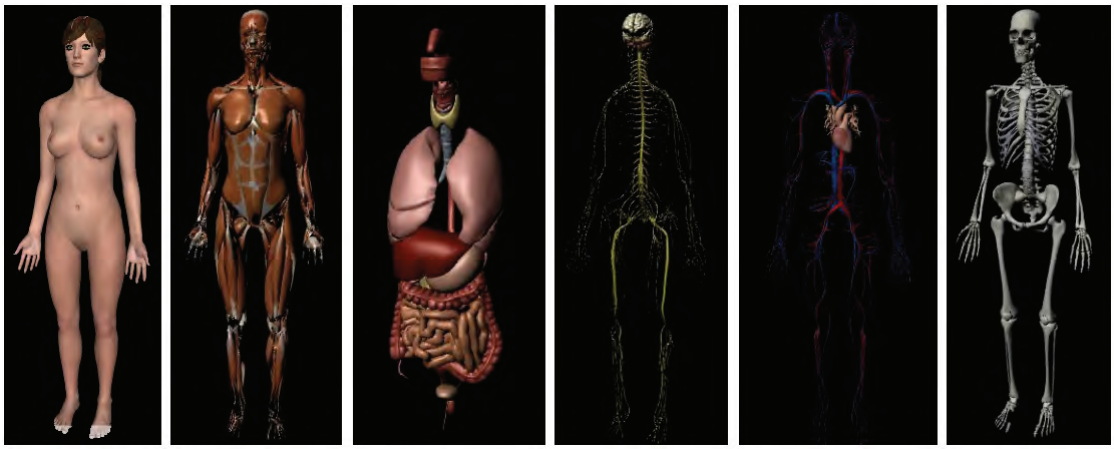

Abbildung 9.9: Einzelmodelle der sieben Körperschichten für das Museumsexponat *Anatomisches Theater*.

- Kardiovaskuläres System
- Lymphatisches System
- Nervensystem
- Skelett

Die einzelnen Schichten des Modells können über die Steuerung einer sichtbaren Clippingebene nacheinander entfernt bzw. wiederhergestellt werden. Dafür stehen dem Anwender zwei Schaltflächen mit Pfeilsymbolen zur Verfügung (siehe Abbildung 9.8), auf die er einfach deuten muss, um die Clippingebene und deren visuellen Repräsentation in der Szene auf- bzw. abwärts zu bewegen. Ein permanentes visuelles Feedback durch einen 3D-Cursor, der die derzeitige Position der Interaktion auf dem Bildschirm anzeigt, erleichtert dabei die Interaktion. Eine Anforderung an das Exponat war ursprünglich eine Drehung des Modells und die Möglichkeit, in die Szene hineinzuzoomen und aus der Szene herauszuzoomen. Nach ersten Tests stellte sich allerdings heraus, dass auf diese Art der Interaktion verzichtet werden musste, da sich aufgrund der Natur der dargestellten Szene (insbesondere in der obersten Ebene des Modells) das Risiko von unerwünschter Bedienung ergibt. Die Rotation des Modells wurde daher auf eine Drehung um die longitudinale Körperachse (mathematisch die y-Achse) beschränkt, so dass der dargestellte Avatar immer aufrecht in der Szene steht. Auf die Möglichkeit des Hineinzoomens und Herauszoomens wurde ganz verzichtet. Aufgrund dieser Beschränkungen wurde auch die Rotation des Modells durch die einfache Bedienung von zwei Schaltflächen realisiert, die am unteren Rand des Bildschirms links und rechts vom Modell gerendert werden. Alle Schaltflächen sind als MouseOver-Events realisiert, es reicht also aus, den Cursor über die Schaltfläche zu bringen, um direkt das entsprechende Ereignis (Rotation des Modells bzw. Translation der Clippingebene) auszulösen.

Abbildung 9.10: Zeigegestenerkennung im *Anatomischen Theater*: Bundespräsident Horst Köhler testet die videobasierte Gestenerkennung während der Ausstellungseröffnung.

9.5 Virtuelles Schachspiel

Die vierte und damit letzte der in diesem Kapitel vorgestellten Anwendungen beschreibt die Interaktion mit einem virtuellen dreidimensionalen Schachspiel. Implementierungen, die ein Schachspiel zwischen Mensch und Computer ermöglichen, sind seit Langem weit verbreitet. Der erste kommerzielle Schachcomputer für den Heimbedarf erschien im Jahre 1977. In den folgenden Jahren folgten immer leistungsfähigere Schachcomputer (siehe Abbildung 9.12), die dann in den 1990er Jahren allmählich durch reine Softwareprogramme für den häuslichen PC abgelöst wurden. Mit dem immer populärer werdenden Internet tauchten dann auch Programme auf, die ein Spielen über das Internet ermöglichten. Dabei entstanden auch rein auf JavaScript basierte Schachsysteme, die aber naturgemäß eine rein zweidimensionale Ausgabe in einer HTML-Seite verwenden. Sicherlich ersetzt eine Interaktion mit einem virtuellen 3D-Schachprogramm mittels Gestenerkennung nicht das haptische Empfinden eines klassischen Schachcomputers mit realen Figuren, die auf dem Brett gesetzt werden müssen. Dennoch ermöglicht die Gestenerkennung ein einfaches und intuitives Verhalten mit und in der virtuellen Welt. Zudem ermöglichen freie und Open-Source-Implementierungen in JavaScript wie beispielsweise *p4wn*[10] ein einfaches Einbinden des Schachkernprogramms (*chess engine*) in eine VRML-Welt, auch wenn die Spielstärke durch die Art der Implementierung

[10] http://p4wn.sourceforge.net/

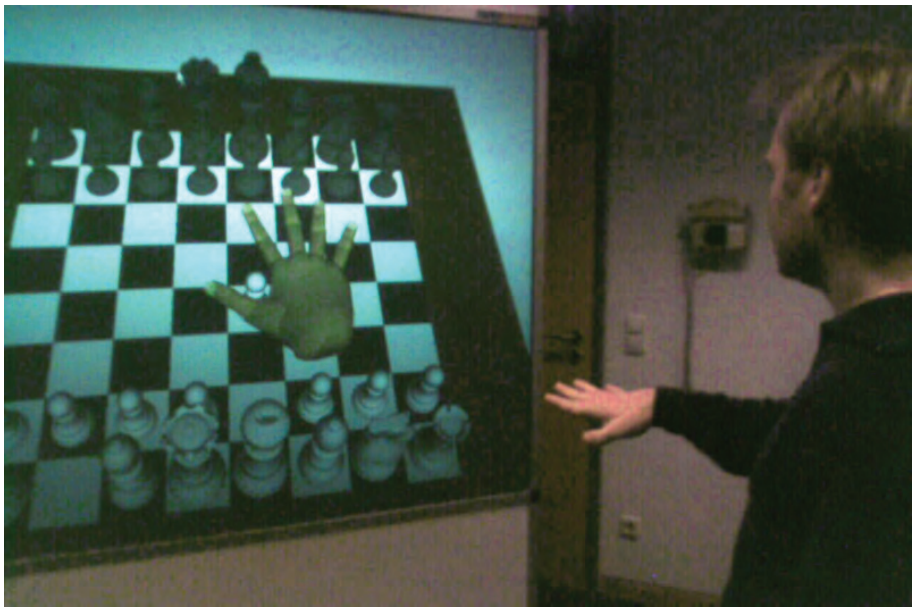

Abbildung 9.11: Interaktion mit einem virtuellen Schachspiel. Das Ziehen einzelner Schachfiguren wird über die 3D-Position der offenen und geschlossenen Hand gesteuert [Mal08b].

begrenzt ist und nicht mit den bekannten Kernsystemen konkurrieren kann.
Die üblichen Eingaben für einen Schachzug in den klassischen Schachprogrammen erfolgen entweder über die Tastatur, indem die entsprechenden Felder angegeben werden (beispielsweise "e5xd4" für "Figur auf e5 schlägt Figur auf d4") oder über die Computermaus, indem der Cursor über die zu ziehende Figur gebracht wird, durch Drücken und Halten der linken Maustaste die Figur festgehalten und über einem neuen Feld die Maustaste wieder freigegeben wird, um die Figur abzusetzen. Um bei der hier beschriebenen Anwendung eine Partie Schach gegen den Computer zu spielen, steht der Anwender direkt vor dem Ausgabebildschirm, auf dem ein dreidimensionales Schachbrett mit den entsprechenden Schachfiguren zu sehen ist. Ist der Spieler am Zug, setzt er die gewünschte Figur auf das entsprechende Feld. Dafür verwendet er analog zur Maus-Interaktion die beiden Gesten der offenen und geschlossenen Hand. Da die Gesten wie in Kapitel 6 beschrieben im dreidimensionalen Raum erkannt und unterschieden werden, kann eine virtuelle Repräsentation der menschlichen Hand in der entsprechenden Pose als dreidimensionaler Cursor verwendet werden (siehe Abbildung 9.11). Um dem Spieler das Aufnehmen und Absetzen von Spielfiguren zu erleichtern, wird eine Schnapp-Methode (*snapping tool selection*) verwendet, die eine nur ungefähre Positionierung der virtuellen Hand im Raum erfordert. Ändert der Spieler die Pose der Hand in der Nähe eine Figur von *offen* nach *geschlossen*, wird die dem Cursor am nächsten stehende Figur ermittelt und an die Geometrie der virtuellen Hand gebunden. Analog dazu wird beim Wechsel der Pose von *geschlossen* nach *offen* das nächstliegende Feld ermittelt und die Spielfigur auf diesem Feld abgesetzt.

Abbildung 9.12: Schachcomputer *Tandy Radio Shack 1650* aus den 1980er Jahren.

9.6 Zusammenfassung

Die in diesem Kapitel vorgestellten Anwendungen zeigen exemplarisch einen kleinen Teil der Demonstratoren, welche die in dieser Arbeit entwickelten Verfahren zur intuitiven Interaktion durch videobasierte Gestenerkennung verwenden. Verschiedene Anwendungen aus dem musealen Umfeld, zum Beispiel zur erweiterten Betrachtung von digitalisierten Gemälden oder zur interaktiven Untersuchung des menschlichen Körpers als mehrschichtiges, dreidimensionales Modell unterstützen die Verwendung der Zeigegestenerkennung als Eingabemodalität. Als Beispiel für eine multimodale Anwendung dient ein virtuelles Sudoku-Spiel, das als Eingabemodalitäten Gestenerkennung und Spracherkennung sinnvoll verbindet. Ein virtuelles Schachspiel verwendet die in dieser Arbeit beschriebene Gesten-Klassifikation, um dem Anwender zu ermöglichen, die Spielfiguren im virtuellen dreidimensionalen Raum vom Schachbrett aufzunehmen und auf einem anderen Feld wieder abzustellen.
Für die ersten vier Anwendungen, die in diesem Kapitel vorgestellt wurden, wird das Verfahren der *Interaktion durch Punktprojektion* als Mittel zur Gestenerkennung eingesetzt, da die Anwendungen ausschließlich durch die Verwendung einer Zeigegeste bedient werden sollen. Alle vier Anwendungen werden in öffentlichen Umgebungen eingesetzt, die von einer hohen Anzahl von unterschiedlichen und teilweise auch technisch unversierten Anwendern besucht werden. Daher erfordert die Interaktion zwangsläufig die Verwendung eines Verfahrens, mit dem potentiell jeder Besucher als neuer Benutzer des System gewonnen werden kann und jeder Nutzer in der Lage ist, die Anwendung zu bedienen. Insbesondere die Möglichkeit, mit individuellen Unterschieden in der Interpretation der Zeigegeste umzugehen, lässt dabei im Vergleich mit der Verfahren der *Rekonstruktion von Aktiven Formen* die Wahl auf das Verfahren der Punktprojektion fallen.
Für das virtuelle Schachspiel wurde das Verfahren der *Interaktion durch Merkmalbasierte Gesten-Klassifikation* ausgewählt. Die wichtigste Forderung an eine dreidimensional gerenderte Schachpartie, die mittels videobasierter Gestenerkennung gespielt wird, ist es, dass die

Spielfiguren vom Anwender auf dem Schachbrett versetzt werden können. Prinzipiell ist dies durch die Verwendung einer Zeigegeste möglich. Der Anwender deutet auf eine Figur, um sie zu selektieren und danach auf ein freies Spielfeld, um die Figur dort wieder abzustellen. Diese Art der Interaktion wird beispielsweise von einer Vielzahl von Schachanwendungen, die klassisch mit der Maus als Eingabegerät gespielt werden, verwendet. Soll der immersive Charakter der Interaktion mit einer virtuellen Anwendung hervorgehoben werden, ist es für einen Schachspieler allerdings intuitiver, die Spielfiguren zu greifen und an der gewünschten Position auf dem Schachbrett wieder abzustellen. Aus diesem Grund wird in der Anwendung die Erkennung der offenen und der geschlossenen Hand verwendet, um die Züge auszuführen.

Kapitel 10

Zusammenfassung und Ausblick

Das Ziel dieser Arbeit bestand darin, Verfahren zu entwickeln, die es ermöglichen, durch die Verwendung von Handgesten intuitiv und einfach mit Computersystemen und deren Anwendungen zu interagieren. Zu diesem Zweck wurden zunächst drei Anforderungen an die zu entwickelnden Verfahren erarbeitet, die notwendig sind, um eine gestenbasierte Interaktion für jeden Anwender nutzbar zu machen:

1. Gerätelose Bedienung

   Gerätelosigkeit ist eine elementare Forderung an die Verfahren der intuitiven Interaktion. Die Verwendung von technischen Hilfsmitteln wie beispielsweise Datenhandschuhen oder anderen Interaktionsgeräten ist auszuschließen. Nur wenn der Anwender ohne Verzögerung mit der Interaktion starten kann, ist eine einfache und intuitive Interaktion möglich. Aus diesem Grund verwenden alle vier in dieser Arbeit entwickelten Verfahren ein Kamerasystem, das die Bewegungen der Hand des Anwenders erfasst und auswertet. Durch die Tatsache, dass die Kameras außerhalb der direkten Sicht des Anwenders positioniert werden können, ist der Ausgabebildschirm das einzige technische Gerät, das für den Anwender sichtbar ist und unterstützt somit den immersiven Charakter der Interaktion. Um ausreichend Informationen wie beispielsweise der Position der Hand im Raum gewinnen zu können, wird in allen vier Verfahren ein kalibriertes Stereokamerasystem mit zwei Kameras verwendet, das in der Lage ist, aus 2D-Punktkorrespondenzen dreidimensionale Punkte zu triangulieren.

2. Reduktion des Trainingsaufwands

   Intuitive Interaktion durch Gestenerkennung ist nur möglich, wenn für einen neuen Anwender kein oder nur ein minimaler Trainingsaufwand notwendig ist und der Anwender in die Lage versetzt wird, ohne störende Verzögerung mit der eigentlichen Anwendung zu interagieren. Aus diesem Grund wurde für die in dieser Arbeit entwickelten Verfahren auf eine individuelle Trainingsphase verzichtet. Eine Ausnahme dazu bildet das in Kapitel 6 beschriebene Verfahren der merkmalbasierten Gesten-Klassifikation, bei dem eine optionale Trainingsphase zum Anlernen der individuellen Gesten die Erkennungsrate des Verfahrens erhöht. Eventuelle, für ein Verfahren notwendige Lernphasen,

wie beispielsweise das Anlernen eines Modells der verwendeten Handgesten, wurden auf einen Zeitpunkt vor der eigentlichen Interaktion des Anwenders verschoben und ermöglichen somit einem neuen Nutzer des Systems, ohne Verzögerung mit der Anwendung zu interagieren.

3. Echtzeitfähigkeit

   Eines der wichtigsten Kriterien für eine intuitive Interaktion ist die Echtzeitfähigkeit der entwickelten Verfahren. Zu einen ist eine Wiederholrate von zwanzig Bildern pro Sekunde notwendig, um eine flüssige Darstellung der Ergebnisse zu gewährleisten. Zum anderen darf die Verzögerung zwischen realem Ereignis der Hand und der entsprechenden Reaktion des Systems nicht länger als 200 Millisekunden betragen, um von einem Anwender nicht als störend empfunden zu werden. Die in den Kapiteln 4, 5 und 6 entwickelten Verfahren erfüllen diese Anforderungen. Das in Kapitel 7 vorgestellte Verfahren der Momenten-Analyse erreicht mit durchschnittlich 13 Bildern pro Sekunde eine Bildwiederholrate, die zwar nicht den Anforderungen der Echtzeitfähigkeit genügt, aber dennoch bereits für eine interaktive Anwendung geeignet ist.

Auf Grundlage der aufgestellten Anforderungen für eine intuitive Interaktion und einer Untersuchung des derzeitigen Stands von Forschung und Technik wurden zunächst zwei Hauptprobleme derzeitiger Algorithmen identifiziert, die für eine Verwendung eines Verfahrens für eine intuitive Interaktion zu lösen sind:

1. Die Initialsuche nach der Geste muss vollständig automatisch und in Echtzeit ablaufen. Der Anwender sollte nicht darauf achten müssen, dass er beispielsweise die Hand vor einer Kamera für das System "richtig" positionieren und orientieren muss, damit das Verfahren die Handgeste korrekt erkennen und verfolgen kann.

2. Die Verfahren müssen in der Lage sein, alle zur gestenbasierten Interaktion und zur eventuellen Klassifikation unterschiedlicher Gesten relevanten Merkmale in Echtzeit zu bestimmen.

Als Lösung der beiden Probleme wurden deshalb in dieser Arbeit vier verschiedene Verfahren entwickelt, die abhängig von den Aufgaben, für die sie eingesetzt werden, unterschiedliche Arten der Interaktion durch videobasierte Handgestenerkennung realisieren:

1. Interaktion durch Rekonstruktion von Aktiven Formen

   Dieses Verfahren ist in der Lage, eine Zeigegeste im Raum zu erkennen und zu verfolgen. Das Verfahren beruht auf der Suche und Anpassung einer zuvor definierten Kontur der Zeigegeste. In einer initialen Lernphase wird aus einer Vielzahl von Trainingsbildpaaren ein statistisches Modell der Geste aufgestellt und berechnet. Das so entstandene Punktverteilungsmodell der Kontur der Zeigegeste dient in den folgenden iterativen Schritten dem Auffinden von geeigneten Startkonturen auf den Bildebenen und der folgenden Feinanpassung der Konturen an die realen Bildinformationen.

Das initiale Auffinden der Startkonturen wird sowohl über eine geeignete Analyse von Segmentierungsergebnissen als auch durch die Verwendung des nichtlinearen Optimierungsverfahrens *Simulated Annealing* realisiert. Für die Feinanpassung werden die Initialkonturen entsprechend ihrer angelernten Deformationsmöglichkeiten an die Kanteninformationen der Kamerabilder angepasst (*Active Shape Model*). Erst in einem letzten Schritt des Verfahrens werden aus den resultierenden 2D-Konturen unter Verwendung des kalibrierten Stereokamerasystems die zur Interaktion notwendigen dreidimensionalen Parameter wie Position der Geste im Raum und die Zeigerichtung ermittelt. Aufgrund der hohen notwendigen Rechenleistung für die Initialsuche ist das Verfahren auf eine einzige vom System angelernte Geste beschränkt. Dieser Umstand kann allerdings auch dazu verwendet werden, um automatisch zu detektieren, wenn ein Anwender eine andere als die vom System zur Interaktion geforderte Zeigegeste verwendet.

2. Interaktion durch Punktprojektion

   Das Verfahren der Punktprojektion beruht auf der Detektion der Spitze des Zeigefingers auf Bildbasis und deren 3D-Rekonstruktion und Projektion mittels eines Referenzpunktes auf die Ausgabefläche des Bildschirms. Dabei macht sich das Verfahren die Informationen des kalibrierten Stereokamerasystems und eine Segmentanalyse zu Nutze, um die Fingerspitze robust zu detektieren und zu verfolgen. Durch eine Reduktion der notwendigen Information über die Zeigegeste auf die Spitze des Zeigefingers benötigt das Verfahren keine modellbildende Trainingsphase und ermöglicht es jedem Anwender, sofort und intuitiv zu interagieren. Geeignete Nachverarbeitungsschritte wie beispielsweise eine Glättung des Zielpunktes der Interaktion durch Spline-Funktionen unterstützen die intuitive Interaktion. Das Verfahren beinhaltet außerdem zwei unterschiedliche Methoden zur Erzeugung von Selektionsereignissen mittels der Analyse von vordefinierten Ereignisregionen und der Analyse der Positionsgeschwindigkeit der Interaktionspunkte. Dadurch wird den Anwender in die Lage versetzt, Elemente einer grafischen Benutzerschnittstelle ausschließlich durch die Verwendung der Zeigegestik zu bedienen. Obwohl dieses Verfahren in seine Allgemeinheit nicht in der Lage ist, unterschiedliche Gesten voneinander zu trennen, ist es für eine intuitive Interaktion mittels einer einzigen Geste wohl am besten geeignet, da es deshalb auch auf individuelle Unterschiede der Anwender bei der Interpretation einer Handgeste robust reagieren kann.

3. Interaktion durch merkmalbasierte Gesten-Klassifikation

   Dieses Verfahren ist in der Lage, verschiedene statische Handgesten in Echtzeit voneinander zu unterscheiden. In Ergänzung zu dem Verfahren der Interaktion durch Punktprojektion kann das Verfahren eingesetzt werden, um beispielsweise neben der Zeigegeste auch die offene und die geschlossene Hand zu erkennen. Dadurch wird insbesondere bei der Interaktion mit virtuellen dreidimensionalen Welten die Möglichkeit geschaffen, virtuelle Objekte aufzunehmen und an einer anderen Stelle wieder abzulegen. Das Verfahren beruht auf der Einordnung von Merkmalsvektoren durch einen Naïven Bayes-Klassifikator. Um die Echtzeitfähigkeit des Verfahrens zu gewährleisten,

werden die Merkmale im Zweidimensionalen, also auf Bildbasis bestimmt. Dazu werden unter Verwendung der Informationen des kalibrierten Stereokamerasystems die Segmente der Handgeste in den Kamerabildern ermittelt. Aus den binarisierten Handsegmenten werden für die Erkennung relevante Merkmale extrahiert, die sowohl für die initiale Modellbildung (Klassifikation) als auch für die Klassierung der Geste zur Laufzeit des Systems verwendet werden. Für die Interaktion stellt das Verfahren neben der ermittelten Geste auch die Position der Hand im Raum bereit. Sowohl die Anzahl als auch die Art der zu unterscheidenden Gesten ist prinzipiell nicht begrenzt. Die einzige Forderung dabei ist, dass die Handposen bereits auf Bildebene auch bei einer Rotation der Hand statistisch unterscheidbare Merkmale aufweisen, die in Echtzeit berechnet werden können. Damit zeigt sich aber auch direkt die Grenze des Verfahrens: Der Algorithmus ist nicht in der Lage, die Orientierung der Hand im Raum zu bestimmen.

4. Interaktion durch Momenten-Analyse

   Das Verfahren der Momenten-Analyse rekonstruiert den dynamischen Prozess einer Handgeste und ermöglicht so eine einfache und intuitive Interaktion beispielsweise durch Zugreifen und Loslassen eines virtuellen dreidimensionalen Objektes mit dem Pinzettengriff, also ausschließlich mit Daumen und Zeigefinger. Das Verfahren beruht auf der Analyse der Pseudo-Zernike-Momente von Handsilhouetten und vergleicht die berechneten Momente eines neuen Kamerabildpaares mit den Datensätzen einer zuvor generierten Silhouettendatenbank. Die künstlichen Silhouetten werden dabei durch die Projektion eines virtuellen Handmodells unter Verwendung der Kamerakalibrierungsdaten erzeugt. Das Verfahren ist insbesondere in der Lage, auch die Orientierung der Hand im Raum zu ermitteln, und ermöglicht so die Rekonstruktion von feinmotorischen Interaktionsprozessen. Sowohl durch eine Diskretisierung des dynamischen Prozesses der Geste in eine vorgegebene Anzahl von statischen Einzelposen als auch durch eine Parallelisierung der bildverarbeitenden Schritte und deren Berechnung auf der Hardware der Grafikkarte erfüllt das Verfahren zwar noch nicht die Anforderung an eine Echtzeitfähigkeit, erreicht mit durchschnittlich 13 Bildern pro Sekunde eine Bildwiederholrate, die bereits für eine interaktive Anwendung geeignet ist.

Diese vier neu entwickelten Verfahren wurden in Hinblick auf die aufgestellten Forderungen an Verfahren zur intuitiven Interaktion, auf die detektierbaren Gesten und auf ihre Einsatzmöglichkeiten für unterschiedliche Anwendungsklassen miteinander verglichen und bewertet. In einer Usablity-Studie wurde das Verfahren der *Interaktion durch Punktprojektion* evaluiert, indem 81 neue Anwender zum einen eine messbare Aufgabe lösen mussten, um ein objektives Kriterium für die Verwendbarkeit des Verfahrens analysieren zu können und zum anderen durch eine Befragung das subjektive Empfinden der Nutzer ausgewertet werden konnte.
Neben den neu entwickelten Algorithmen wurden die für die Verfahrensentwicklung notwendigen Grundlagen der videobasierten Analyse, insbesondere die intrinsische und extrinsische Kalibrierung der Kameras und die Rekonstruktion von korrespondierenden Bildpunkten zum einem dreidimensionalen Punkt erläutert. Die praktische Verwendbarkeit der in dieser Arbeit

entwickelten und implementierten Verfahren wurde anhand verschiedener umgesetzter Anwendungen dokumentiert.
Mit den in dieser Arbeit neu entwickelten Verfahren zur videobasierten Gestenerkennung konnte die in der Einleitung dieser Arbeit aufgestellte Hypothese, dass es möglich ist, *"Verfahren zu entwickeln, die eine intuitive Interaktion zwischen Mensch und Computer ausschließlich durch die Verwendung von Handgesten ermöglichen und dabei die Forderungen nach geräteloser Bedienung, Reduktion des Trainingsaufwandes und Echtzeitfähigkeit erfüllen"*, erfolgreich nachgewiesen werden.

10.1 Ausblick

Mit Ausnahme der Verfahren der Punktprojektion und der merkmalbasierten Gestenklassifikation, die bereits gleichzeitig in einem einzigen System verwendet werden können, sind die in dieser Arbeit entwickelten Algorithmen getrennt voneinander zu verwenden. Die Auswahl des Verfahrens ist abhängig von der Anwendung, für die eine gestenbasierte Interaktion eingesetzt werden soll. Dies unterstützt zwar die Idee, dass Interaktion immer anwendungsbezogen und damit auch benutzerorientiert sein soll, erschwert aber ein schnellen Wechsel der Anwendung selbst, da in diesem Fall immer auch das Interaktionssystem gewechselt werden muss. Aufgabe für zukünftige Arbeiten muss es deshalb sein, die in dieser Arbeit vorgestellten Verfahren in ein einziges System zu integrieren und eine parallele Verwendung der Verfahren oder zumindest einen automatischen Wechsel der Verfahren zu ermöglichen.

Rechnersystem und Computerprogramme, die vom Anwender nicht mehr bewusst als solche wahrgenommen werden, spielen eine immer wichtiger werdende Rolle. Dies zeigt sich in den wissenschaftlichen Publikationen und den Produktentwicklungen der letzten Zeit in den aktuellen Forschungs- und Entwicklungsgebieten der Umgebungsintelligenz (engl.: *Ambient Intelligence*, AmI) [Enc08] und der Rechnerallgegenwart (engl.: *Ubiquitous Computing*, UC). Damit erhöht sich aber auch die Forderung nach Interaktionsformen, die natürlich und intuitiv anwendbar sind. Videobasierte Gestenerkennung kann hier einen Beitrag leisten, Computersysteme ohne technische Hilfsmittel zu bedienen und somit intuitive Interaktion zu ermöglichen. Wenn sich der Anwender aber nicht bewusst ist, dass er mit einem Computersystem interagiert, können auch keine Einschränkungen vorausgesetzt werden, die der Anwender einhalten muss. Für ein gestenerkennendes Verfahren bedeutet dies, dass beispielsweise auch eine beidhändige Interaktion möglich sein muss. Aufgabe zukünftiger Arbeiten sollte es deshalb sein, die in dieser Arbeit entwickelten Verfahren sowohl für den Einsatz einer beidhändigen Interaktion als auch für den Einsatz in einer Mehrbenutzerumgebung, in der mehrere Anwender gleichzeitig interagieren können, weiter zu entwickeln.

Wie im Verfahren der Momentenanalyse gezeigt, kann die immense Steigerung der Rechenleistung der letzten Zeit von Haupt- und Grafikprozessoren dazu verwendet werden, aufwendige Berechnungen, die für ein videobasiertes Gestenerkennungsverfahren notwendig sind, massiv parallel zu verarbeiten und somit die möglichen Anwendungsgebiete der intuitiven Gestenerkennung zu erweitern. Dieses Konzept der Parallelisierung von Aufgaben für die Rekonstruktion von statischen und dynamischen Handgesten ist in zukünftigen Arbeiten fortzuführen und zu erweitern.

Abkürzungen

AABB	Umschließendes Rechteck (*Axis-aligned bounding box*)
AFA	Akademie der bildenden Künste (*Academy of Fine Arts*)
AmI	Umgebungsintelligenz (*Ambient Intelligence*)
AR	Erweiterte Realität (*Augmented Reality*)
ASL	Amerikanische Gebärdensprache (*American Sign Language*)
ASM	Aktive Formen (*Active Shape Model*)
COP	Projektionszentrum (*Center of Projection*)
CPU	Hauptprozessor (*Central Processing Unit*)
CUDA	*Compute Unified Device Architecture*
DASA	Deutsche Arbeitsschutzausstellung
DOF	Freiheitsgrade (*Degree Of Freedom*)
FPS	Bilder pro Sekunde (*frames per second*)
FSM	Endlicher Automat (*Finite State Machine*)
GFLOPS	*Giga Floating Point Operations Per Second*
GHz	*Giga Hertz*
GPGPU	*General Purpose Computation on Graphics Processing Unit*
GPU	Grafikprozessor (Graphics Processing Unit)
GUI	Grafische Benutzeroberfläche (*Graphical User Interafce*)
HCI	Mensch-Computer Interaktion (*Human-Computer Interaction*)
HNF	Heinz Nixdorf Museumsforum
ISO	Internationale Organisation für Normung (*International Organization for Standardization*)
MR	Vermischte Realität (*Mixed Reality*)
OBB	Orientiertes, umschließendes Rechteck (*Oriented bounding box*)
PDM	Punktverteilungsmodell *Point Distribution Model*
PCA	Hauptkomponentenanalyse (*Principal Components Analysis*)
SA	Simulierte Abkühlung (*Simulated Annealing*)
SDK	Entwicklungsumgebung (*Software Development Kit*)
SVM	Stützvektormethode (*Support Vector Machine*)
UC	Rechnerallgegenwart (*Ubiquitous Computing*)
VR	Virtuelle Realität (*Virtual Reality*)
VRML	Virtual Reality Modeling Language
X3D	Extensible 3D

Literaturverzeichnis

[ABEN08] G. Amayeh, G. Bebis, A. Erol, and M. Nicolescu. Hand-based verification and identification using palm-finger segmentation and fusion. In *Computer Vision and Image Understanding (CVIU)*, 2008.

[AHS03] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and animation of anatomically based human hand models. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 98–109, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[AP95] A. Azarbayejani and A. Pentland. Camera self-calibration from one point correspondence. 1995. MIT media laboratory, perceptual computing technical report 341.

[ARHN08] M. Al-Rajab, D. Hogg, and K. Ng. A comparative study on using zernike velocity moments and hidden markov models for hand gesture recognition. In *AMDO '08: Proceedings of the 5th international conference on Articulated Motion and Deformable Objects*, pages 319–327, Berlin, Heidelberg, 2008. Springer-Verlag.

[BDC06] B. R. Bedregal, G. P. Dimurp, and A. C. Costa. Hand gesture recognition in an interval fuzzy approach. In *Tendencias em Mathematica Aplicada e Computational TEMA*, volume 8, pages 21–31, 2006.

[BDR04] J. Behr, P. Dähne, and M. Roth. Utilizing x3d for immersive environments. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pages 71–78, New York, NY, USA, 2004. ACM.

[BH08] A. Birdal and R. Hassanpour. Region based hand gesture recognition. In Steve Cunningham, editor, *WSCG 2008. Communications Papers*, Plzen, Check Republic, 2008. University of West Bohemia, European Association for Computer Graphics (Eurographics).

[BI98] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[BKMM+04] M. Bray, E. Koller-Meier, P. Mueller, L. Van Gool, and N. N. Schraudolph. 3d hand tracking by rapid stochastic gradient descent using a skinning model. In A. Chambers and A. Hilton, editors, *1st European Conference on Visual Media Production (CVMP)*, pages 59–68. IEE, March 2004.

[BLL02] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 423, Washington, DC, USA, 2002. IEEE Computer Society.

[BS02] R. Bowden and M. Sarhadi. A non-linear model of shape and motion for tracking finger spelt american sign language. 20(9-10):597–607, August 2002.

[Bun08] Statistisches Bundesamt. *Wirtschaftsrechnungen Private Haushalte in der Informationsgesellschaft - Nutzung von Informations-und Kommunikationstechnologien (IKT)*. Fachserie 15 IKT 2007 Reihe 4. Statistisches Bundesamt, Wiesbaden, April 2008.

[Bux02] W. Buxton. Less is more (more is less). In P. Denning, editor, *The invisible future: the seamless integration of technology into everyday life*, pages 145–179, New York, NY, USA, 2002. McGraw-Hill, Inc. 0-07-138224-0.

[BW54] A. B. Bhatia and E. Wolf. On the circle polynomials of zernike and related orthogonal sets. In *Philosophical Society of Cambridge*, volume 50, pages 40–48, 1954.

[BWK+04] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. pages 391–401. Springer-Verlag, 2004.

[CAHS06] T. Coogan, G. Awad, J. Han, and A. Sutherland. Real time hand gesture recognition including hand segmentation and tracking. In *ISVC (1)*, volume 4291 of *Lecture Notes in Computer Science*, pages 495–504. Springer, 2006.

[CBV03] C. Colombo, A. Del Bimbo, and A. Valli. Visual capture and understanding of hand pointing actions in a 3-d environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(4):677–686, 2003.

[CESJG05] Q. Chen, A. El-Sawah, C. Joslin, and N. D. Georganas. A dynamic gesture interface for virtual environments based on hidden markov models. In *IEEE Intl. Workshop on Haptic, Audio and Visual Environments and their Applications, HAVE*, pages 110–115, Ottawa, Ontario, Canada, 2005. IEEE Computer Society.

[Cor08] NVIDIA Corporation. Nvidia cuda programming guide version 2.1. *[Internet]*, 2008. URL: http://www.nvidia.com/cuda/.

[CRM03] C.-W. Chong, P. Raveendran, and R. Mukundan. An efficient algorithm for fast computation of pseudo-zernike moments. *International Journal of Pattern Recognition and Artificial Intelligence IJPRAI*, 17(6):1011–1023, 2003.

[CT01] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engeneering, University of Manchester, 2001.

[CTCG95] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision Image Understanding*, 61(1):38–59, 1995.

[CVB04] S. Carbini, J. E. Viallet, and O. Bernier. Pointing gesture visual recognition by body feature detection and tracking. In *ICCVG (International Conference on Computer Vision and Graphics 2004)*, 2004.

[Dor99] K. Dorfmüller. Robust tracking for augmented reality using retroreflective markers. *Computers & Graphics*, 23(6):795–800, 1999.

[DS99] B. Deimel and S. Schröter. Improving hand-gesture recognition via video based methods for the separation of the forearm from the human hand. In *Gesture Workshop'99*, 1999.

[DT02] J. Deng and H. T. Tsui. A pca/mda scheme for hand posture recognition. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 294–299, May 2002.

[EAHAM08] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis. A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In *Proceedings of the 19th International Conference on Pattern Recognition ICPR 2008*, December 2008. ISBN: 978-1-4244-2175-6.

[EBN+07] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.*, 108(1-2):52–73, 2007.

[ELD01] M. Ehrenmann, T. Liitticke, and R. Dillmann. Dynamic gestures as an input device for directing a mobile platform. In *In Proceedings of the 2001 International Conference on Robotics and Automation*, pages 21–26, Seoul, Korea, 2001.

[Enc08] J. L. Encarnação. Ambient intelligence - the new paradigm for computer science and for information technology (ambient intelligence - das neue paradigma der informatik und der informationstechnologie). *it - Information Technology*, 50(1):5–6, 2008.

[EPdRH02] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks - a review. *Pattern Recognition*, 35(10):2279–2301, 2002.

[Fau93] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[FI02] E. Foxlin and InterSense Inc. *Motion Tracking Requirements and Technologies*. L. Erlbaum Associates Inc., Mahwah, NJ, 2002. In: Kay Stanney (ed.) Handbook of virtual environment technology (Chapter 8).

[Fin03] G. A. Fink. *Mustererkennung mit Markov-Modellen*. Leitfäden der Informatik. B. G. Teubner, Stuttgart – Leipzig – Wiesbaden, 2003.

[FKW04] D. D. Frantz, S. R. Kirsch, and A. D. Wiles. Specifying 3d tracking system accuracy - one manufacturer's view. In *Proceedings des Workshops Bildverarbeitung für die Medizin 2004*, pages 234–238, Berlin, 2004. Springer Verlag.

[Flu06] J. Flusser. Moment invariants in image analysis. *Proceedings of the World Academy of Science, Engineering and Technology*, 11:196–201, February 2006.

[Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. In *IRE Transactions on Electronic Computers*, pages 260–268, 1961. EC-10(2).

[FRF08] Jonas Fredriksson, Sven Berg Ryen, and Morten Fjeld. Real-time 3d hand-computer interaction: optimization and complexity reduction. In *NordiCHI '08: Proceedings of the 5th Nordic conference on Human-computer interaction*, pages 133–141, New York, NY, USA, 2008. ACM.

[FTJL08] O. M. Foong and S. Wibowo T. J. Low. Hand gesture recognition: Sign to voice system (s2v). In *Proceedings of World Academy of Science, Engineering and Technology*, volume Volume 32, pages 32–36, August 2008. ISSN: 2070-3740.

[Fun02] S. Funck. Video-based handsign recognition for intuitive human-computer-interaction. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 26–33, London, UK, 2002. Springer-Verlag.

[FWCL07] Y. Fang, K. Wang, J. Cheng, and H. Lu. A real-time hand gesture recognition method. In *ICME*, pages 995–998, 2007.

[Gav99] D. M. Gavrila. The visual analysis of human movement: a survey. *Comput. Vis. Image Underst.*, 73(1):82–98, 1999.

[GSH+04] S. Göbel, U. Spierling, A. Hoffmann, I. Iurgel, O. Schneider, J. Dechau, and A. Feix, editors. *Technologies for Interactive Digital Storytelling and Entertainment, Second International Conference, TIDSE 2004, Darmstadt, Germany, June 24-26, 2004, Proceedings*, volume 3105 of *Lecture Notes in Computer Science*. Springer, 2004.

[GT05] H. Guan and M. Turk. 3d hand pose reconstruction with isosom. In *ISVC*, pages 630–635, 2005.

[Hal08] T. R. Halfhill. Parallel processing with cuda. InStat Microprocessor Report, January 28 2008.

[HE04] P. R. Harding and T. J. Ellis. Recognizing hand gesture using fourier descriptors. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*, pages 286–289, Washington, DC, USA, 2004. IEEE Computer Society.

[HH97] T. Heap and D. Hogg. 3d deformable hand models. In *Proceedings of Gesture Workshop on Progress in Gestural Interaction*, pages 131–139, London, UK, 1997. Springer-Verlag.

[HL00] J. Hoey and J. J. Little. Representation and recognition of complex human motion. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1752, 2000.

[HMLW03] C. Hu, M.Q. Meng, P.X. Liu, and X. Wang. Visual gesture recognition for human-machine interface of robot teleoperation. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2:1560–1565 vol.2, Oct. 2003.

[HN04] H. Hse and A. R. Newton. Sketched symbol recognition using zernike moments. In *International Conference on Pattern Recognition*, pages 367–370, 2004.

[HSW08] R. Hassanpour, A. Shahbahrami, and S. Wong. Adaptive gaussian mixture model for skin color segmentation. In *Proceedings of World Academy of Science, Engineering and Technology, Vol. 31, ISSN 1307-6884*, pages 1–6, July 2008.

[HTH00] P. Hong, M. Turk, and T. S. Huang. Constructing finite state machines for fast gesture recognition. In *In Proc. 15th ICPR*, pages 691–694, 2000.

[Hu62] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, February 1962.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[ICLB05] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu. Dynamic hand gesture recognition using the skeleton of the hand. *EURASIP Journal of Applied Signal Processing*, 2005(1):2101–2109, 2005.

[JAMS89] David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. Optimization by simulated annealing: an experimental evaluation. part i, graph partitioning. *Operations Research*, 37(6):865–892, 1989.

[Jäh01] B. Jähne. *Digitale Bildverarbeitung*. Springer, 5. edition, June 2001.

[Jol02] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[KH95] J.J. Kuch and T. S. Huang. Human computer interaction via the human hand: a hand model. In *In: 1994 Conference Record of the Twenty-Eight Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1252–1256, Washington, DC, 1995. IEEE Computer Society.

[Kje97] F. C. M. Kjeldsen. Visual interpretation for hand gestures as a practical interface modality. Technical report, Columbia University, 1997.

[KK05] S. S. Kim and S. Kweon. Automatic model-based 3d object recognition by combining feature matching with tracking. *Machine Vision and Applications*, 16(5):267–272, 2005.

[KLP04] Rick Kjeldsen, Anthony Levas, and Claudio Pinhanez. Dynamically reconfigurable vision-based user interfaces. *Machine Vision Applications*, 16(1):6–12, 2004.

[KM07] T. Kobayashi and N. Machida. Identifying hand gesture images by using genetic algorithms. In *MVA*, pages 240–243, 2007.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[Koh96] M. Kohler. Vision based remote control in intelligent home environments. In *University of Erlangen-Nuremberg/Germany*, pages 147–154, 1996.

[KU02] K.Dorfmüller-Ulhaas. *Optical Tracking - From User Motion To 3D Interaction*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2002.

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. Int'l Conf. on Computer Vision*, pages 259–269, 1987.

[Lau94] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[LC03] T. W. Lin and Y.-F. Chou. A comparative study of zernike moments. In *IEEE / WIC International Conference on Web Intelligence*, pages 516–519, 2003.

[Len87] R. Lenz. Linsenfehlerkorrigierte eichung von halbleiterkameras mit standarobjektiven für hochgenaue 3d-messungen in echtzeit. In *Mustererkennung 1987, 9. DAGM-Symposium*, pages 212–216, London, UK, 1987. Springer-Verlag.

[Lev44] K. Levenberg. A method for the solution of certain problems in least squares. In *Quarterly of Applied Mathematics*, volume 2, pages 164–168, 1944.

[LGS08] Yun Liu, Zhijie Gan, and Yu Sun. Static hand gesture recognition and its application based on support vector machines. In *SNPD*, pages 517–521. IEEE Computer Society, 2008.

[LH99] C. Lien and C. Huang. The model-based dynamic hand posture identification using genetic algorithm. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume I*, pages 706–713, London, UK, 1999. Springer-Verlag.

[LMS03] M. Li, M. Magnor, and H.-P. Seidel. Hardware-accelerated visual hull reconstruction and rendering. In *In Graphics Interface 2003*, pages 65–71, 2003.

[LS04] A. Licsar and T. Sziranyi. Hand gesture recognition in camera-projector system. In *In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science*, pages 81–91. Springer, 2004.

[MA07] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, 37:311–324, 2007.

[Mah36] P.C. Mahalanobis. On the generalised distance in statistics. In *Proceedings of the National Institute of Science of India*, volume 12, pages 49–55, 1936.

[Mar63] D. W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. volume 11, pages 431–441, 1963.

[MHK06] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[MN97] T. Miyasato and R. Nakatsu. Allowable delay between images and tactile information in a haptic interface. In *VSMM '97: Proceedings of the 1997 International Conference on Virtual Systems and MultiMedia*, page 84, Washington, DC, USA, 1997. IEEE Computer Society.

[MOC06] A. Malima, E. Ozgur, and M. Cetin. A fast algorithm for vision-based hand gesture recognition for robot control. *Signal Processing and Communications Applications, 2006 IEEE 14th*, pages 1–4, April 2006.

[MRR+53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[NHOD07] O. A. Nierop, A. Helm, K. J. Overbeeke, and T. J. Djajadiningrat. A natural human hand model. *Visual Computing*, 24(1):31–44, 2007.

[OH99] H. Ouhaddi and P. Horain. 3d hand gesture tracking by model registration. In *Proc. IWSNHC3DI '99*, pages 70–73, 1999.

[OSK02] K. Oka, Y. Sato, and H. Koike. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 429, Washington, DC, USA, 2002. IEEE Computer Society.

[OZR02] R. O'Hagan, A. Zelinsky, and S. Rougeaux. Visual gesture interfaces for virtual environments. *Interacting with Computers*, 14(3):231–250, 2002.

[PBKM07] G. A. Papakostas, Y. S. Boutalis, D. A. Karras, and B. G. Mertzios. A new class of zernike moments for computer vision applications. *Information Sciences*, 177(13):2802–2819, 2007.

[Pop07] R. Poppeld. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, 2007.

[PP07] I. Petras and I. Podlubny. State space description of national economies: the v4 countries. *Computational Statistics and Data Analysis*, 52:1223, 2007.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[Rad08] W. Radermacher, editor. *Statistisches Jahrbuch 2008 für die Bundesrepublik Deutschland*. Statistisches Bundesamt, Wiesbaden, September 2008. ISBN: 978-3-8246-0822-5.

[RNL06] T. Rhee, U. Neumann, and J. P. Lewis. Human hand modeling from surface anatomy. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 27–34, New York, NY, USA, 2006. ACM.

[RVB02] D. Reiners, G. Voß, and J. Behr. Opensg: Basic concepts. In *In 1. OpenSG Symposium OpenSG*, pages 200–2, 2002.

[RW08] G. D. Rey and K. F. Wender. *Neuronale Netze - Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Hans Huber Verlag, 2008.

[Sch06] B. Schwald. *Punktbasiertes 3D-Tracking starrer und dynamischer Modelle mit einem Stereokamerasystem für Mixed Reality*. Dissertation, Technische Universität Darmstadt, Fachbereich Informatik, GRIS, 2006.

[Sch08] W. Schneider. *Ergonomische Gestaltung von Benutzungsschnittstellen - Kommentar zur Grundsatznorm DIN EN ISO 9241-110*. Beuth-Verlag, Berlin, 2. edition, 2008.

[SEM00] S. Sun, M. Egerstedt, and C. F. Martin. Control theoretic smoothing splines. *IEEE Transactions on Automatic Control*, 45:2271–2279, 2000.

[SFJSL04] M. Schreiner, B. Frühmann, D. Jembrih-Simbürger, and R. Linke. X-rays in art and archaeology: An overview. *Powder Diffraction*, 19:3–+, 2004.

[SHB07] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, third edition, 2007.

[SK07] M. Schlattman and R. Klein. Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers. In *VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 39–42, New York, NY, USA, 2007. ACM.

[SLM+03] T. Starner, B. Leibe, D. Minnen, T. Westyn, A. Hurst, and J. Weeks. The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3d reconstruction for augmented desks. *Machine Vision and Applications*, 14(1):59–71, 2003.

[SMC01] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based 3d tracking of an articulated hand. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:310, 2001.

[SP95] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *In International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[SS01] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, December 2001.

[STTC03] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1063, Washington, DC, USA, 2003. IEEE Computer Society.

[STTC06] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1372–1384, 2006.

[SZ94] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Compututer Graphics and Applications*, 14(1):30–39, 1994.

[Tay92] C. J. Taylor. Active shape models - 'smart snakes. In *In British Machine Vision Conference*, pages 266–275. Springer-Verlag, 1992.

[TC88] C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analalysis and Machine Intelligence*, 10(4):496–513, 1988.

[THS+01] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. Vrpn: a device-independent, network-transparent vr peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55–61, New York, NY, USA, 2001. ACM.

[Tön05] K. D. Tönnies. *Grundlagen der Bildverarbeitung*. Pearson Studium, München, 2005.

[Tsa86] R. Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *Computer Vision Pattern Recognition*, 86:364–374, 1986.

[Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf cameras and lenses. RA-3(4):323–344, August 1987.

[TWYL05] X. Teng, B. Wu, W. Yu, and C. Liu. A hand gesture recognition system based on local linear embedding. *Journal of Visual Languages and Computing*, 16(5):442 – 454, 2005. Special issue section on Context and Emotion Aware Visual Interaction - Part I, pages 383- 441.

[UEB+06] J. Usabiaga, A. Erol, G. Bebis, R. Boyle, and X. Twombly. Global hand pose estimation by multiple camera ellipse tracking. In *ISVC (1)*, pages 122–132, 2006.

[UO99] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1473, 1999.

[Vap99] V. N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, November 1999.

[VCJT03] F. J. Valero-Cuevas, M. E. Johanson, and J. D. Towles. Towards a realistic biomechanical model of the thumb: the choice of kinematic description may be more critical than the solution method or the variability/uncertainty of musculoskeletal parameters. *Journal of Biomechanics*, 36(7):1019–1030, July 2003.

[VSA03] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *in Proc. Graphicon-2003*, pages 85–92, 2003.

[Wah90] G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[WF05] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, second edition, 2005. ISBN 0-12-088407-0.

[WHT03] L. Wang, W. M. Hu, and T. N. Tan. Recent developments in human motion analysis. 36(3):585–601, March 2003.

[WO03] A. Wilson and N. Oliver. Gwindows: robust stereo vision for gesture-based control of windows. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 211–218, New York, NY, USA, 2003. ACM.

[WSE+05] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler. A real-time hand gesture interface for medical visualization applications. *In the 10th Online World Conference on Soft Computing in Industrial Applications*, 2005.

[WWWR03] B. Watson, N. Walker, P. Woytiuk, and W. Ribarsky. Maintaining usability during 3d placement despite delay. In *VR '03: Proceedings of the IEEE Virtual Reality 2003*, page 133, Washington, DC, USA, 2003. IEEE Computer Society.

[Zer34] F. Zernike. Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode. *Physical 1*, pages 689–701, 1934.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[ZS06] N. A. Zaidi and N. M. Shiekh. Character recognition using statistical parameters. In *SSIP'06: Proceedings of the 6th WSEAS International Conference on Signal, Speech and Image Processing*, pages 58–61, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

Eigene Veröffentlichungen

[BCD+01] E. Boyle, T. Curran, A. Demiris, K. Klein, C. Garcia, C. Malerczyk, and C. Bouville. The creation of mpeg-4 content and its delivery over dvb infrastructure. In *Proceedings of the first Joint IEI/IEE Symposium on Telecommunications Systems Research*, Dublin, Ireland, November 2001. IEE The Institute of Electrical Engineering.

[CMNP08a] Z. Cernekova, C. Malerczyk, N. Nikolaidis, and I. Pitas. Interaction in edutainment applications using monocular posture tracking. In *Computer Graphics and Geometry Online Journal*, volume 10-2, pages 25–33, Moscow, Russia, 2008. Moscow Engineering Physics Institute (MEPhI). ISSN 1811-8992.

[CMNP08b] Z. Cernekova, C. Malerczyk, N. Nikolaidis, and I. Pitas. Single camera pointing gesture recognition for interaction in edutainment applications. In Karol Myszkowski, editor, *Spring Conference on Computer Graphics SCCG 2008. Conference Proceedings*, Bratislava, 2008. Comenius University, ACM SIGGRAPH.

[DGM+02] A. M. Demiris, C. Garcia, C. Malerczyk, K. Klein, K. Walczak, P. Kerbiriou, C. Bouville, and E. Reusens M. Traka, E. Boyle, J. Wingbermühle, and N. Ioannidis. Sprinting along with the olympic champions: Personalized, interactive broadcasting using mixed reality techniques and mpeg-4. In W. Abramowicz, editor, *Proceedings of Business Information Systems, BIS*, Poznan, Poland, 2002.

[DTR+01] A. Demiris, M. Traka, E. Reusens, K. Walczak, C. Garcia, K. Klein, C. Malerczyk, P. Kerbiriou, C. Bouville, E. Boyle, and N. Ioannidis. Enhanced sports broadcasting by means of augmented reality in mpeg-4. In Venetia Giagourta, editor, *Proceedings of the International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging*, pages 10–13, Mykonos, Greece, June 2001. European Commission / European Project INTERFACE IST.

[JKKM03] N. Calonego Junior, C. Kirner, T. Kirner, and C. Malerczyk. Interação com máquinas virtuais usando avalon. In *Proceedings of the VI Symposium on Virtual Reality - SVR 2003*, pages 199–209, Ribeirão Preto, 2003. UNICOC.

[KM01] K. Klein and C. Malerczyk. Creating a "personalised, immersive sports tv experience" via 3d reconstruction of moving athletes. In *Computer Graphics topics*, volume 13. Darmstadt, 2001. ISSN 0936-2770.

[KMWW02] K. Klein, C. Malerczyk, T. Wiebesiek, and J. Wingbermühle. Creating a "personalised, immersive sports tv experience" via 3d reconstruction of moving athletes. In Witold Abramowicz, editor, *Proceedings of Business Information Systems (BIS)*, Poznan, Poland, 2002. Best Paper Award of the INI-GraphicsNet 2003 Nominee.

[Mal04] C. Malerczyk. Interactive museum exhibit using pointing gesture recognition. In Vaclav Skala, editor, *Journal of the 12th International Conference in Central Europe on Computer Graphics, Visualization & Computer Vision 2004, WSCG*, volume Short Communications Volume II, pages 165–171, Plzen, Czech Republic, 2004. University of West Bohemia, European Association for Computer Graphics (Eurographics).

[Mal07] C. Malerczyk. 3d-reconstruction of soccer scenes. In *Proceedings of 3DTV-CON 2007 [CD-ROM] : Capture, Transmission and Display of 3D Video*. Institute of Electrical and Electronics Engineers (IEEE), 2007.

[Mal08a] C. Malerczyk. Dynamic gestural interaction with immersive environments. In Steve Cunningham, editor, *WSCG 2008. Communications Papers*, pages 161–166, Plzen, Check Republic, 2008. University of West Bohemia, European Association for Computer Graphics (Eurographics).

[Mal08b] C. Malerczyk. Gestural interaction using feature classification. In F. Perales and R. Fisher, editors, *Lecture Notes in Computer Science, Articulated Motion and Deformable Objects*, volume 5098/2008, pages 228–237, Berlin / Heidelberg, 2008. Springer Verlag. ISBN 978-3-540-70516-1.

[MDS05a] C. Malerczyk, P. Daehne, and M. Schnaider. Exploring digitized artworks by pointing posture recognition. In Mark Mudge, editor, *6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*, pages 113–119, Pisa, Italy, November 2005. ACM SIGGRAPH.

[MDS05b] C. Malerczyk, P. Daehne, and M. Schnaider. Pointing gesture-based interaction for museum exhibits. In *HCI International 2005. [Proceedings CD-ROM]*, volume 11th International Conference on Human Computer Interaction HCI, Mahwah, New Jersey, July 2005. Lawrence Erlbaum Associates, Inc.

[ME09] C. Malerczyk and T. Engelke. Intuitive interaction with vr applications using video-based gesture recognition. In *Proceedings of 2nd SEARIS Workshop at IEEE VR 2009*, Lafayette, Lousiana USA, 2009.

[MKW03] C. Malerczyk, K. Klein, and T. Wiebesiek. 3d reconstruction of sports events for digital tv. In *Journal of the 11-th International Conference in Central Europe on Computer Graphics, Visualization & Computer Vision, WSCG*, Plzen,

Czech Republic, 2003. University of West Bohemia, European Association for Computer Graphics (Eurographics).

[MS03a] C. Malerczyk and M. Schnaider. Video based interaction for arts and cultural heritage applications. In *Proceedings of the 1st International Workshop on Information and Communication Technologies (ICTs), Arts and Cultural Heritage*, Donostia-San Sebastian, Spain, May 2003.

[MS03b] C. Malerczyk and H. Seibert. 3d reconstruction of sports events for digital tv. In Vaclav Skala, editor, *Journal of WSCG International Conference in Central Europe on Computer Graphics, Visualization an Computer Vision (WSCG)*, volume 11 No. 2, pages 306–313, Plzen, Czech Republic, 2003. University of West Bohemia, European Association for Computer Graphics (Eurographics).

[MS04] C. Malerczyk and M. Schnaider. A mixed rality-supported interactive museum exhibit. In James Hemsley, editor, *Proceedings of EVA 2004 International Conference on Electronic Imaging & the Visual Arts*, pages 19.1–19.11, London, UK, July 2004. University College London The Institute of Archaeology, EVA Conferences International (ECI).

[MS07] C. Malerczyk and H. Seibert. 3d-reconstruction of soccer scenes. In Hamid R. Arabnia, editor, *Proceedings of the 2007 International Conference on Computer Graphics & Virtual Reality CGVR 2007*, pages 46–52, Las Vegas, 2007. CSREA Press.

[MSG03] C. Malerczyk, M. Schnaider, and T. Gleue. Video based interaction for a mixed reality kiosk system. In Julie A. Jacko, editor, *HCI International 2003. Proceedings of the 10th International Conference on Human-Computer Interaction*, volume 4: Inclusive Design in the Information Society, pages 1148–1152, Mahwah, New Jersey, June 2003. Lawrence Erlbaum Associates, Inc.

[MSS07] C. Malerczyk, H. Seibert, and B. Schwald. Usability evaluation report of edutainment demonstrator. Technical Report FP6-507609, SIMILAR, Network of Excellence, November 2007.

[SM02] B. Schwald and C. Malerczyk. Controlling virtual worlds using interaction spheres. In Creto Augusto Vidal, editor, *Proceedings of the 5th Symposium on Virtual Reality (SVR)*, pages 3–14, Fortaleza, CE, Brazil, 2002. Brazilian Computer Society (SBC).

[SM03] M. Schnaider and C. Malerczyk. Applying mixed reality technology to cultural applications. In *Proceedings of 2nd International Workshop on Information and Communication Technologies (ICTs), Arts and Cultural Heritage*, Athens, Greece, October 2003.

[SM04] M. Schnaider and C. Malerczyk. The augmented reality gallery application. In *Archaeology, Architecture & History ARCH - IT Symposium*, London, UK, July 2004.

[SMS01] V. Sá, C. Malerczyk, and M. Schnaider. Vision based interaction within a multimodal framework. In Joaquim Jorge, editor, *Proceedings of the 10th Portuguese Computer Graphics Meeting*, pages 61–67, Lisbon, Portugal, 2001.

[YMG09] S. Yoon, C. Malerczyk, and H. Graf. 3d skeleton extraction from volume data. In *WSCG 2009*, Plzen, Czech Republic, 2009. University of West Bohemia, European Association for Computer Graphics (Eurographics). ISSN 1213-6964.

Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Rostock, August 2009

Cornelius Malerczyk

Lebenslauf

Persönliche Daten:

Cornelius Malerczyk
Krifteler Strasse 14
65719 Hofheim am Taunus
geb. am 09. August 1968 in Kiel,
verheiratet, zwei Kinder

Schulausbildung:

1974 - 1978	Grundschule Kiel-Holtenau
1978 - 1988	Gymnasien in Kiel, Königstein und Hofheim am Taunus Abitur: Juni 1988 am Gymnasium der Main-Taunus-Schulen, Hofheim

Hochschulausbildung:

1996 - 1997	Studium der Mathematik mit Nebenfach Medizin an der Johann Wolfgang Goethe Universität Frankfurt
1997 - 2000	Studium der Mathematik an der Fachhochschule Gießen-Friedberg Hauptstudium Schwerpunkt: Mathematik und Angewandte Informatik Thema der Diplomarbeit: "Evaluierung von Active Shape Models zur Erkennung von Handgesten" Abschluss: Diplom-Mathematiker (FH) Auszeichnung der Diplomarbeit: Diplompreis der Fachhochschule Gießen-Friedberg 2000

Studienbegleitende Tätigkeiten:

1996 - 2000	Freiberufliche, statistische und pharmakokinetische Auswertungen klinischer Studien für die Hoechst AG / Aventis AG, Frankfurt am Main

1999 - 2000 Wissenschaftliche Hilfskraft in der Forschungsabteilung *Visual Computing* am Zentrum für Graphische Datenverarbeitung e.V. (ZGDV), Darmstadt

Berufstätigkeit:

1988 - 1996 Technischer Leiter des Theater auf Tour, Frankfurt am Main

2001 - 2008 Wissenschaftlicher Mitarbeiter in der Forschungsabteilung *Visual Computing* am Zentrum für Graphische Datenverarbeitung e.V. (ZGDV), Darmstadt
Tätigkeiten:
Mitarbeit (Konzeption und Implementierung) an diversen nationalen und EU-Forschungsprojekten
Technische Projektleitung von nationalen und europäischen Forschungsprojekten
Administrative Projektleitung im EU-Projekt Similar (Leitung der Arbeitsgruppe *Edutainment and Learning Assistance*)
Betreuung von Diplom- und Bachelorarbeiten
Betreuung von studentischen Hilfskräften und Praktikanten

seit 2008 Lehrkraft für besondere Aufgaben am Fachbereich Mathematik, Naturwissenschaften und Datenverarbeitung (MND) an der Fachhochschule Gießen-Friedberg (75%-Stelle)

seit 2009 Wissenschaftlicher Mitarbeiter in der Forschungsabteilung *Virtuelle und Erweiterte Realität* am Fraunhofer Institut für Graphische Datenverarbeitung, Darmstadt (25%-Stelle)

Weitere Tätigkeiten:

2001 - 2008 Lehrbeauftragter für Graphische Datenverarbeitung an der Fachhochschule Gießen-Friedberg im Fachbereich Mathematik, Naturwissenschaften und Datenverarbeitung (MND) mit 4-6 Semesterwochenstunden

Thesen

1. Interaktion durch videobasierte Handgestenerkennung fördert die intuitive Bedienbarkeit einer Computeranwendung und erleichtert somit die Schnittstelle zwischen Mensch und Computer. Die Verwendung von Kameras unterstützt dabei den intuitiven Charakter der Eingabe, da der Nutzer kein technisches Gerät bedienen muss, sondern ausschließlich mit seinen eigenen Händen agieren kann.

2. Intuitive Interaktion fordert eine Gerätelosigkeit des Systems für den Anwender. Nur wenn der Nutzer keine technischen Hilfsmittel in die Hand nehmen muss, ist intuitive Interaktion möglich. Daneben ist es außerdem wichtig, dass alle technischen Geräte wie beispielsweise die aufnehmenden Kameras selbst, außer Sichtweite des Anwenders angebracht werden können und der anzeigende Bildschirm das einzige für den Nutzer sichtbare technische Gerät ist.

3. Für eine intuitive Bedienbarkeit eines Systems ist es erforderlich, dass ein neuer Anwender ohne oder mit nur minimalem Lernaufwand mit der Interaktion starten kann. Insbesondere müssen die verwendeten Verfahren in der Lage sein, korrekt auf individuelle Unterschiede in der Verwendung von Handgesten automatisch reagieren zu können.

4. Verfahren zur intuitiven Interaktion müssen echtzeitfähig und damit in der Lage sein, Bildwiederholraten von 20 Bildern pro Sekunde oder mehr zu erreichen, um eine flüssig wirkende Interaktion zu ermöglichen. Außerdem muss die Verzögerung zwischen realem Ereignis der Hand und der daraus resultierenden Reaktion des Systems sehr kurz sein, da Reaktionszeiten von mehr als 200 Millisekunden von den meisten Menschen als unnatürlich empfunden werden.

5. Ausgehend vom derzeitigen Stand der Forschung und Technik ist es möglich, Verfahren zu entwickeln, die eine intuitive Interaktion zwischen Mensch und Computer ausschließlich durch die Verwendung von Handgesten ermöglichen und dabei die Forderungen nach geräteloser Bedienung, Reduktion des Trainingsaufwandes und Echtzeitfähigkeit erfüllen.

6. Die Verwendung eines kalibrierten Stereokamerasystems ermöglicht es, auf Bildbasis ermittelte Parameter der Geste präzise zu einem Punkt im Raum zu rekonstruieren und erleichtert somit die Erkennung und Verfolgung von Handgesten im dreidimensionalen Raum.

7. Die Kombination aus dem nichtlinearen Optimierungsverfahrens *Simulated Annealing*, einer Anpassung von aktiven Konturen auf Bildbasis und deren 3D-Rekonstruktion ermöglicht eine robuste Erkennung und Verfolgung einer Zeigegeste. Das Verfahren der *Interaktion durch Rekonstruktion von aktiven Formen* ermittelt dabei alle für die Interaktion notwendigen Parameter im dreidimensionalen Raum in Echtzeit.

8. Das Verfahren der *Interaktion durch Punktprojektion* ermöglicht es, auf unterschiedliche Ausprägungen einer Zeigegeste durch verschiedene Anwender korrekt zu reagieren. Durch eine Reduktion der Information über die Zeigegeste auf die Spitze des Zeigefingers benötigt das Verfahren keine modellbildende Trainingsphase und ermöglicht es jedem neuen Anwender, sofort und intuitiv zu interagieren.

9. Die von vielen Anwendungen geforderte Möglichkeit, lokal Selektionsereignisse auslösen zu können, kann mittels einer Analyse von vordefinierten Ereignisregionen und der Analyse der Positionsgeschwindigkeit der Interaktionspunkte erreicht werden und versetzt damit den Anwender in der Lage, Elemente einer grafischen Benutzerschnittstelle ausschließlich durch die Verwendung der Zeigegestik zu bedienen.

10. Das auf der Einordnung von Merkmalsvektoren durch einen Naïven Bayes-Klassifikator beruhende Verfahren der *Interaktion durch merkmalbasierte Gesten-Klassifikation* ermöglicht eine Unterscheidung von verschiedenen statischen Handgesten in Echtzeit. Damit schafft das Verfahren beispielsweise durch die Unterscheidung einer geschlossenen und offenen Hand die Möglichkeit, virtuelle Objekte im Raum zu bewegen.

11. Das auf einer Analyse von Pseudo-Zernike-Momenten von Handsilhouetten beruhende Verfahren der *Interaktion durch Momenten-Analyse* rekonstruiert den dynamischen Prozess einer Handgeste und ermöglicht so eine präzise und intuitive Interaktion beispielsweise durch Zugreifen und Loslassen eines virtuellen dreidimensionalen Objektes mit dem Pinzettengriff.

12. Je nach Anwendungstyp und den daraus resultierenden Anforderungen an die Eingabemodalitäten des Systems können die unterschiedlichen Verfahren dieser Arbeit variabel eingesetzt werden und bestehende Verfahren oder vorhandene Eingabegeräte ersetzen.

Zusammenfassung

Videobasierte Handgestenerkennung ist seit Langem ein intensives Thema der wissenschaftlichen Forschung. Dahinter steht die Vision, die Interaktion zwischen Mensch und Computer losgelöst von klassischen Eingabegeräten wie Computermaus und Tastatur zu realisieren. Das in dieser Arbeit angestrebte Ziel leitet sich aus der Fragestellung und Vision der videobasierten Handgestenerkennung ab: Es sollen Verfahren entwickelt werden, die unter Verwendung von Techniken der Bildverarbeitung und der *Computer Vision* eine robuste und fehlerarme Erkennung menschlicher Handgesten realisieren und so die Bedienung eines Computersystems auch für technisch unerfahrene Anwender nutzbar machen. Dazu stellt diese Arbeit zunächst drei elementare Anforderungen an die zu entwickelnden Verfahren: Gerätelosigkeit des Systems für den Anwender, Bedienbarkeit ohne oder nur mit minimaler Trainingsaufwand und Echtzeitfähigkeit des Systems in Bezug auf Bildwiederholraten und Systemreaktionszeiten. Diese Arbeit stellt ausgehend vom derzeitigen Stand der Forschung und Technik die Hypothese auf, dass es möglich ist, Verfahren zu entwickeln, die eine intuitive Interaktion zwischen Mensch und Computer ausschließlich durch die Verwendung von Handgesten ermöglichen. Dabei werden zwei der wichtigsten Probleme der videobasierten Gestenerkennung adressiert: Zum einen müssen die Verfahren in der Lage sein, die Hand des Anwenders vollständig automatisch zu erkennen (*Initialsuche der Hand*). Zum anderen müssen die Verfahren die zur Interaktion notwendigen Informationen anhand der Bildeigenschaften gewinnen können (*Merkmalsextraktion und Klassifikation*). Als technische Grundlage dient für die Verfahren ein kalibriertes Stereokamerasystem, mit dem es ermöglicht wird, auf Bildbasis ermittelte Parameter der Geste präzise zu einem Punkt im Raum zu rekonstruieren. In dieser Arbeit werden vier verschiedene Verfahren entwickelt, die abhängig von den Aufgaben, für die sie eingesetzt werden, unterschiedliche Arten der Interaktion durch videobasierte Handgestenerkennung realisieren: Die Verfahren der *Rekonstruktion von aktiven Formen* und der *Interaktion durch Punktprojektion* sind in der Lage, insbesondere die Zeigegeste der menschlichen Hand als wichtigste statische Geste zur Interaktion zu erkennen, zu verfolgen und die Zeigerichtung der Geste in Echtzeit zu bestimmen. Das Verfahren der *merkmalbasierte Gesten-Klassifikation* ermöglicht neben der Berechnung der Position der Handgeste im dreidimensionalen Raum auch eine Unterscheidung verschiedener statischer Handgesten des Anwenders. Mit dem Verfahren der *Momenten-Analyse* können dynamische Prozesse einer Geste in Echtzeit bestimmt und für die Interaktion nutzbar gemacht werden. Mit der Entwicklung der vier Verfahren leistet diese Arbeit einen Beitrag zur Vision einer gestenbasierten Schnittstelle zwischen Mensch und Computer und ermöglicht durch den videobasierten Ansatz eine für den Anwender gerätelose Interaktion mit Computersystemen und deren Anwendungen.

Abstract

Video based hand gesture recognition is an extensive area of scientific research since many years. The vision behind this research is to realise a new kind of interaction between humans and computer beyond the classical input devices such as computer mouse and keyboard. The aim of this thesis arises directly from this vision: New algorithms are to be developed using image processing and computer vision techniques, which enable a robust and accurate recognition of human hand gestures and allow interaction with computer systems even for technically unversed users. Therefore, this thesis makes three elementary demands of the algorithms to be developed: Usability of the system without any device hold or worn by the user, operability without or only with a minimum of training expenditure and real time capabilities of the system with respect to frame rates and system reaction times. Based on the current state of the art it is hypothesised that algorithms can be developed that allow intuitive human computer interaction using video based gesture recognition. Two basic problems of video based recognition and tracking are addressed: The algorithms have to be able to detect the human hand completely automatically (initial registration) and they have to extract all information necessary for the interaction (feature extraction and classification). The technical basis of all algorithms is a calibrated stereo camera system, which allows a precise 3d reconstruction of corresponding image features. In this thesis four different algorithms are developed that can be used for intuitive interaction purposes depending on the demands and needs of different scenario applications: The algorithms of *reconstructing active shapes* and *interaction through point projection* have the ability to recognise and track the human pointing gesture and to determine the pointing direction of the user's hand in real time. The algorithm of *feature based gesture classification* allows, in addition to the 3d position of the user's hand, the differentiation between various static gestures. Using the algorithm of an *analysis of moments* it is possible to reconstruct dynamic gestures such as a precise grabbing and releasing of objects in 3d space. With the development of these four algorithms this thesis makes a contribution to the vision of a gesture-based interface between humans and computer and enables an intuitive and deviceless interaction with computer systems and their applications.