Entwicklung einer prototypischen Gestenerkennung in Echtzeit unter Verwendung einer IR-Tiefenkamera

Diplomarbeit

Vorgelegt von
Pia Breuer

UNIVERSITÄT KOBLENZ · LANDAU

Institut für Computervisualistik
Arbeitsgruppe Computergraphik

Fraunhofer Institut Medienkommunikation

Betreuer: Dipl.-Phy. Christian Eckes
Prüfer: Prof. Dr.-Ing. Stefan Müller

Dezember 2005

Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit selbstständig und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Diese Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Bad Neuenahr-Ahrweiler, den 19. Dezember 2005

Pia Breuer

Inhaltsverzeichnis

1 Einleitung **1**

2 Kamera **3**

2.1 SwissRanger™2 3

2.2 Technik 3

2.2.1 Distanzbestimmung mittels time-of-flight Prinzip . . . 4

2.3 Kalibrierung 6

2.4 Ausblick 8

3 Literatur Review **10**

3.1 Übersicht der vorherigen Arbeiten im Bereich Gestenerkennung 10

3.2 Tabellarischer Überblick 11

4 Konzept **26**

5 Umsetzung **27**

5.1 Kalibrierung 27

5.2 Tiefensegmentierung 31

5.3 Grobschätzung mittels Hauptkomponentenanalyse 34

5.3.1 Hauptkomponentenanalyse 34

5.3.2 Anwendung im System 35

5.4 Visualisierung der Ergebnisse 35

5.4.1 Darstellung als Punktwolke 35

5.4.2 Darstellung als Mesh 36

5.4.3 Auswertung 37

5.5 Modellbasiertes Feinmatching 38

6 Ergebnisse **43**

7 Fazit und Ausblick **48**

A Berechnungen **52**

B Visualisierungen **55**

Abbildungsverzeichnis

1 **Kamera - SR2** 3
2 **modulierte Signale** 5
3 **Aufbau der lock-in Pixel** 6
4 **Versuchsaufbau bei 1,0m Entfernung** 27
5 **Versuchsaufbau bei 1,5m Entfernung** 28
6 **Visualisierung der Messwerte bei 1,0m Entfernung** . . 29
7 **Visualisierung der Messwerte bei 1,0m Entfernung** . . 29
8 **Modell zur Koordinatenermittlung [KI05a]** 30
9 **Strahlensatz** 30
10 **Kamera mit Maßband und Lot** 32
11 **Hand mit Lot für die ROI-Ermittlung** 33
12 **Auswahl der Visualisierungen mit dem SR2-Tool** . . . 34
13 **Punktwolke mit Hauptachsen** 36
14 **Visualisierung als Mesh** 37
15 **Handmodell** 39
16 **Verarbeitungsfolge** 42
17 **Bildsequenz 1 (Frontalansicht)** 43
18 **Bildsequenz 1 (seitliche Ansicht)** 44
19 **Bildsequenz 2** 44
20 **Skelett (Zeichnung)** 50
21 **Szenengraph der artikulierten Arm-Hand-Geometrie** . 50
22 **Skelett im Programm** 51
23 **Messwerte in MatLab visualisiert** 55
24 **versch. Tiefen - mit dem SR2-Tool aufgenommen** . . . 56

1 Einleitung

Heutzutage spielt die Mensch-Technik-Interaktion in vielen Bereichen eine große Rolle. Dabei besteht selbstverständlich auch der Wunsch nach möglichst einfachen Interaktionsformen. Die Interaktion durch (natürliche) Gesten ist eine dieser Formen. Dazu ist es notwendig, Hand- bzw. Ganzkörperbewegungen zu erkennen, um diese dann für die Mensch-Maschine-Interaktion zu verwenden, wobei eine Ganzkörperrekonstruktion, wegen der geringeren Anzahl an möglichen Freiheitsgraden, einfacher durchzuführen ist, als eine Handrekonstruktion.

Die Natürlichkeit ist allgemein sehr wichtig. Je weniger Hilfsmittel für eine Interaktion vonnöten sind, desto höher ist die Akzeptanz dieser Art von Kommunikation gegenüber. Es sollte möglichst das Prinzip „come as you are“ gelten, was bedeutet, dass ganz auf Hilfsmittel wie Datenhandschuhe oder aufgeklebte Marker verzichtet wird, und einzig die Person an sich interagiert.
In kommerziellen Anwendungen wie dem EyeToy für die PS2 wird diese Art der Interaktion auch schon in einfachster Weise praktiziert. Allein durch die Bewegung des eigenen Körpers werden dort beispielsweise virtuelle Fenster geputzt, geboxt oder ein Ball hüpfend über die Bildfläche bewegt (vgl. [Eye05]).
Die Erkennung und Wiedergabe von Zeichensprache ist auch ein Gebiet, das immer wieder als Hintergrund für Forschungen in diesem Bereich genannt wird. Will man Zeichensprache hör- oder lesbar machen, so muss man die gezeigten Gesten erkennen und in gesprochene bzw. geschriebene Sprache umsetzen. Umgekehrt könnten auch Gesten rekonstruiert werden, um Gesprochenes sichtbar zu machen.

Im Rahmen dieser Arbeit soll deshalb nun eine prototypische Gestenerkenneung in Echtzeit konzipiert und umgesetzt werden. Hierbei geht es jedoch nicht darum, spezifische Gesten als solche zu identifizieren, sondern das Hauptziel der hier entwickelten Gestenerkennung ist es, die Hand mit ihren Bewegungen für die Interaktion zwischen Mensch und Technik nutzbar zu machen. Die Entscheidung zur Verwendung von Tiefendaten fiel, da sich mit Hilfe der vorhandenen Miniatur-TOF-Kamera Tiefenkarten in Echtzeit (30fps) generieren lassen. Außerdem ist kein spezieller Hintergrund für die Segmentierung nötig, wodurch die Ausführbarkeit der Anwendung ortsunabhängig ist. In diesem Zusammenhang geht es nun darum herauszufinden, ob und inwieweit Tiefenbilder für die Gestenerkennung verwendbar sind. Ein weiterer Augenmerk liegt darauf zu erkennen, wo dabei die Grenzen liegen und ob man eventuell andere Sensoren zur Unterstützung einsetzen kann oder sogar muss.

Die weitere Gliederung dieser Ausarbeitung ergibt sich daher wie folgt: Kapitel 2 und 3 beschäftigen sich mit den Grundlagen der Arbeit. Dabei be-

schreibt Kapitel 2 die genutzte Kamera mit ihrer Funktionsweise, den dazugehörigen technischen Daten und den Kalibrierungsmöglichkeiten und bietet außerdem einen Ausblick auf das neueste Modell. Kapitel 3 liefert einen Überblick über die vorherigen, im Bereich Gestenerkennung veröffentlichten Arbeiten, woraus sich dann das in Kapitel 4 aufgeführte Konzept ergibt. Mit dessen Umsetzung befasst sich Kapitel 5. Die einzelnen im Konzept aufgezählten Punkte, sowie ihre Realisierung werden dort eingehend behandelt. Kapitel 6 präsentiert die Ergebnisse und in Kapitel 7 findet das ganze mit einem Fazit und dem Ausblick auf Ausbau- und Verbesserungsmöglichkeiten seinen Abschluss. Im Anhang befinden sich noch dazu Berechnungen und weitere Visualisierungen wieder.

2 Kamera

2.1 SwissRanger™2

Die Miniatur-Tiefenkamera SwissRanger SR-2 entwickelt von CSEM ist ein optisches Bildsystem, das hochaufgelöste 3D-Bilddaten in Echtzeit bietet. Sie basiert auf dem time-of-flight-Prinzip und arbeitet mit einer modulierten Infrarot-Lichtquelle. Im Gegensatz zu herkömmlichen Kameras bestimmt der SR-2 nicht nur die lokale Helligkeit der Szene, sondern eine komplette Tiefenkarte. Sowohl diese Tiefenbilder als auch die konventionellen Grauwert-Bilder werden über ein USB 2.0 Interface geliefert. [CSE05]

Abbildung 1: **Kamera - SR2**

2.2 Technik

Die Kamera besteht aus den zwei Hauptkomponenten Lichtsender und Lichtempfänger. Die Beleuchtungseinheit umfasst 48 LEDs. Die Sensoreinheit beinhaltet die Kontrollelektronik. Die folgende Tabelle zeigt einen Überblick der technischen Daten (entnommen aus [CSE05]):

Anzahl der Pixel	max. 124x160*
Tiefenauflösung	bis zu 5mm
Wellenlänge	870nm
Beleuchtungsstärke	800mW (optisch)
max. Entfernung	7,5m
Frame-Rate	bis zu 30fps**
diag. Sichtfeld	±30° (±21° x ±23°)
Interface	USB 2.0
Verbindung	Mini USB Typ-B
Stromversorgung	+12V / 1.5A DC
Stromverbrauch	18W max.
Linse	f = 8mm, F/# = 1.4, M 12x0.5
Ausmaße	135(B)x45(H)x32(T)
Gewicht	0.2kg

* je nach ROI-Einstellung
** je nach Kamera-Einstellung

2.2.1 Distanzbestimmung mittels time-of-flight Prinzip

Es wird ein RF (radiofrequency)-moduliertes, optisches Strahlungsfeld im Infrarotbereich ausgesendet (20MHz), welches von der Szene diffus reflektiert und von der Kamera erkannt wird. Dies basiert auf dem time-of-flight Prinzip (TOF), das besagt, dass die benötigte Zeit, die das Licht braucht, um zum Objekt und zurück zu gelangen, der Distanz D entspricht

$$TOF = \frac{2D}{c} \quad \text{mit Lichtgeschwindigkeit } c = 3 \cdot 10^8 m/s$$

Bei dieser Kamera werden jedoch keine Impulse ausgesendet und die Zeit gemessen, bis sie zurückkehren, sondern eine kontinuierlich modulierte Lichtwelle, bei der dann die Phasenverschiebung zwischen ausgesendeter und reflektierter Welle ermittelt wird.

Abbildung 2 zeigt den Zusammenhang zwischen ausgesendetem Signal P_e und empfangenem Signal P_r. Dafür lässt sich Folgendes festhalten:
Die Phasenverschiebung φ entsteht durch die Entfernung, die das Licht vom Sender zur Szene und zurück überwinden muss. Daraus wird die Distanz ermittelt.
Das ankommende Signal ist durch einen Offset (P_B), der vom Hintergrundlicht und einer nicht perfekten Modulation herrührt, verschoben. Dieser Beschreibt die Intensität.
Die Amplitude ist um den Faktor k verkleinert, was durch optische Verluste verursacht wird. Ihre Höhe bestimmt die „signal-to-background“-Rate und

infolgedessen die erreichbare Auflösung. Stellt P_A die Amplitude des ausgesendeten Signals dar, so entspricht $k \cdot P_A$ der Amplitude des empfangenen Signals (vgl. [BOL+05]).

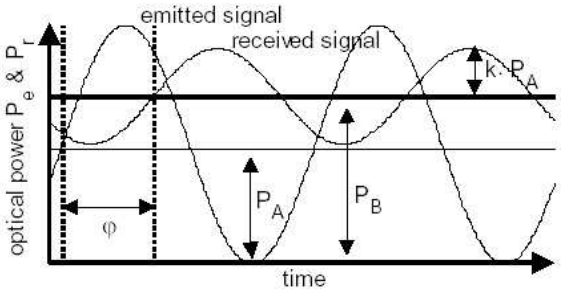

Abbildung 2: **modulierte Signale** entnommen aus [BOL+05]

Die Demodulation wird mit Hilfe von Sampling durchgeführt. Das ankommende Signal wird viermal pro Modulationsperiode (jeder Sample um 90° verschoben) abgetastet, wodurch das Signal eindeutig rekonstruiert werden kann. Dabei entspricht jede dieser Abtastungen der Integration der photogenerierten Elektronen über den entsprechenden Bruchteil der Modulationsperiode. Um das Signal-zu-Rausch-Verhältnis zu reduzieren, werden über mehrere Perioden die jeweiligen Werte der Abtastungen aufsummiert. Bei einer Modulationsfrequenz von 20MHz und einer Integrationszeit von 10ms wird folglich über 200.000 Modulationsperioden integriert. Basierend auf den vier Samples $\tau_0 = 0°$, $\tau_1 = 90°$, $\tau_2 = 180°$ und $\tau_3 = 270°$ ergeben sich somit die folgenden Formeln für die einzelnen Parameter:

- *Phasen- bzw. Laufzeitverschiebung*

$$\varphi = \arctan\left(\frac{c(\tau_3) - c(\tau_1)}{c(\tau_0) - c(\tau_2)}\right)$$

- *Offset*

$$B = \frac{c(\tau_0) + c(\tau_1) + c(\tau_2) + c(\tau_3)}{4}$$

- *Amplitude*

$$A = \frac{\sqrt{[c(\tau_3) - c(\tau_1)]^2 + [c(\tau_0) - c(\tau_2)]^2}}{2}$$

- *Distanz*

$$D = L \cdot \frac{\varphi}{2 \cdot \pi} \quad mit \quad L = \frac{c}{2 \cdot f_m}$$

L: eindeutiger Distanzbereich
c: Lichtgeschwindigkeit
f_m: (radiofrequency) RF Modulationsfrequenz

Dieses Verfahren wird wegen der viermaligen Abtastung pro Periode allgemein als „four buckets“- oder auch „four steps“-Methode bezeichnet. [OMLR+03]

Da jeder Pixel diese Demodulation für sich durchführt und außerdem einzeln ansteuerbar ist, werden sie auch „lock-in Pixel“ genannt. Die Integration wird für jedes Sample über die halbe Modulationsperiode vorgenommen. Den Aufbau der Pixel und die Funktionsweise des Abtastungsprozesses verdeutlicht Abbildung 3.

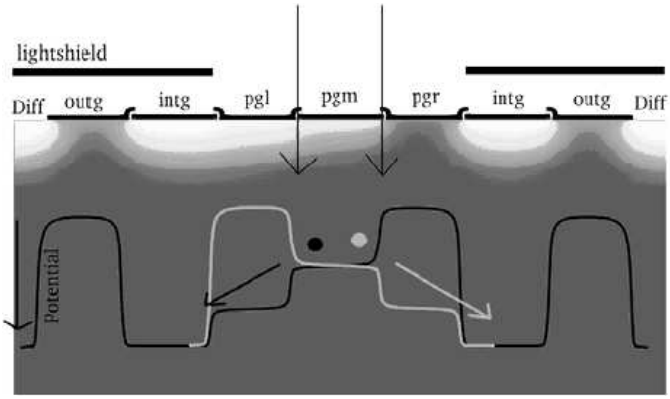

Abbildung 3: **Aufbau der lock-in Pixel** entnommen aus [BOL+05]

Die Pixel bestehen aus überlappenden poly-silikon Gattern und liefern zwei Abtastungen des ankommenden Signals zur gleichen Zeit. Das Signal wird unmittelbar über den „photo-gates“ (Lichtgatter: links PGL, mitte PGM und rechts PGR) erst in Elektronenpaare umgewandelt. Durch eine angelegte Kontrollspannung werden die Ladungen im Lichtgatter entsprechend der charakteristischen Potenzialverteilung nach rechts oder links abgelenkt und dort im Integrationsgatter (intg) zwischengespeichert. Zwischen rechtem und linkem Lichtgatter wird entsprechend der Modulationsfrequenz umgeschaltet, so dass die Ladungspakete an den Integrationsgattern jeweils einem Sample entsprechen. Durch Verminderung der dortigen Spannung gelangen die Ladungen über das „out gate“ (outg) schließlich zur Diffusionsmessschicht (diff), wo die Ladungsmenge bestimmt wird [BOL+05].

2.3 Kalibrierung

Um den Messwerten möglichst genaue, reale Entfernungen zuordnen zu können, und daraus dann 3D-Positionen zu bestimmen, ist es nötig, die Kamera zu kalibrieren. Folgende Faktoren sind dabei zu beachten: Der Einfallswinkel des Lichts am Objekt, die interne Temperatur (Selbstaufheizung) der Kamera, die äußere Temperatur, die Intensität des reflektierten Strahls

und die Integrationszeit. Der Einfallswinkel ist zu berücksichtigen, da trotz diffuser Reflexion für den Hauptteil der zurückgestreuten Strahlung gilt: Einfallswinkel $\approx$ Ausfallswinkel ($alpha_{in} \approx alpha_{out}$). Ist ($alpha_{in} \ll 90°$) oder ($alpha_{in} \gg 90°$), so wird der Großteil der Strahlen nicht dorthin zurückgestreut wo er ausgesendet wurde, sondern in eine völlig andere Richtung. Interne sowie äußere Temperatur verursachen einen Drift der Werte. Durch die Selbstaufheizung vergrößern sich die Messwerte und die Szene scheint sich insgesamt zu entfernen. Bei der äußeren Temperatur lässt sich sogar eine Verschiebung von 8mm/°C erkennen [KI05b]. Die Intensität des reflektierten Strahls und die Integrationszeit beeinflussen die Reichweite der Kamera. Je weiter ein Objekt entfernt ist, desto geringer ist die Intensität des an ihm reflektierten Stahls, d. h. es kommen weniger Photonen beim Sensor an, und wenn die Anzahl zu gering ist, kann dort die Tiefe nicht mehr exakt gemessen werden. Die Integrationszeit entspricht der Belichtungszeit. Reduziert man diese, so wird früher ausgelesen wieviele Photonen registriert wurden. Da von weiter entfernten Objekten ohnehin schon weniger Photonen reflektiert werden, wird die Anzahl derer, die gemessen werden, durch das frühere Auslesen noch einmal verringert. Demnach kann man festhalten: je kürzer die Integrationszeit, desto geringer die maximal messbare Tiefe.

Kahlmann und Ingensand haben sich mit den Möglichkeiten dieser Kalibrierung eingehend beschäftigt (vgl. [KI05a] und [KI05b]). Dabei stellte sich heraus, dass sich die oben genannten Einflüsse auf die Genauigkeit der Messwerte nicht unmittelbar durch die Anwendung einer einzigen Kalibrierungsmatrix reduzieren lassen. Bei Gut [Gut04] wird sogar von der Notwendigkeit einer „multiplen Kalibrierung“ gesprochen. Bei seinen Untersuchungen beobachtete dieser zudem eine weitere Ungenauigkeit der Messwerte, die sich nicht einfach „weg-kalibrieren“ lässt, den sogenannten Kometenschweif („tail of a comet„). Dabei strahlen die gemessenen Tiefenwerte an den Objekträndern nach hinten aus und in der Seitenansicht ähnelt dies dann dem Schweif eines Kometen. Zur Entstehung dieses Effekts kommt es, da jeder Pixel des Chips über einen kurzen Zeitraum belichtet wird und dabei auch alle mehrmals reflektierten Infrarotstrahlen berücksichtigt und für die endgültige Distanzmessung gemittelt werden. Außerdem muss jeder Pixel wegen der geringen Auflösung über das Objekt und den Hintergrund am Objektrand mitteln, da er sowohl Photonen vom Objekt als auch vom Hintergrund aufsammelt. Der resultierende Fehler könnte nur vermieden werden, wenn ausschließlich das erste ankommende Signal ausgewertet würde. Diese Verbesserung wäre allerdings nicht durch eine Kalibrierung zu erreichen, sondern müsste schon „innerhalb der Kamera“ vorgenommen werden.

Die oben erwähnte Integrationszeit sowie einige andere Parameter lassen sich jedoch über ein USB 2.0 Interface je nach Einsatzgebiet individuell anpassen. Die folgende Tabelle enthält eine Auswahl der einstellbaren Kameraparameter mit ihren Standardeinstellungen:

Amplitude Threshold LSB	96
Amplitude Threshold MSB	0
LongIntegrationTime	60
ShortIntegrationTime	16
Intensity Saturation LSB	0
Intensity Saturation MSB	252
Start Row	0
Start Column	0
End Row	159
End Column	123

Amplitude Threshold LSB und *Amplitude Threshold MSB* legen den Amplitudengrenzwert fest (MSB · 256 + LSB). Dieser Wert funktioniert wie ein Filter. Die Amplitude repräsentiert den Betrag des reflektierten Lichts. Die Messungen, bei denen die Amplitude kleiner als der Grenzwert ist, werden in den weiteren Berechnungen nicht berücksichtigt und für diese Pixel wird 0 als Wert zurückgegeben. Durch setzen des Amplitudenfilters kann eine gewisse Beständigkeit der Messwerte garantiert werden. Die Definition der Belichtungszeit geschieht über *LongIntegrationTime* (default: 60[*256μs] = 15,36ms) und *ShortIntegrationTime* (default: 16[*256μs] = 4,096 ms). Ihr Übergang wird mittels der *Intensity Saturation* definiert. Geht der Offset für die Messung mit der längeren Belichtungszeit über den Schwellwert hinaus, so wird an ihrer Stelle die Messung der kürzeren Belichtungszeit berücksichtigt. Schließlich lässt sich über *Start Row*, *End Row*, *Start Column* und *End Column* eine Region of Interest innerhalb des Sichtfeldes wählen, wobei ($0 \leq$ Start Row $<$ End Row < 160) und ($0 \leq$ Start Column $<$ End Column < 124) gelten müssen.

Wie diese Parametereinstellungen vorteilhaft genutzt werden können, sowie die Bestimmung der realen Entfernungen und der daraus resultierenden 3D-Positionen werden in 5.1 genau erläutert.

2.4 Ausblick

Eine Weiterentwicklung der hier genutzten Kamera ist der SwissRanger SR3000. Dieser wird Ende des Jahres auf den Markt kommen. Er enthält drei wesentliche Neuerungen (vgl. [BOL$^+$05]):

1. *Unterdrückung des Hintergrundlichts:* Die Formel für die Phasenverschiebung (vgl Kapitel 2.2.1) zeigt, dass die Distanz auf den Differenzen der Abtastungen beruht, d. h. auf der Auswertung der durch das Signal generierten Elektronen. Alle vom Hintergrundlicht zusätzlich erzeugten Elektronen werden in selber Anzahl zu allen Samples dazu

addiert. Zieht man die „gesammelten“ Elektronen der zweiten zwei Abtastungen von denen der ersten ab, so werden die beiden Differenzen $\tau_0 - \tau_1$ und $\tau_2 - \tau_3$ vom jeweiligen Pixel zurückgeliefert. Eine Sättigung, durch Hintergrundlicht verursacht, kommt somit nicht länger vor.

2. *Integrationszeit pro Pixel einstellbar:* Wird die Ausgabekapazität überschritten, so wird die Integration der Ladungen sofort gestoppt und für weitere Messungen selbstständig angepasst.

3. *höhere Auflösung:* Die neue Auflösung beträgt 176 x 144 Pixel.

3 Literatur Review

3.1 Übersicht der vorherigen Arbeiten im Bereich Gestenerkennung

Die Arbeiten, mit denen sich das folgende Kapitel befasst, stellen einen Überblick des neuesten Stands der Forschung im Bereich Gestenerkennung dar. Sie wurden alle - bis auf wenige Ausnahmen - innerhalb der letzten fünf Jahre veröffentlicht.

Der Hauptteil beschäftigt sich mit der Erkennung von Gesten anhand von 2-dimensionalen Bildern, wobei vorwiegend nach der Hautfarbe gesucht oder ein bekannter Hintergrund subtrahiert wird. Ein zweiter großer Bereich befasst sich mit der Erkennung mit Hilfe eines Datenhandschuhs. Die eigentliche Erkennung erfolgt dann (meist) durch Vergleiche mit bestehenden bzw. gelernten Datensätzen oder Handmodellen. Die Verwendung von 3D-Daten und auch der Einsatz von Infrarot-Kameras, wie bei dieser Arbeit, sind noch nicht sehr verbreitet.

Bei Starner et al. [SLM+03] werden zwar 3D-Positionen ermittelt, jedoch mit erheblichem Aufwand. Mehrere Lichtquellen und zwei Kameras sind vonnöten, um die Ebene, in der sich der Arm befindet, sowie die Position der Fingerspitzen und den Eintrittspunkt des Arms ins Bild zu erkennen, und daraus die 3D-Positionen von Oberarm und Fingerspitzen zu bestimmen. Theobalt et al. [TAH+04] berechnen 3D-Positionen mit Hilfe von vier Kamerabildern und der Epipolargeometrie. Bei ihrem Ansatz werden der Hintergrund und die Person ganz in Schwarz gehüllt und nur die Hand mit leuchtenden, farbigen Klebepunkten markiert. So können zwar die Markerpositionen besser erkannt werden, jedoch benötigt man immer einen speziell präparierten Raum und Kleidung, um die Anwendung zu nutzen.

Unter den Ersten, die sich mit Tiefendaten auseinandersetzten, sind Malassiotis et al. [MTM+01]. Mit zwölf Bildpaaren pro Sekunde gelangen sie mittels Stereo-Sensor und „structured light“-Näherung zu den 3D-Informationen. Sie nutzen jedoch nicht ausschließlich diese Tiefeninformationen, sondern auch Farbinformationen, um damit anhand der Hautfarbe zu segmentieren. Auch Bray et al. [BKMG04] nutzen „structured light“, um Tiefenkarten zu generieren, wobei allerdings wiederum die Hautfarbe zum Ausschluss des Hintergrunds dient.

Perrin et al. [PCI04] betreiben 3D-Tracking mittels eines zielgerichteten Laserstrahls und zweier Spiegel. Über den Absolutwert des zurückgeworfenen Lichts bestimmen sie dann die Entfernung. Da jedoch nur durch einen Strahl abgetastet wird, verhält sich die Erkennungsdauer proportional zur Größe der Szene und somit ist diese Anwendung keinesfalls echtzeitfähig.

Die Einzigen, die sich schon einmal mit Gestenerkennung durch Infrarot-Tiefendaten einer time-of-flight Kamera beschäftigt haben, sind Liu und Fujimura [LF04]. Sie erkennen in Echtzeit, nutzen die 3D-Daten jedoch nicht vollständig aus. Ausschließlich die grobe Segmentierung der Hand wird über die Tiefe vorgenommen, d. h. der nächste Bereich wird herausgeschnitten und als Hand „erkannt“. Die Bewegungsanalyse und die eigentlich Erkennung basieren dann jedoch nur noch auf den 2D-Abbildungen, wobei unter Anderem ein Umrissvergleich durchgeführt wird.

In dieser Arbeit soll es deshalb nun darum gehen, bis zuletzt auf den Tiefendaten zu arbeiten, die von der Kamera in Echtzeit (30fps) geliefert werden, ein Handmodell in die Punktwolke einzupassen, und dadurch eine Erkennung zu ermöglichen.

3.2 Tabellarischer Überblick

In der folgenden Tabelle sind sämtliche im Vorfeld bearbeiteten Veröffentlichungen aufgeführt. Dabei wurde darauf geachtet, welche Daten genutzt wurden und wozu, wie die Verarbeitung ablief und was jeweils für die zukünftige Forschung und Weiterarbeit vorgesehen war.

Tabelle 1: Übersicht der bearb. Veröffentlichungen (k. A. ≡ keine Angaben)

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Albrecht-Haber-Seidel-2003 [AHS03]	eigenes Referenz-Hand-Modell Fotos (2D)	Modell einer menschlichen Hand mit anatomischer Struktur, angemessen für Echtzeit-Animation unter Verwendung einer physikalisch-basierten Muskel-Simulation und elastischen Haut-Eigenschaften	hybrides Muskel-Modell, bestehend aus Pseudo-Muskeln und geometrischen Muskeln; Deformations-Technik: Verformung der kompletten Struktur des Referenz-Hand-Modells hin zum individuellen Hand-Modell, das aus einem Foto extrahiert wurde; Markierungspunkte im Foto werden den gleichen Punkten auf dem Referenz-Modell zugeordnet; Funktion f verformt die Ausgangspunkte s_i hin zu den Zielpunkten t_i	Geometrische Muskeln sollen die Knochen bewegen; Schwerkraft in das hybride Muskel-Modell integrieren; Optimierung des Prozesses für minimale Muskelkontraktionen (Finger); Kollisionserkennung integrieren; Greifen von externen Objekten; automatische Generierung von Texturen aus den Fotos

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Athitsos-Sclaroff-2003 [AS03]	ein Clutter-Bild (durcheinander, gestreut)	Methode zur Herstellung einer bewerteten Liste von plausiblen 3D-Hand-Konfigurationen, die am Besten zum gegebenen Bild passen = zwei neue clutter-tolerante Index-Methoden für die DB	**1.** Indexing-Methode: (berechnungs-)effiziente Annäherung der „index-to-model champfer"-Distanz durch Einbettung eines binären Kanten-Bildes in einen hoch-dimensionalen Euklidischen Raum **2.** Matching-Methode: identifiziert die korrespondierenden Liniensegmente zwischen Model und gegebenem Bild, die am unwahrscheinlichsten rein zufällig vorkommen	Such-Genauigkeit noch zu gering für ein Stand-Alone-Modul zur 3D-Gesten-Erkennung interessant zu sehen, wie das System mit einem 3D-Hand-Tracker zusammenarbeiten würde Erkennung von noch mehr Hand-Formen (z.Z. ein paar 10er)
Bebis-EtAl-2002;-Kapitel4.2 Hand-Pose Estimation and Tracking [BHE+02]	synthetisch generierte Daten aus eigenem Hand-Visualisierungsmodul (Kap. 4.3); eigentlich: Handschuh mit verschiedenfarbigen Fingern und einer Ellipse auf dem Handrücken in mehreren Bildsequenzen	Erkennung der Handhaltung und Tracking	in jedem Frame werden herausgefunden: Silhouette der Hand, Finger, Position der Fingerspitzen, Gelenk-Positionen und die Ellipse auf dem Handrücken; Silhouette und Finger in allen Bildern: Rekonstruktion des Volumen der Hand; 2D Fingerspitzen- und Gelenkpositionen: mittels Triangulierung auf 3D-Positionen gemappt; Ellipse: Schätzung der Position und Orientierung der Handfläche; zusammen: Erkennung des aktuellen Zustands der Hand; Tracking mittels Kalman Filter	Erweiterung des Space-Carving-Modul bezüglich Geschwindigkeit und Genauigkeit

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Bendels-Kahlesz-Klein-2004 [BKK04]	3D Hand-Model (→ multiple Kameras); video tracking; akt. System: Hand-Tracking ↔ Dataglove Subsystem	VisionBased-HandTracking; Manipulation von Objekten	DB mit vielen Hand-Positionen; 7 Inputs: Position, Orientierung und Grab/noGrab; **1.** DB-Index mit visuellen Merkmalen **2.** Verfeinern mit *analysis-by-synthesis*	volle Integration von video-basiertem Hand-Tracking höhere Robustheit der DB-Suche
Benko-EtAl2005 [BIF05]	Daten-hand-schuh (P5) 2D-touch-sensitive-Display	cross-dimensionale Gesten-Interaktion unter Verwendung von 2D und 3D; → Interaktionstechniken	k. A.	beide Hände sollen mittels video-basiertem Gesten-Tracking verfolgt werden, was dann den Datenhandschuh überflüssig machen würde
Borst-EtAl2005 [BI05]	Cyber-Glove mit 22 Sensoren Ascension MiniBird Magnet-Tracker Hand-Model	physikalisch basierte Annäherung an Greifen und Manipulation von virtuellen Objekten mit realistischen Ergebnissen	dynamische Simulation: Feder-Modell stützt das virtuelle Greifen und Manipulieren die Feder-Hand folgt der getrackten Hand	formelle Evaluation der Qualität des Greifens and des haptischen Feedbacks Interaktion mit verformbaren und starren Objekten präzisere Richtlinien für die Abstimmung der Simulation und die Feder-Modell-Parameter
Bray-Koller-VanGool-2004 [BKMG04]	3D-Hand-Modell Tiefen-Karten generiert durch einen „structured light sensor“(Shape-Snatcher von Eyetronics)	3D-Hand-Tracking mit Hilfe des neuen SPF (smart particle filter); robust bei: Clutter und Verdeckung	neuer Algorithmus: kombiniert die Optimierung des SMD (stochastic meta-descent) mit einem Partikel-Filter, um „schlaue Partikel“ zu bilden SPF trackt hoch-dimensionale, bewegte Strukturen mit wesentlich weniger Samples als vorausgehende Methode Matchen des Hand-Modells mit den Tiefen-Karten Hautfarben-Erkennung, um den Hintergrund auszuschließen	Gebrauch mehrerer Kameras

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Bretzner-Laptev-Lindeberg-2002 [BLL02]	multi-scale Farb-eigen-schaften (Hierar-chie)	Algorithmen und ein Prototyp-System für Hand-Tracking und Erkennung der Hand-Haltung; Echtzeit-Prototyp	Partikel-Filter für Erkennung und Tracking Kombination von Umriss- und Farbhinweisen	k. A.
Buades-Perales-Varona-2004 [BPV04]	Farb-Video-Bilder von Stereo-Kameras (digitale IE-EE1394 Farb-Kameras)	robustes Echtzeit-3D-tracking-System (für Hände und Gesicht); → nutzbar als Interface bei VR-Aktivitäten in einer Workbench-Umgebung	ganzes System = Input-Capture- und Kalibrati-ons-Modul + Echtzeit-Farbsegmentierungsmodul + Datenverbindungs- und Tracking-Modul + Proze-dur zur Visualisierung von VRML und H-ANIM Tracking-Algorithmus: Hautfarbe-Pixel-Erkennung + Datenverknüpfung	Tracking von wei-teren, interessanten Körperteilen Interpretation von Bewegungen und Gesten, wofür Tracking von Fin-gern und Augen nötig werden
Corradini-EtAl1998 [CBB+98]	Video-Bilder	visuelle Lokalisa-tion von Personen mittels neuraler Felder	Lokalisation basiert auf ty-pischen Konturen; einfaches Kontur-Prototyp-Modell be-stehend aus einer Anordnung orientierter Filter für eine stückweise Annäherung der oberen Formen (Kopf, Schul-tern) einer frontal ausgerich-teten Person; zentrales Pro-blem: Selektion der heraus-ragenden Bildregionen durch 3D dynamische neurale Fel-der mit winner-take-all Pro-zess (WTA)	k. A.
Corso-Burschka-Hager2003 [CBH03]	zwei Kamera-Bilder	4D-Touchpad: Plattform für Mensch-Computer-Interfaces für robuste, ungehin-derte Interaktion = direkte Interaktion mit den Interfa-cekomponenten durch intuitive Ak-tionen und Gesten ohne irgendwelche künstlichen Tools oder Features	das Vision-System überwacht den Video-Stream für eine Sequenz aus VICs (visual in-teraction cues) grundlegende Komponente: VICon → kennt nur die jeweilige region-of-interest; → jedes definiert eine funktionsspe-zifische Menge an bildver-arbeitenden Komponenten; → hält Ausschau nach Be-wegungen, Farbanhäufungen und bekannten Gesten	Erkennung komple-xerer Gesten Erforschung von komplexe-ren statistisch-basierten Me-thoden, um die Robustheit der State-Übergänge in den VICons zu sichern (eine Möglichkeit wäre HMM)

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Couturier-Arfib-2003 [CA03]	Touch-screen zwei Semi-Gloves: vier 3D-Sensoren (flock of birds), jeweils Daumen und Zeigefinger	Pointing Fingers: ein neues Interface für musikalische Darbietungen; → User berührt einen Bildschirm, auf dem graphische Objekte dargestellt sind, und kann mit diesen simultan mit mehreren Fingern interagieren (keine Echtzeit, da die flock-of-birds Verzögerungen verursachen)	Konzept basiert auf der Überlagerung der räumlichen Anordnung der Geste und des visuellen Feedbacks; die Daten der Sensoren werden in der Max/MSP-Umgebung verarbeitet; → Sensoren senden absolute Position und Orientierung relativ zum Transmitter; → Rückgabe der (X,Y)-Koordinaten der vier Fingerspitzen auf dem Bildschirm	Entwicklung neuer Objekte Implementierung anderer Synthese-Techniken Verbesserung des Systems, so dass eine komplette Umgebung erstellt werden kann, um ein neues, digitales Musikinstrument zu kreieren
CuiSung-2004 [CS04]	3D-Hand-Modell Farbbild	3D-Modell-basiertes Verfahren zur Erfassung uneingeschränkter Handbewegungen + effektive Bewertungsmethode basierend auf Chamfer-Distanz für Umriss und Kanten	Algorithmus zur Abschätzung der Hand-Pose: integriert Lern-Methode in ein Optimierungs-Framework basierend auf dem 3D-Modell Tracking-Methode nutzt GA (generic algorithm)-basierte Partikelfilter	k. A.
Derpanis-Wildes-Tsotsos-2004 [DWT04]	temporale, monokulare Sequenz von Bildern, die ein einziges Bewegungsphonem schildern	Hand-Gesten-Erkennung innerhalb eines linguistisch basierten Frameworks (auf Phonem-Level)	Ansatz nutzt bisherige sprachliche Analysen von Zeichensprachen, die dynamische Gesten in ihre statischen und dynamischen Komponenten zerlegt: Umriss, Position und Bewegung; Lokalisation der Hand im ersten Frame durch automatisierten Prozess mittels temporaler Veränderungen und Haut-Farbe-Erkennung	Unterscheidung von lexikalischem und Satz-Level
Eisenstein-EtAl2001 [EGH+01]	haptische Daten von einem Handschuh	Erkennung von Handzeichen mittels eines einfachen Frameworks (mit Trainingsphase und Lookup-Phase)	Trainingsphase: User führt mehrmals verschiedene Zeichen vor, während er einen haptischen Datenhandschuh trägt; → das System wendet eine spezielle Clustering-Technik an (K-Mean wird mit adaptivem verglichen) und konstruiert Cluster, die den Zeichen entsprechen; Lookup-Phase: User trägt den Handschuh und macht verschiedene Zeichen; → Identifikation durch Vergleich mit den Clustern	weitere Vergleiche des Clusterings mit anderen Ansätzen des Maschinenlernens zeitliche Merkmale der Daten müssen bedacht werden

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Eisenstein-EtAl2003 [EGG+03]	Datenhandschuh	Framework basierend auf einer mehrschichtigen Betrachtung bei Gestenerkennung	mehrstufige Repräsentation der Sensor-Daten: **1.** grobe Daten (hier aus den Sensoren des Datenhandschuhs) **2.** Haltungseigenschaften: ein Satz von Eigenschaften, die die Hand-Position beschreiben **3.** temporale Eigenschaften (future work) **4.** Gesten-Masken: Satz von Masken entsprechend jeweils einer Hand-Geste Eigenschaftserkenner konvertieren die groben Sensor-Daten in einen Vektor aus Eigenschaften und der Masken-Vergleicher identifiziert dann die nächstgelegene Gesten-Maske	Testen des Systems mit anderen Handschuhen Zeigen, dass andere/verschiedenartige Zeichensprachen erkannt werden können Unterstützung von Applikationen abseits der Zeichensprache temporale Eigenschaften einfügen
Hernandez-EtAl2004 [LRKL04]	Accele-Glove zweigliedriges Arm-Skelett zwei Knöpfe zur Anzeige von Beginn und Ende der Geste	neue Methode zur Erkennung und Übersetzung isolierter Gesten der „Amerikan Sign Language“ in gesprochene und geschriebene Wörter	Gesten werden auf wesentliche Phoneme heruntergebrochen: Posen und Bewegungen; Training mit Hilfe von 12 Freiwilligen; Klassifikationsalgorithmus = Entscheidungsbaum	Kombination von Widerstand- und Trägheitssensoren mit dem Skelett Verwendung kabelloser Technologien, um das System für andere Bereiche (Animation, VR, ...) nutzbar zu machen

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Huang-Jeng2001 [HJ01]	Video-Sequenzen; Frame-Größe: 256x256; Frame-Rate: 30fps; Kamera: SONY XC7500	modell-basierte Handgesten-Erkennung mit drei Phasen: Merkmal-Extraktion, Training und Erkennung	Merkmal-Extraktion: Kombination räumlicher Informationen (Kanten) mit temporalen Informationen (Bewegung) pro Frame zur Gewinnung der Feature-Bilder; Training: Nutzung der PCA (principal component analysis), um räumliche Form-Variationen zu charakterisieren Nutzung der HMM, um die zeitlichen Form-Variationen zu beschreiben Anwendung einer modifizierten Hausdorff-Distanz-Bewerung zur Messung der Ähnlichkeit zwischen den Feature-Bildern und den zwischengespeicherten PCA-Modellen Erkennung: Generierung von Beobachtungsmustern aus der Eingangssequenz mittels vortrainierter PCA-Modelle und HMM, und Identifizierung der Geste durch Anwendung des Viterbi-Algorithmus	größere Flexibilität des vereinfachten HMM Eliminierung des Problems, dass es komplizierter und zeitaufwändiger wird, je mehr Gesten man erkennen möchte
Kahlesz-Zachmann-Klein2004 [KZK04]	Daten-Handschuh	Erstellung einer Abbildung von den Sensoren des Handschuhs auf die Gelenkwinkel der Hand mit einer hohen visuellen Genauigkeit	Abspreizung als Funktion aus drei Sensoren: Links-Biegung, Rechts-Biegung, aktuelle Spreizung	Entwicklung eines gleichwertigen Daumen-Kalibrations-Schemas
Koelsch-Turk2004 [KT04]	Grau-Wert-Bilder	robuste Handerkennung als wichtiger Schritt in Richtung leicht zu entwickelnder und zuverlässiger vision-basierter Hand-Gesten-Interfaces	Umwandlung in „Integral-Bilder“ Benutzung des Viola-Algorithmus (u.a. Fourier-Transformation) in optimierter Form	k. A.
Kong-Ranganath-2004 [KR04]	Polhemus-Tracker	hierarschiche Methode zur Erkennung isolierter 3D-Handgesten-„Flugbahnen“für SEE (Signing Exact English)	Erkennung periodischer Gesten mittels Fourier-Analyse Erkennung der Hand-Trajectories mittels VQPCA (Vector Quantization Principal Component Analysis)	k. A.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
LaViola-1999 Kapitel5 [Jr.99]	Super-Glove bzw. CyberGlove	Eingabe mittels Handgesten und Sprache; multimodale Interaktion	Erkennungskomponente bearbeitet die Rohdaten und findet wiedererkennbare Haltungen, die vorher in einer Template-Datei definiert wurden; Erkennung anhand von Square-Statistiken z.B. Ähnlichkeitsstatistiken	k. A.
Lee-Cohen-2004 [LC04]	Gelenk-Modell mit den Constraints der Kinematik einer Hand monokulare Bild-Sequenz	Möglichkeit, allgemeine, artikulierte Hand-Bewegungen mit Eigenverdeckung und Drehung der Handfläche einzufangen	Vergleich des Hand-Modells mit der erfassten Hand-Silhouette → Segmentierung der Silhouette aus dem Video-Stream und Angleichung des Hand-Modells an die 2D-Maske der Hand Motion-Tracking: *tracking-by-synthesis*	einfach zu erweitern für Echt-Zeit und Multi-View Entwicklung einer effizienten *joint angle*-Tracking-Technik für Objekte mit großer Verdeckung auf Grund der Manipulation durch den Benutzer
Lenmann-Bretzner-Thuresson-2002 [LBT02]	2D-Farb-Video-Bild	Studie über die Verwendung von Handgesten zur Interaktion (hier basierend auf ComputerVision) → keine natürlichen Gesten oder Zeichensprache	Tracking und Erkennung basieren auf einer Kombination aus „multiscale color feature“-Erkennung, view-basierten hierarschichen Hand-Modellen und Partikel-Filterung	Verbesserung der Geschwindigkeit und der Stabilität des Trackings für → größere Positionsunabhängigkeit bei der Erkennung, → größere Toleranz bei Veränderungen der Lichtverhältnisse, → bessere Performance ber komplexen Hintergründen Benuntzung anderer Menüs
Licsar-Sziranyi-2004 [LS04]	2D-Farb-Video-Bild	vision-basiertes Handgesten-Erkennungssystem, integriert in ein Kamera-Projektor-System für ein AR-Tool	Hand-Lokalisation: basiert auf Hintergrund-Subtraktions-Methode, jeweils angepasst an den projizierten Hintergrund Hand-Haltung: beschrieben durch eine Methode, die auf modifizierten Fourier-Deskriptoren basiert, die das Entfernungsmaß in die Nearest-Neighbour-Klassifikation mit einbeziehen	interaktives Training der Gesten, um erneutes Training aller Gesten zu vermeiden, wenn ein neuer User das System benutzen möchte

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Lien2005 [Lie05]	Farb-Bild-Sequenz	modellbasierte, skalierbare Hand-Posen-Analyse → 3 neue Techniken: Generierung skalierbarer Lösungen der inversen Kinematik für den Finger-Positionierungsprozess den Größenkalibrierungsprozess für das 3D-Hand-Modell die 3D-Marker-Positionsaussage-Methode für verdeckte Marker	Tracking basiert auf der Erkennung der Finger und des Handgelenks; Handgelenk: modifizierter genetischer Algorithmus (modGA) findet angenäherten Rotationswinkel; Finger: Nutzung der inversen Kinematik	Analyse komplexer Handbewegungen
LinWu-Huang-2004 [LWH04]	Video-Sequenz	Hand-Tracking mittels eines stochastischen Simplex-Algorithmus unter Verwendung des Satzes aller Trainingsbeispiele	Best-Match-Suche mittels Kombination aus sequentieller MonteCarlo-Technik und NelderMeat-Simplex-Suche	k. A.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Liu-Fujimura-2004 [LF04]	Sequenz von Echtzeit Tiefen-Daten; Tiefen- und Farbinformationen	Stellung und Bewegung der Hand	aus Video extrahiert; repräsentiert in einem Gesten-Raum: Hand-Umriss, Position, Bewegunsinformation; Hauptmerkmale: simultane Analyse vieler Merkmale: Position, Umriss, Bewegung, Orientierung, Geschwindigkeit Erkennung ohne speziellen Hintergrund, Marker oder Handschuh Echtzeit-Gesten-Erkennung wenig anfällig für Beleuchtungsänderungen (wegen Infrarot) Bestandteile: • Hand-Erkennung • Umriss-Analyse • Analyse des Bewegungsablaufs • Klassifikation von ¡Umriss,Position,Bewegungsablauf,Orientierung¿ Tiefeninformation wird nur für die Extraktion genutzt, dann werden Umrisse verglichen	mehr Bewegungs-Pattern Verstehen der Zeichensprache Erweiterung für den Fall, dass sich sowohl Benutzer als auch Kamera bewegen
MalassiotisEt-Al2001 [MTM+01]	aktiver Stereo-Sensor mit „structured light“-Näherung zur Erlangung von 3D-Informationen	neue 3D-Bild-Analyse-Algorithmen zur Segmentierung und Modellierung von Gesichtern und Händen	Erweiterung der CLA (coded light approach] zu einer CCLA (Color CLA); Gestenerkennung: Benutzung von Tiefen- und Farbinformationen **1.** Segmentierung der Hautfarbe **2.** Segmentierung des Vordergrunds anhand der Tiefeninformation → erste Segmentierung der Hand Aus der ermittelten Position der Fingerspitzen, deren Orientierung und der Handflächen-Mitte und ihrer Orientierung wird ein grobes Modell der Hand erstellt, ausreichend für statische und dynamische Gestenerkennung in Echtzeit	k. A.

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Mammen-EtAl2002 [MCA02]	eigener Hand-Tracker aus früherem Paper → trackt rechteckiges Fenster, das die Hand-Region beinhaltet	zweistufiges Schema zur dynamischen Handgesten-Erkennung	Erkennung der Gesten mit Hilfe des Outputs eines vorhandenen Trackers; hierarschiches Schema zur dynamischen Handgesten-Erkennung basieren auf einer Zustandsrepräsentation der „dominant feature trajectories“durch Verwendung von a priori Wissen über die Art und Weise, wie die Geste ausgeführt wurde **1.** Phase: Klassifizierung mittels der dominanten Features **2.** Phase: Erkennung der Geste durch Zustände, die die „dominant feature trajectories“beschreiben	k. A.
McGuire-EtAl2004 [MHRS+04]	Datenhandschuh: Accele-Glove	mobiler one-way American Sign Language Übersetzer	Kombination aus Accele-Glove-Hardware mit dem GeorgiaTechGestureToolkit (GT2K); Training anhand von 665 Sätzen; Beginn und Ende jeweils mit Knopfdruck bestätigt; Training des HMM-basierten Erkenners mit GT2K	Erfassung eines Zeichen und Phrasen Lexikons und Test des Systems damit andere, schwerere Szenarien sollen verfügbar werden Kombination des AcceleGlove mit einem Computer-Vision System: Welche Zeichen und Features kann jedes System am Besten erkennen? Gespräche mit Gehörlosen: Was wäre noch wünschenswert?
Nam-Wohn-1996 [NW96]	Einhand-VPL-Datenhandschuh mit angehängtem Polhemus-Tracker	Erkennung von Raum-Zeit-Handgesten (Bewegung, Pose und Handflächen-Orientierung) mittels HMM	Handschuh liefert 10 Winkel (zwei pro Finger) Tracker liefert absolute 3D-Position im Raum 3D-Daten werden auf 2D-Daten reduziert, dienen dann als Input für die HMMs HMM modelliert die räumliche und zeitliche Varianz	Erweiterung des Vokabulars für 3D CAD Einbau expliziter temporaler Informationen (z. B. „Pause“ zwischen den aufeinanderfolgenden Gesten) in die HMMs

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Ong-Bowden-2004 [OB04]	Video-Sequenzen (Grauwert-Bilder)	Training eines effizienten und robusten Detektors zur Erkennung der Präsenz einer Hand in einem Bild UND Klassifizierung des Hand-Umriss'	**1.** Clustern einer Bild-DB mittels „k-mediod clustering"-Algorithmus mit einer Distanz-Metrik, die auf dem Umriss-Kontext basiert **2.** Daraus wird eine Baumstruktur aus verstärkten Kaskaden konstruiert → der oberste Knoten liefert einen allgemeinen Hand-Detektor, die einzelnen Zweige klassifizieren einen korrekten Umriss, zu welchem der vorher festgesetzten Cluster er gehört; Hand finden bzw. extrahieren: Modellierung von Haut-Farbtönen mit „single Gaussian"→ Regionen mit hoher Konzentration finden	Evaluation der Genauigkeit der Erkennung und Klassifizierung bei schwierigeren Hintergründen (draußen) Kombination des Detektors mit anderen Informationsformen: Farbe, Bewegung, Hintergrund
Ong-Ranganath-2004 [OR04]	Bild-Sequenz einer digitalen Video-Kamera manuell unterteilt	Entschlüsselung der grammatikalischen Informationen aus der Art und Weise der Gesten-Bewegung → Erweiterung einer eigenen Veröffentlichung aus 2003: mehr Test-Personen und größere Anpassung an die verschiedenen Personen	Erkennung von Basis- und „Layered"-Gesten mittels des „probabilistic"Framework von Bayesian Networks (BN); automatische Segmentierung der Hand in jedem Bild mittels Thresholding basierend auf Farb- und Bewegungsinformationen → aus resultierendem Binärbild wird herausgefunden, wo der Schwerpunkt der Hand liegt und der Winkel der Achse mit der geringsten Trägheit; Definition von 6 Informationskanälen zur Extraktion der folgenden Features zur Klassifikation in Kategorien pro Kanal: • Orientierung der Hand zu Beginn und am Ende der Geste • Bewegungsrichtung bzw. -orientierung • Größe/Länge der Bewegungsbahn • Geschwindigkeitsprofil der Bewegung	Clustering der Daten einzelner Test-Personen auf Channel-Ebene, damit weniger Classifier trainiert werden müssen Erkennung kontinuierlicher Gesten-Sequenzen

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Ozeki-EtAl2004 [OINO02]	mehrere Bildsensoren: RGB-Kamera Stereo-Kamera IR-Kamera	Erkennung von Händen und in der Hand gehaltenen Objekten in Video-Raten, sogar wenn sich andere, bewegte Objekte oder Personen mit hautfarbenen Farbtönen im Hintergrund aufhalten	RGB-Kamera: Haut-Regionen (mit Kondou's research) und bewegte Regionen; IR-Kamera: Haut-Temperatur-Regionen; Stereo-Kamera: in-volume-Regionen (man geht davon aus, dass der Bereich, an der sich die Personen und Objekte befinden können, bekannt ist); → durch Integration der oben genannten Regionen lassen sich dann die Hand-Regionen und die der gehaltenen Objekte bestimmen	Tracking kleinerer Objekte effizientes Tracking durch Bewegungsvorhersage Benutzbarkeit für video-basierte Multimedia-Produktionen
Perrin-Cassinelli-Ishikawa-2004 [PCI04]	zurückgestreutes Licht während einer Laser-„Saccade“	Erkennung von Finger-Gesten basierend auf aktivem Tracking	Erkennung mittels Hidden Markov Modellen ein Weitwinkel-Photodetektor und ein zielgerichteter Laserstrahl, der von einer Laserdiode erzeugt und von zwei Mikro-Spiegeln gesteuert wird, ermöglichen 3D-Gesten Tracking: Analyse eines temporären Signals entsprechend zum Betrag des zurückgeworfenen Lichts 3D-Tracking: der Absolutwert des zurückgeworfenen Lichts entspricht der Entfernung	effizientes Interface basierend auf 3D-Gesten-Erkennung Verwendung der Gesten in Verbindung mit anderen Modalitäten (Sprache) → multimodales System, das die Komplexität und die Natürlichkeit der menschlichen Interaktion verbindet
Shamaie-Sutherland-2001 [SS01]	Bild-Sequenz (Farbe? SW?)	neuer Algorithmus zur Erkennung verdeckter und nicht-verdeckter Handgesten durch Matching der Gesten-Graphen in einem Eigen(wert)-Raum	Nutzung der Principal Component Analysis (PCA) und Projektion der Sequenz von Hand-Umrissen in einen Eigen(wert)-Raum; Finden des besten Match zwischen dem „main manifold“ einer einzelnen Geste und dem „manifold“ einer unbekannten Geste	Erkennung von „Co-Artikulation“ (eine Geste beeinflusst die nächste)
Shamaie-Sutherland-2003 [SS03]	Farbbild-Sequenz	hierarschicher Gesten-Erkennungs-Algorithmus zur Erkennung einer großen Anzahl von Gesten	**1.** Trennung der Hand vom Hintergrund durch Farbsegmentierung **2.** Erkennung des Beginns einer Geste mittels Kalman-Filter-Prozess **3.** HMM **4.** Graph-Matching	Was passiert, wenn die Anzahl der Gesten enorm ansteigt? Was passiert bei Verdeckung?

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
Starner-EtAl2003 [SLM+03]	mehrere IR-Lichtquellen Kamera-Schatten-Bild Kamera-Bild von der Seitenansicht	Perceptive Workbench (wahrnehmende Werkbank): unwillkürliches und unbehindertes Interface zwischen physikalischer und virtueller Welt; darin integriert: Identifikation der 3D Hand-Position, Zeigerichtung und pauschale Armgesten	**1.** Kamera: Schattenbild + bekannter Standort der Lichtquelle = Ebene, in der sich der Arm befindet, kann berechnet werden **2.** Kamera: Erkennung, wo der Arm in das Bild eintritt und der Position der Fingerspitze = Linie → Schnittpunkte von Ebene und Linie = 3D-Position von Oberarm und Fingerspitze	vision-basiertes Tracking des Kopfes Anwendung von HMM zur Erkennung symbolischer Handgesten, um das Interface zu kontrollieren mehrere Nutzer gleichzeitig durch sorgfältige und wirksame Verteilung der Ressourcen
Stenger-EtAl2004 [STTC04]	Bild *einer* Kamera	Lokalisation und Tracking eines *articulated object*	Unterteilung des Bildes in Unterbereiche (initiale Klassifizierungen eliminieren diese) Erstellung einer DB mit modell-generierten Bildern und Benutzung der Nearest-Neighbour-Suche, um die beste Übereinstimmung zu finden Kaskade von Klassifizierungen zur Lokalisation des Objektes Maskenvergleich: Kanten vs. Farbe	Kombination der Vorzüge von effizienter Erkennung und Tracking
Theobalt-EtAl2004 [TAH+04]	mehrfach belichtete Bilder (4 Kameras + Stroboskop) mit Markierungen auf der Hand	passive, optische Methode zum Capturen von high-speed Bewegung mittels mehrfach belichteter Bilder, die durch low-cost Stand-Kameras und ein Stroboskop erlangt werden	Rest der Person und der Hintergrund werden schwarz gemacht, man sieht nur die leuchtenden Klebepunkte; **1.** Hintergrund-Subtraktion, um die Marker-Positionen herauszufinden (jedem Marker-Typ wird ein Farb-Intervall zugewiesen) **2.** Verbindungen zwischen den verschiedenen Kamera-Sichten werden durch Epipolargeometrie festgelegt → Ermittlung der 3D-Position im Raum	Erweiterung, so dass Ganzkörperbewegungen erkannt werden können

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Veröff.	Daten	Nutzen	Verarbeitung	Ausblick
vonHarden-berg-Berard-2001 [HB01]	Video-Bilder (2D)	Interaktion zwischen Mensch und Computer mit der bloßen Hand in Echtzeit → eigener Algorithmus, um die Finger zu finden und die Handposition zu erkennen	Erkennung Identifikation Tracking modifizierter BildUnterscheidungs-Algorithmus, z.B. mit einem Referenzbild, das wegen der Beleuchtungsänderungen konstant aktualisiert wird Finden des Finger-spitzen-Umriss, woraus dann die Position der Hand ermittelt wird	schnelleren Selektions-Mechanismus finden Verbesserung der Segmentierung mit einem projizierten Hintergrund unter schwierigen Lichtverhältnissen
Wilson-Bobick-1999 [WB99]	Stereo Interactive Virtual Environment (STIVE) mit Stereo-Kameras liefert: 6D-Vektor mit der kartesischen Position jeder Hand zu jeder Zeit	neue Methode zur Repräsentation, Erkennung und Interpretation parametrisierter Gesten (zeigen systematische, räumliche Variationen; z.B. Zeigegeste, bei der nur die 2D-Richtung relevant ist)	Erweiterung der Standard-HMM-Methode zur Gestenerkennung durch Einfügen einer globalen, parametrischen Variation in die Output-Wahrscheinlichkeiten der HMM-Zustände Formulierung einer EM (expectation-maximization)-Methode mittels lineatem Abhängigkeitsmodell zum Training der parametrischen HMM	Parametrisie-rung auf spezifische Gesten zuschneiden intuitivere Parametrisierung
Wren-EtAl1997 [WADP97]	Farb-Video	Pfinder (person finder) = Echtzeit-System für Personen-Tracking und zur Interpretation deren Verhaltens	Nutzung eines multi-class, statistischen Modell von Farbe und Umriss, um eine 2D-Repräsentation des Kopfes und der Hände zu erhalten, für ein weites Feld an verschiedenen Sichtbedingungen	k. A.

4 Konzept

1. **Kalibrierung:** Zunächst muss eine Tiefenkalibrierung vorgenommen werden. Dabei wird ermittelt, welcher Z-Wert welchem realen mm-Wert entspricht. Dann folgt die Kamerakalibrierung, wodurch die (x,y)-Pixel-Werte zu einer Position in der realen Welt (abhängig von der Entfernung) umgerechnnet werden können.

   Eine ganz genaue Kalibrierung der Kamera ist jedoch nicht so einfach möglich - was ja auch schon in 2.3 beschrieben wurde - und kann wegen des gegebenen Zeitrahmens dieser Arbeit demnach auch nicht vollständig durchgeführt werden. Ziel ist es allerdings auch nicht, eine hundertprozentige Abbildung zu erhalten, sondern die Hand (dennoch möglichst genau) im Raum zu platzieren.

2. **Tiefensegmentierung:** Arm und Hand sollen mittels der Tiefendaten segmentiert und somit vom Rest der Szenen getrennt werden.

3. **Grobschätzung mittels Hauptkomponentenanalyse:** Durch die Hauptkomponentenanalyse werden der Schwerpunkt und die Hauptachsen der Punktwolke ermittelt. Diese sind für das spätere Feinmatching wichtig, da sie einen Anhaltspunkt für die Lage des Arms und der Hand im Raum bieten.

4. **Visualisierung der Ergebnisse:** Die bis zu diesem Zeitpunkt erlangten Ergebnisse werden sowohl als Punktwolke als auch als Mesh visualisiert.

5. **Modellbasiertes Feinmatching:** Nachdem die Vorarbeit geleistet wurde, wird ein Modell in die Punktwolke gelegt. Dabei sollen Translation, Rotation und Skalierung ermittelt werden.

5 Umsetzung

5.1 Kalibrierung

Der Versuchsaufbau besteht aus einem Rollcontainer und zwei Büchern. Alles wurde genau vermessen (vgl. Abb. 4 und 5), in 1.0m und 1.5m Entfernung jeweils dreimal aufgenommen (Abb. 6 und 7, sowie 23 in Anhang B) und die Ausgaben der Kamera analysiert. Die Ergebnisse wurden gemittelt und ergaben, dass ein Z-Wert von 8000 einem mm-Wert von 1000 entspricht.

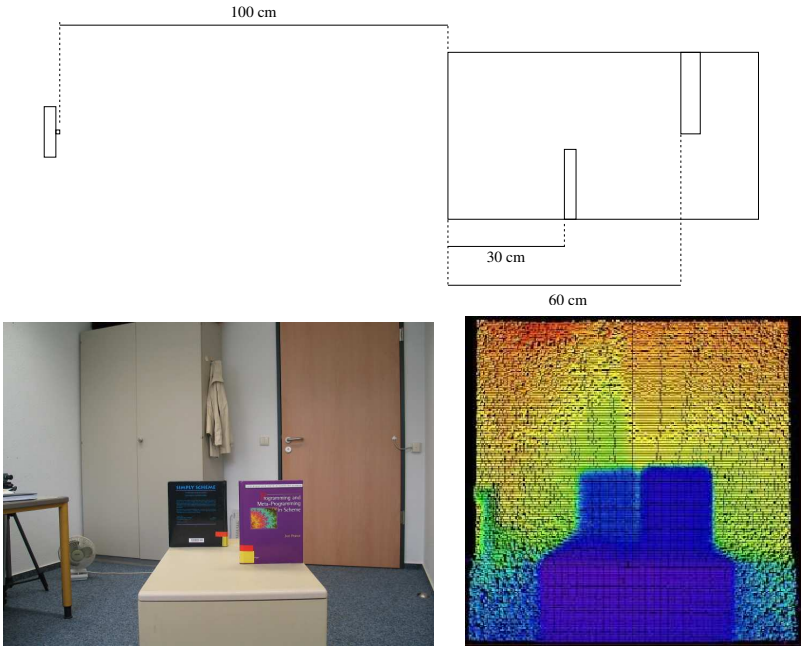

Abbildung 4: **Versuchsaufbau bei 1,0m Entfernung**

Um die Pixelwerte, im Folgenden $(u\ v)^T$ genannt, entfernungsabhängig zu einer Position in der realen Welt umzurechnen, wird folgende Formel genutzt:

$$\vec{x} \equiv \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} + \begin{pmatrix} -u \\ -v \\ f \end{pmatrix} \cdot \frac{d}{\sqrt{u^2 + v^2 + f^2}} \tag{1}$$

$(x\ y\ z)^T$: 3D-Koordinaten
$(u\ v)^T$: Kamera-Koordinaten
f: Brennweite der Kamera
d: gemessene Distanz

Wie sich diese aus Abb. 8 (entnommen aus [KI05a]) mit Hilfe des Strahlensatzes ergibt, zeigen die Formeln 5 und 6 (im Anhang A).

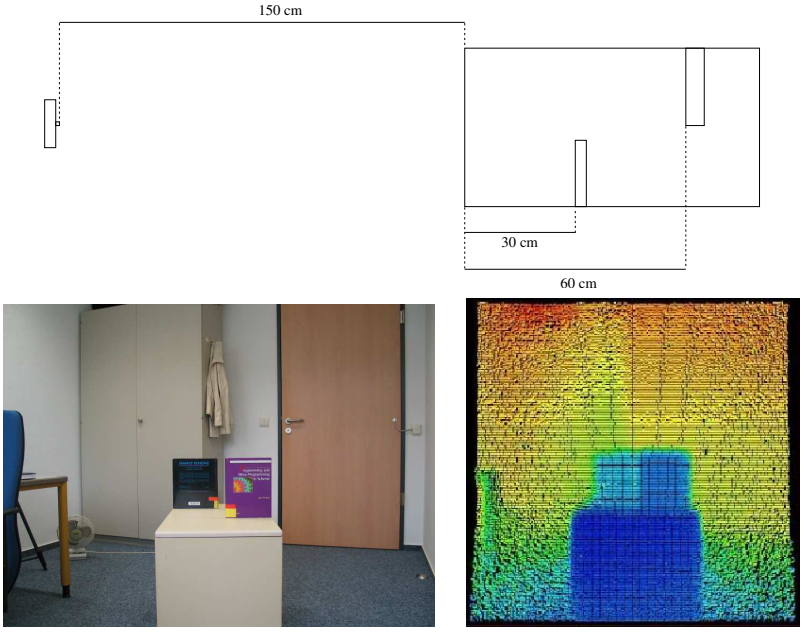

Abbildung 5: **Versuchsaufbau bei 1,5m Entfernung**

Da eine Wurzelberechnung immer zeitaufwändig ist, kann man auch eine Näherung verwenden, die davon ausgeht, dass die gemessene Distanz schon dem Z-Wert entspricht. Hierbei wird für die Umrechnung von u und v ein vereinfachter Strahlensatz (Abb. 9) verwendet, wobei man $z = d$ annimmt.

Der dabei entstehende Fehler ergibt sich durch einen Vergleich der genauen Formel mit der Näherung. Im Folgenden werden die relativen Fehler ϑ_z und ϑ_{xy} für z bzw. x und y ermittelt. Die ausführlichen Berechnungen können den Formeln 7 und 8 in Anhang A entnommen werden.

$$
\begin{aligned}
&\text{Formel für z:} && z &&= f \cdot \frac{d}{\sqrt{u^2+v^2+f^2}} \\
&\text{Näherung (für } u^2+v^2 \ll f^2\text{):} && z_n &&\approx d \\
&\text{relativer Fehler für z:} && \vartheta_z &&:= \frac{|z-z_n|}{z} \\
& && &&= \frac{\left|\frac{f\cdot d}{\sqrt{u^2+v^2+f^2}} - d\right|}{\frac{f\cdot d}{\sqrt{u^2+v^2+f^2}}} \\
& && &&= \left|1 - \frac{\sqrt{u^2+v^2+f^2}}{f}\right|
\end{aligned}
\tag{2}
$$

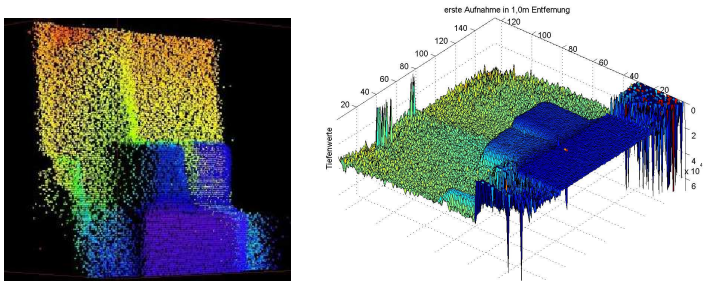

Abbildung 6: **Visualisierung der Messwerte bei 1,0m Entfernung:** Darstellung im SR2-Tool (links) und in MatLab (rechts)

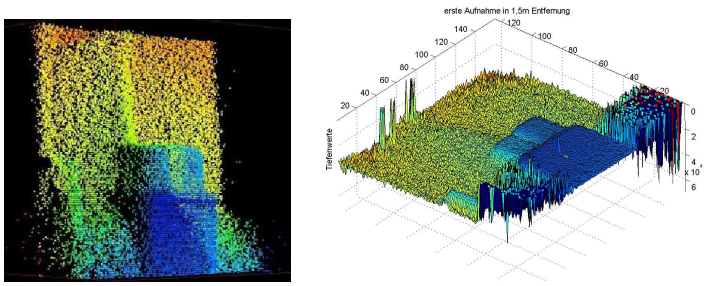

Abbildung 7: **Visualisierung der Messwerte bei 1,5m Entfernung:** Darstellung im SR2-Tool (links) und in MatLab (rechts)

Formel für x und y: $s' = s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}}$

Näherung (für $u^2 + v^2 \ll f^2$): $s'_n \approx s \cdot \frac{d-f}{f}$

relativer Fehler für x und y:

$$\begin{aligned} \vartheta_{xy} &:= \frac{|s'-s'_n|}{s'} \\ &= \frac{\left| s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}} - s \cdot \frac{d-f}{f} \right|}{s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}}} \\ &\approx \left| 1 - \frac{\sqrt{s^2+f^2}}{f} \right| = \left| 1 - \frac{\sqrt{u^2+v^2+f^2}}{f} \right| \end{aligned} \tag{3}$$

$s = (u\ v)^T$: Kamera-Koordinaten
$s' = (x\ y)^T$: 3D-Koordinaten

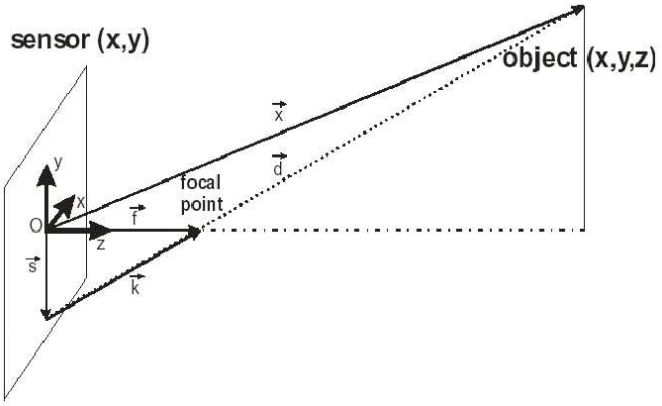

Abbildung 8: **Modell zur Koordinatenermittlung [KI05a]**

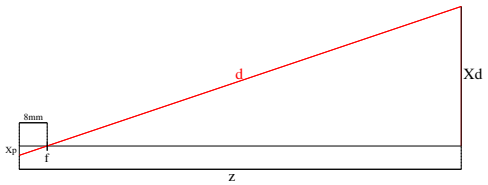

Abbildung 9: **Strahlensatz**

Je weiter die Pixel vom Zentrum des Sensors entfernt liegen, desto größer wird der Fehler. Beim größtmöglichen u und v ist der gemessene Tiefenwert um 16,5% größer als die wirkliche Entfernung, denn der maximale Fehler beträgt

$$\vartheta_{z_{max}} = \left| 1 - \frac{\sqrt{u_{max}^2 + v_{max}^2 + f^2}}{f} \right| \approx 0,165 = 16,5\% \tag{4}$$

mit $f = 8mm$, $u_{max} = 3,46mm$ und $v_{max} = 3,30mm$

Da $\vartheta_z \approx \vartheta_{xy}$, und folglich auch $\vartheta_{xy_{max}}$ mit 16,5% einen erheblichen Fehler ergibt, darf die Näherung also wirklich nur für sehr kleine u und v ($u^2+v^2 \ll f^2$) angenommen werden.

Demnach muss überlegt werden, wie die genaue Berechnung effizienter und dadurch weniger rechenaufwändig gemacht werden kann. Wegen der nur begrenzten Anzahl verschiedener (u,v)-Positionen (sie entspricht der Größe des Bildes), kann man den Wert der Wurzel $\sqrt{u^2 + v^2 + f^2}$ auch in einem Bild abspeichern und dieses dann als LookUp-Tabelle nutzen. Somit müssen alle Wurzeln lediglich einmalig beim ersten Aufruf berechnet werden.

5.2 Tiefensegmentierung

Um Arm und Hand mittels Tiefendaten richtig vom Rest der Szene zu trennen, muss eine Region of Interest (ROI) definiert werden. Die bestmögliche Bestimmung dieses Bereichs, in dem sich Hand und Arm befinden dürfen, um erkannt zu werden, ist jedoch ein großer Balanceakt. Zum Einen soll die Hand einen möglichst großen Bereich des Sensors füllen, zum Anderen aber natürlich eine Überbelichtung vermieden werden.

Der Bereich, den die Hand auf dem Sensor einnimmt, wird umso größer, je näher sie an der Kamera ist. Dieses wirkt sich allerdings auch wieder auf die Sättigung der Kamera aus. Denn je näher die Hand an der Kamera ist, desto mehr beeinflusst das an der Oberfläche gestreute Licht die Messwerte in der Umgebung. Das ausgesendete Licht wird von den Objekten der Szene, also auch von der Hand und dem Arm, diffus zurückgestreut. In großer Entfernung wirkt sich das nicht nennenswert auf die Messungen der Kamera aus. Befindet sich das Objekt hingegen in der Nähe des Sensors, so kommen die dort gestreuten Strahlen eher - vor den eigentlich zugehörigen Strahlen, die weiter hinten reflektiert werden und deshalb einen längeren Weg zurückzulegen haben - bei den Pixeln in der Umgebung an. Demzufolge werden sozusagen die falschen Strahlen für die Entfernungsberechnung genutzt, erwecken den Eindruck, dass an einem nahegelegenen Objekt reflektiert wurde und beeinflussen somit negativ die Messung in diesem Bereich. Das letzte Bild in Abb. 12 zeigt deutlich diese fehlerhaften Messungen, denn der Hintergrund scheint im unteren Bereich entgegen dem Ausgangszustand (Bild in der oberen Mitte) näher gekommen zu sein. Die Hand ist dadurch nicht mehr eindeutig vom Hintergrund zu trennen, was eine gute Segmentierung in diesem Tiefenbereich dementsprechend sehr schwierig macht. Aus diesem Grund muss also ein geeigneter Kompromiss gefunden werden, um eine Überbelichtung zu vermeiden und trotzdem eine möglichst große Abdeckung des Sensors durch die Hand zu erhalten.

Zu diesem Zweck wurde der genauere Zusammenhang von Entfernung, Sensor-Abdeckung und Überbelichtung eingehend untersucht. Hierbei wurde eine Hand in verschiedenen Entfernungen von der Kamera aufgenommen (vgl. Abb. 10 und 11) und die Ergebnisse analysiert. Abbildung 12 zeigt eine Auswahl dieser Ergebnisse, die ganze Aufnahmeserie befindet sich in Abb. 24 im Anhang.

Die ersten fünf Bilder der Serie in Abb. 24 zeigen die Hand in einer Entfernung von 2,0m bis 1,6m aufgenommen. Dabei nimmt die Hand nur eine sehr kleine Region des Bildes ein und ist deshalb nicht so gut für die Erkennung verwendbar. Im Bereich zwischen 1,5m und 1,0m (Bilder 6 - 11 der Serie) ragt die Hand besonders gut heraus. Sie ist nah genug an der Kamera, so dass man sie gut erkennen kann, und dennoch treten keine nennenswerten Überstrahlungen auf. Bei den nächsten drei Aufnahmen (0,9m - 0,7m) ist un-

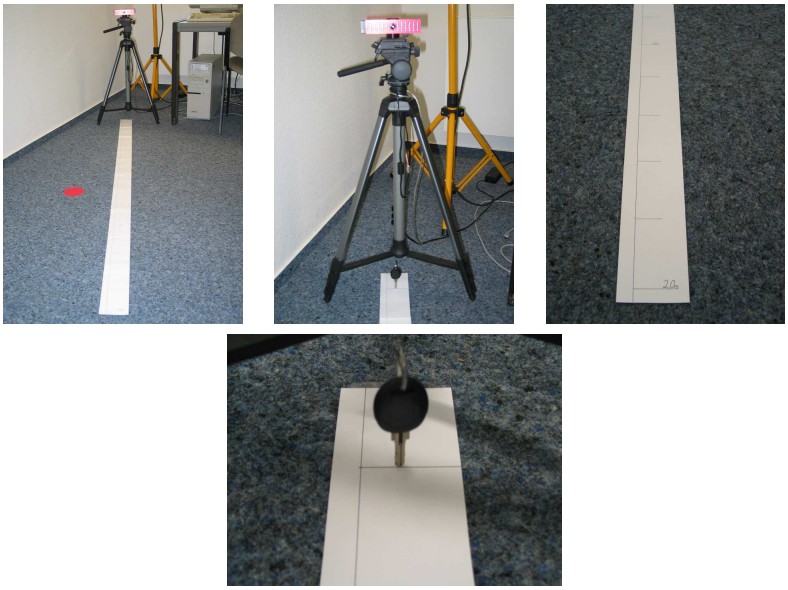

Abbildung 10: **Kamera mit Maßband und Lot:** Um die tatsächliche Entfernung zur Kamera exakt bestimmen zu können, wurde ein Maßband erstellt und dieses in Blickrichtung der Kamera genau platziert.

weit des Arms und der Hand eine Überstrahlung zu erkennen. Im Vergleich zum Ausgangszustand scheint die Szene am unteren Rand näher gekommen zu sein. Eine besonders deutliche Überstrahlung ist, wie oben schon erwähnt, auszumachen, wenn sich die Hand in direkter Nähe der Kamera befindet. Dies zeigen die vorletzten beiden Bilder der Sequenz, vor Allem noch einmal im Vergleich zum Ausgangszustand (letztes Bild). Die Hand und der untere Bereich des Hintergrunds scheinen gleich weit von der Kamera entfernt zu sein - gut erkennbar durch die Farbkodierung der Aufnahmen (Lila (0m) bis Rot (7,5m)) - demzufolge eine Erkennung in diesem Bereich nicht gegeben ist. Als geeigneter Kompromiss zeichnet sich infolgedessen der Bereich zwischen 1,0m und 1,5m ab, d. h. zwischen den Tiefenwerten 8000 und 12000 der Kamera.

Wie schon in Kapitel 2.3 angesprochen, bietet die Kamera jedoch zusätzlich noch die Möglichkeit, einige eigene Einstellungen, je nach Einsatzgebiet, an ihr vorzunehmen. Die oben thematisierten Versuche wurden mit der „Voreinstellung“ (vgl. Tabelle in 2.3) der Kamera gemacht. Durch kleine Veränderungen an den Parametern kann man jedoch noch ein besseres Ergebnis erzielen. Setzt man die Werte der Belichtungszeiten neu (LongIntegrationTime = 48 ([*256μs] = 12,288ms) und ShortIntegrationTime = 16 ([*256μs] = 4,096ms)), so kann man die Überstrahlung minimieren und bis auf einen halben Meter an die Kamera herangehen, ohne einen nen-

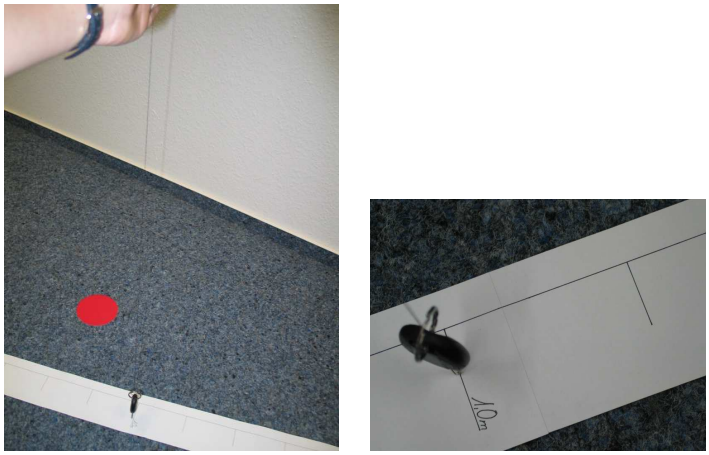

Abbildung 11: **Hand mit Lot für die ROI-Ermittlung:** Um die Hand in den zu messenden Entfernungen genau platzieren zu können, wurde an ihr ein Lot befestigt, das auf dem Maßband die Entfernung anzeigte.

nenswerten Fehler in der Selektierung zu bekommen; d. h. die Hand nimmt einen größeren Bereich im Sichtfeld der Kamera ein und kann so besser erkannt werden, überstrahlt dabei aber nicht die ganze Szene. Durch diese Veränderung wird der Nahbereich besser erkannt, wohingegen nicht mehr die volle Messdistanz von 7,5m erreicht werden kann. Verringert man die Integrationszeit, so wird früher ausgelesen wieviele Photonen registriert wurden (vgl. Kapitel 2.3). Von weiter entfernten Objekten kommen ohnehin schon weniger Photonen zurück und durch das frühere Auslesen wird die Anzahl derer, die gemessen werden, noch einmal verringert. Da bei dieser Anwendung jedoch auch nur der Nahbereich betrachtet wird, ist die Erhaltung der vollen Reichweite auch nicht vonnöten, wodurch sich der Region of Interest der Bereich zwischen 0,5m und 1,0m (Werte der Kamera zwischen 4000 und 8000) zuweisen lässt.

Durch diese Festlegung wird von jedem Bild, das pro Frame von der Kamera übergeben wird, eine Maske erstellt. In dieser werden die Pixel, deren jeweiliger Tiefenwert innerhalb des Bereichs liegt, mit 1 markiert und die restlichen mit 0. Für diese Maske wird daraufhin eine „connected component“-Analyse durchgefürt. Dabei werden mit 1 markierte, benachbarte Pixel als zusammengehörig gekennzeichnet. Nur der größte, zusammenhängende Bereich wird dann als relevant betrachtet und für die Weiterverarbeitung genutzt. Dies ist notwendig, da es immer wieder kleine Gruppen von Ausreißern gibt, wobei Punkte in der relevanten Tiefe gemessen werden, obwohl in Wirklichkeit dort gar nichts existiert und sie also auch nicht für die weitere Erkennung verwendet werden dürfen.

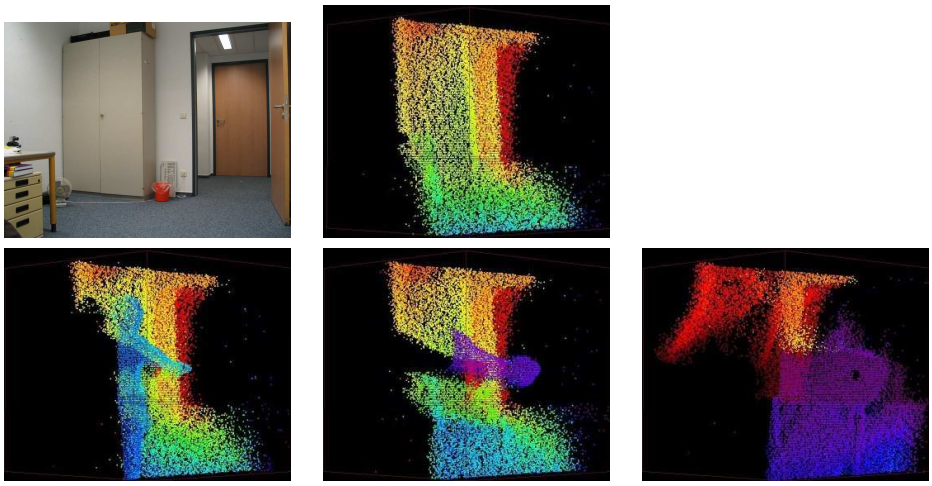

Abbildung 12: **Auswahl der Visualisierungen mit dem SR2-Tool:** Die erst Zeile stellt den Ausgangszustand dar, in der Mitte die Sicht der Kamera als Punkte visualisiert, links eine Foto-Aufnahme der Szene zum Vergleich. In der zweiten Zeile sind die Aufnahmen der Hand in den Entfernungen 2,0m, 1,0m und 0,5m zu sehen. Das linke Bild zeigt die Hand in weiter Entfernung, kaum noch zu erkennen. Das rechte Bild hingegen stellt die Abbildung der Hand in sehr geringer Entfernung dar, wobei die deutliche Überstrahlung auszumachen ist. Dies sind die beiden Extreme, die auftreten können. Das mittlere Bild veranschaulicht den eingegangenen Kompromiss. Es sind keine Überstrahlungen zu sehen, und trotzdem nimmt die Hand einen ausreichend großen Bereich des Sensors ein, um sie gut zu erkennen. Die Darstellung der Gesamtserie dieser Messungen befindet sich im Anhang (Abb. 24)

5.3 Grobschätzung mittels Hauptkomponentenanalyse

Die tiefensegmentierten Bilddaten werden im Folgenden als 3D-Punktwolke betrachtet. Da es für das spätere Feinmatching wichtig ist, die Lage der Punkte im Raum zu ermitteln, bietet sich als geeignete Methode die Hauptkomponentenanalyse an, denn sie liefert den Schwerpunkt und die Hauptachsen der Punktwolke zurück.

5.3.1 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (principal component analysis) ist eine Methode, um die Hauptausdehnungsrichtungen einer Punktwolke zu bestimmen. Dafür muss die Punktwolke zunächst durch eine quadratische Matrix repräsentiert werden, die Kovarianzmatrix. Die Eigenvektoren dieser Matrix bestimmen dann die Hauptausdehnungsrichtungen der Punktwolke.

Aus [Dre05]: Gegeben sei eine Folge $x = x_1, \ldots, x_n \in R^m$ von m-dimensionalen Datenpunkten. Dann bezeichnet die $m \times m$-Matrix

$$\mathrm{CovMat}(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}) \cdot (x_i - \bar{x})^T$$

die *Kovarianzmatrix* von x. Der Vektor $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ ist der Vektor der Mittelwerte der einzelnen Komponenten von $x_1, \ldots, x_n$.
Da hier im dreidimensionalen Raum gearbeitet wird, besteht jeder Vektor x aus drei Werten, der Pixelposition (x,y) im Bild und dem Tiefenwert der Kamera. Dadurch sieht die Kovarianzmatrix dann folgendermaßen aus:

$$\mathrm{CovMat} = \begin{pmatrix} \mathrm{Cov}_{x,x} & \mathrm{Cov}_{x,y} & \mathrm{Cov}_{x,z} \\ \mathrm{Cov}_{y,x} & \mathrm{Cov}_{y,y} & \mathrm{Cov}_{y,z} \\ \mathrm{Cov}_{z,x} & \mathrm{Cov}_{z,y} & \mathrm{Cov}_{z,z} \end{pmatrix}$$

wobei $\mathrm{Cov}_{a,b} = \frac{1}{n} \sum_{i=1}^{n} (a_i - \bar{a}) \cdot (b_i - \bar{b})^T$ die Kovarianzen sind.

5.3.2 Anwendung im System

Mittels der Hauptkomponentenanalyse werden der Schwerpunkt und die Hauptachsen der Punktwolke ermittelt. Diese sind für das spätere Feinmatching wichtig, da sie einen geeigneten Anhaltspunkt für die Lage des Arms und der Hand im Raum bieten.

Durchführung:

1. Berechnung des Mittelwerts $\bar{x}$ der Punktwolke
2. Berechnung der Kovarianzmatrix
3. Berechnung der Eigenwerte und -vektoren der Kovarianzmatrix

5.4 Visualisierung der Ergebnisse

5.4.1 Darstellung als Punktwolke

Die Visualisierung als Punktwolke mit den Hauptachsen (die dazu eingeblendet werden; vgl. Abb.13), dient dazu, zu erkennen, ob der Bereich der Hand richtig aus den Tiefendaten herausgeschnitten wird, und ob die berechneten Achsen richtig im Raum liegen. Dabei werden die Messpunkte unter Verwendung eines GeoSet-Knotens in Avango/Performer visualisiert. Die in 3D-Positionen umgerechneten Werte werden übergeben und dienen im GeoSet-Knoten als Koordinaten. Als Ausgabeform *PrimType* werden

Punkte gewählt. Über die Anpassung des Parameters *numPrims* an die jeweilige Anzahl übergebener 3D-Positionen wird gewährleistet, dass nur die neu übergebenen Werte als Koordinaten dienen und nicht, weil die Punktwolke vorher mehr Punkte beinhaltete, noch alte Werte im nachfolgenden Speicherbereich mit berücksichtigt werden. Die Hauptachsen werden als Linien dargestellt, wobei sich Anfangs- und Endpunkt wie folgt ergeben: Anfangspunkt der Linie ist immer der übergebene Mittelpunkt der Punktwolke, der Endpunkt ergibt sich aus der Summe des Mittelwertes und des jeweiligen, darzustellenden Eigenvektors.

Abbildung 13: **Punktwolke mit Hauptachsen**

5.4.2 Darstellung als Mesh

Durch die Visualisierung als Mesh (Abb. 14) bekommt man einen besseren Eindruck davon, wie die Punkte zueinander im Raum liegen. Die Werte werden in 2D trianguliert und dann als Dreiecke visualisiert. Es wird nicht in 3D trianguliert, da nur der fordere Teil der Oberfläche vermessen wird. Es werden keine Voxel gemessen und so ergibt sich insgesamt auch keine vollständige Oberfläche, sondern eher eine Seitenansicht. Man könnte das Ganze als $2\frac{1}{2}$D bezeichnen, denn die Tiefe ist zwar gegeben, jedoch nur von der Vorderseite des Objekts. Umgesetzt wird die Triangulierung dabei mit Hilfe der CGAL Basic Library. CGAL (Computational Geometry Algorithms Library) umfasst insgesamt drei Bibliotheken, wobei die verwendete Basic Library eine Sammlung standardmäßiger Datenstrukturen und geometrischer Algorithmen zur Verfügung stellt [CGA05], wovon hier die *Constrained_Delaunay_triangulation_2* verwendet wurde.

Auch in diesem Fall werden die Messpunkte mit Hilfe eines GeoSet-Knotens visualisiert, jedoch als Ausgabeform Dreiecke gewählt. Die Werte werden der Triangulierungsfunktion übergeben und können nach deren Berechnung

mit Hilfe eines Iterators extrahiert werden. Jeweils die drei zu einem Dreieck gehörende Punkte werden der Triangulierung entnommen und dem GeoSet-Knoten als Koordinaten übergeben. Auch hier wird der Parameter *numPrims* angepasst, er muss der Anzahl der Dreiecke entsprechen, die bei der Triangulierung entstandenen sind.

Betrachtet man sich die Seitenansicht der Darstellung (Bild links unten in Abb. 14) genauer, so ist hier sehr deutlich der Effekt des Kometenschweifs (vgl. 2.3) zu erkennen. Die gemessenen Tiefenwerte strahlen eindeutig an den Objekträndern nach hinten aus und ähneln so dem Schweif eines Kometen.

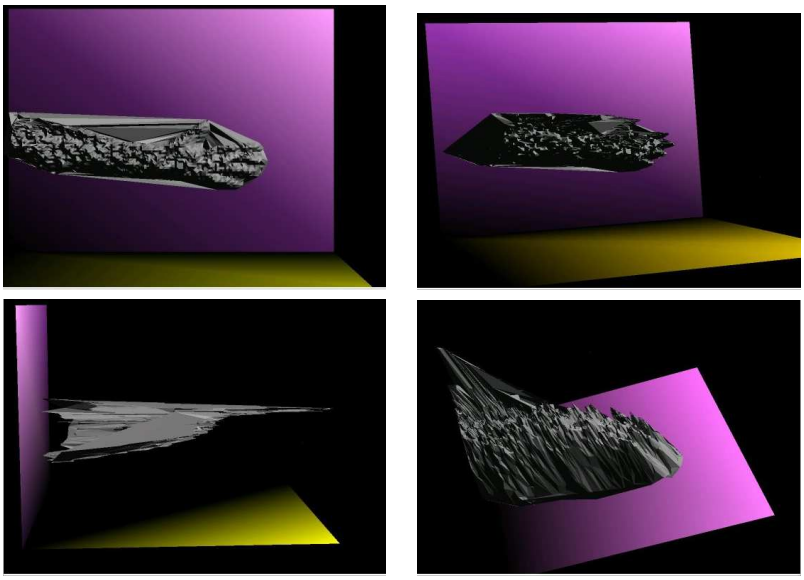

Abbildung 14: **Visualisierung als Mesh**

5.4.3 Auswertung

Es zeigt sich, dass die erste Hauptachse immer in die Tiefe zeigt: Das hängt mit der großen Varianz der Tiefenwerte zusammen. Dieses Phänomen ist jedoch nicht so leicht (z.B. mittels eines Medianfilters) zu entfernen, weil es mit dem oben schon angesprochenen Kometenschweif in Beziehung steht. Da somit das Problem kontinuierlich auftritt, lässt sich folgendes festlegen:

1.Hauptachse: zeigt in die Tiefe

2.Hauptachse: Zeigerichtung des Arms (+ Hand)

3.Hauptachse: „Dicke“ des Arms

Mit den zu diesem Zeitpunkt bestehenden Ergebnissen wäre auch schon eine einfache Steuerung möglich. Allein durch die Verfolgung des Schwerpunktes könnte beispielsweise bereits ein Zeiger gesteuert oder ein Objekt bewegt werden. Außerdem wäre es denkbar, damit Musik zu machen. Der X-Wert bestimmt die Schnelligkeit der Tonfolgen, der Y-Wert die Tonhöhe und der Z-Wert die Lautstärke. Des Weiteren könnten die Hauptachsen genutzt werden, um die verschiedensten Dinge damit anzusteuern, denen man nicht nur eine Position sondern auch eine Richtung geben will. Denkt man sich ein Dartspiel, so ist es möglich über den Schwerpunkt und eine der Hauptachsen den Pfeilflug zu beeinflussen. Der Schwerpunkt markiert die Abwurfstelle und die Hauptachse die -richtung.

5.5 Modellbasiertes Feinmatching

Durch die Ermittlung des Schwerpunktes und der Hauptachsen der Punktwolke und die Erkenntnisse aus der Visualisierung der Ergebnisse ergibt sich eine gute Grundlage, um nun ein Modell in die Punktwolke zu legen. Ziel ist es, Translation, Rotation und Skalierung des Modells entsprechend zu ermitteln. Ein Skelett oder auch dreidimensionales Handmodell soll so in der Punktwolke positioniert werden, dass sich dessen Abstand zu ihr minimiert.

Da die Punktwolke jedoch eine dreidimensionale Repräsentation des Arms und der Hand im Raum darstellt, wohingegen das Skelett keine Ausdehnung in der dritten Dimension hat, kann es nicht alleine zur genauen Positionierung genutzt werden.

Zu diesem Zweck wurde ein dreidimensionales Handmodell verwendet. Dieses Modell besteht aus 1515 Vertex-Punkten und 3000 Dreiecken und entspricht einer menschlichen Hand mit anatomischer Struktur. Es wurde angemessen für Echtzeit-Animation unter Verwendung einer physikalisch-basierten Muskel-Simulation und elastischer Hauteigenschaften erstellt und mir für diese Arbeit von Frau Irene Albrecht [AHS03] zur Verfügung gestellt. Das Muskel-Modell kommt hier nicht zum Einsatz. Nur das reine Dreiecksnetz bzw. die Vertex-Punkte werden verwendet.

Diese Punkte müssen zunächst eingeladen und an die Position der Skelletthand gebracht werden, bevor man mit der genauen Einpassung in die Punktwolke beginnen kann. Wie oben schon erwähnt, soll dabei der Abstand des Modells zur Punktwolke minimiert werden. Als erste Möglichkeit bietet sich dafür die Betrachtung der reinen Punktabstände an. Da das Handmodell jedoch auch als Dreiecksnetz vorliegt, besteht eine weitere Möglichkeit darin, die Oberflächen miteinander zu vergleichen. Die Umwandlung der Punktwolke in ein Dreiecksnetz für diesen Vergleich wurde ohnehin schon in 5.4.2 realisiert. Die Hausdorff-Distanz stellte dabei eine Option dar. In diesem Falle werden jedoch nicht explizit die Oberflächeneigenschaften ausgenutzt,

Abbildung 15: **Handmodell**

sondern auch „nur“ versucht, die Punktwolken bzw. -mengen möglichst nahe zueinander zu bringen. So gesehen ist sie der ICP (iterative closest point)-Registrierung nicht unähnlich, zu diesem Zeitpunkt jedoch noch nicht einsetzbar. Das Modell muss zuerst schon recht gut in der Punktwolke positioniert werden, um diesen Methoden einen geeigneten Ausgangspunkt zu liefern, und dann mit ihnen eine noch präzisere Registrierung vorzunehmen. Somit werden im Folgenden für die Lösung des Korrespondenzproblems, dem Matching der Modelldaten auf die Sensordaten, die reinen Abstände der Messpunkte zu den Modellpunkten betrachtet, und diese versucht zu minimieren. Daraus ergibt sich folgende Kostenfunktion für die Summe der minimalen quadratischen Abstände:

$$K(\vec{\alpha}, t) := \sum_{<i,j>} \left[\left[\vec{x}_i^{(P)}(t) - \vec{x}_j^{(M)}(\vec{\alpha}) \right]^2 M_{ij} \right]$$

$\vec{\alpha} = (t_x, t_y, t_z, \phi_x, \phi_y, \phi_y, s)$: Vektor, der die drei Translationsfreiheitsgrade, drei Rotationsfreiheitsgrade und die homogene Skalierung (entsprechend der möglichen DOFs) enthält
$\vec{x}_i^{(P)}(t)$: $i$ter Punkt der Punktwolke zum Zeitpunkt t
$\vec{x}_j^{(M)}(\vec{\alpha})$: $j$ter Punkt des Modells nach Durchführung sämtlicher in $\vec{\alpha}$ enthaltener Transformationen
$M_{ij} \in \{0, 1\}$: diese Matchmatrix repräsentiert die Korrespondenz von 3D-Punkt i mit Modellvertex j (oder umgekehrt); wurde ein 3D-Punkt i ermittelt, der dem betrachteten Modellpunkt j am nächsten liegt, so beträgt der Wert der Matchmatrix an der Stelle ij 1; über die Korrespondenzen werden die Paare mit den kürzesten Abständen definiert.

Durch die Veränderung der Transformationsparameter innerhalb von $\vec{\alpha}$ soll diese Funktion dann minimiert werden. Dabei ist es wichtig, die Sichtbarkeit der Punkte des Modells zu berücksichtigen, wie sich bei der Skalierung noch zeigen wird. Es gelten nur die Punkte als sichtbar, die nicht aus dem Sichtfeld herausragen und deren vom Dreiecksnetz ausgehenden Normalen nicht von der Kamera weg zeigen.

Da durch die im Vorfeld durchgeführte Hauptkomponentenanalyse nicht nur die grobe Lage der Punktwolke, sondern dadurch auch die des Skeletts ermittelt wurde, kann man sich dies zunutze machen. Zuerst wird das ganze Skelett zum berechneten Schwerpunkt der Punktwolke verschoben (globale Translation) und dann so rotiert, dass es auf der zweiten Hauptachse liegt. Durch die Bestimmung der „Länge“ des sich im Sichtfeld befindenden Objekts (Ausdehnung der Punktwolke in Richtung der zweiten Hauptachse) mit Hilfe der Bounding-Box, kann das Modell entsprechend entlang dieser Achse (dem Unterarm) verschoben werden, so dass die Fingerspitzen mit dem Rand der Wolke abschließen.

Schließlich wird das Modell noch skaliert, denn je weiter man von der Kamera weggeht, desto weniger Pixel „sehen“ die Hand, d.h. die zur Hand gehörenden Punkte werden weniger. An diese perspektivische Verzerrung muss das Modell angepasst werden. Es gibt zwei Varianten, während bei beiden die Verschiebung entlang des Unterarms jeweils entsprechenden angepasst wird, damit die Fingerspitzen die Punktwolke nicht verlassen, d. h. je größer skaliert wird, desto weiter muss das Modell unter Einbeziehung seiner bekannten Ausmaße entlang des Unterarms „zurückverschoben“ werden.

1.Variante: Es wird ermittelt, bei welcher Entfernung das Handmodell optimal in die Punktwolke passt und dieser Tiefenwert als Entscheidungshilfe, ob und wieweit man größer oder kleiner skalieren muss, genutzt. Dazu wird die mittlere Distanz der „Punkte, die am nächsten zur Hand sind“, berechnet und entsprechend skaliert.

2.Variante: Diese Variante folgt dem Grundsatz „einfach Skalieren und dann die Summen der Distanzen vergleichen, um herauszufinden, ob die Maßnahme in die richtige Richtung ging.“ Das bedeutet, das Modell wird bei jeder Iteration jeweils entweder um den Faktor 0,1 größer oder kleiner skaliert. Verringern sich die Punktabstände, so wird die Skalierung in diese Richtung fortgesetzt, andererseits abgebrochen und in die andere Richtung fortgeführt. Ist der Punkt erreicht, an dem weder Größer- noch Kleinerskalierung die Reduzierung der Abstände zur Folge haben, so ist der endgültige Skalierungsfaktor gefunden. Wie oben schon erwähnt, ist dabei die Berücksichtigung der Sichtbarkeit sehr wichtig, da ansonsten versucht wird, das ganze Modell in immer weniger Punkte einzupassen. Geht man mit der Hand immer mehr aus dem Sichtfeld der Kamera heraus und berücksichtigt die Sichtbarkeit nicht, so folgt die Modellhand ihr nicht nach, sondern schrumpft, bis sie vollständig in der Punktwolke enthalten ist. Allerdings tauchen durch die Berücksichtigung der Sichtbarkeit auch wieder neue Probleme auf. Je weniger Punkte zu sehen sind, desto geringer wird auch die Summe der Distanzen und der Algorithmus produziert eine riesige Hand, die kaum noch sichtbare Punkte enthält. Abhilfe schafft man durch die Einführung weiterer Kriterien. Zum Einen wird die ganze Erkennung und somit auch die Skalierung ausgesetzt, wenn nur noch weniger als 1/3

der Punkte des Modells im sichtbaren Bereich sind, zum Anderen werden nicht nur die Distanz-Summen sondern auch die mittleren Distanzen verglichen. Entfernt sich nun die Modellhand durch die Größerskalierung aus dem Sichtfeld, verringert sich dadurch die Anzahl der sichtbaren Punkte und gleichzeitig auch die Summe der Distanzen. Betrachtet man dann jedoch zusätzlich den mittleren Abstand des Modells zur Punktwolke, so stellt sich heraus, ob zu Recht größer skaliert wurde, oder ob sich die Modellpunkte durch diese Skalierung nur noch weiter von der Punktwolke entfernt haben. Ist also der mittlere Abstand im Gegensatz zur Distanz-Summe angewachsen, so darf nicht noch größer skaliert werden und der Algorithmus bricht die Maßnahme in diese Richtung ab.

Schließlich ist demnach festzuhalten, dass es durch die beschriebenen Verarbeitungsschritte (visualisiert in Abb. 14) möglich ist, die globale Translation (t_x,t_y,t_z), Rotation (ϕ_x, ϕ_y, ϕ_y) und Skalierung des Modells entsprechend der Punktwolke zu bestimmen und weiterführende Maßnahmen nun dahingehend verfolgt werden müssen, zusätzliche Gelenkwinkel zu ermitteln, um dann auch ein artikuliertes Modell verwenden zu können.

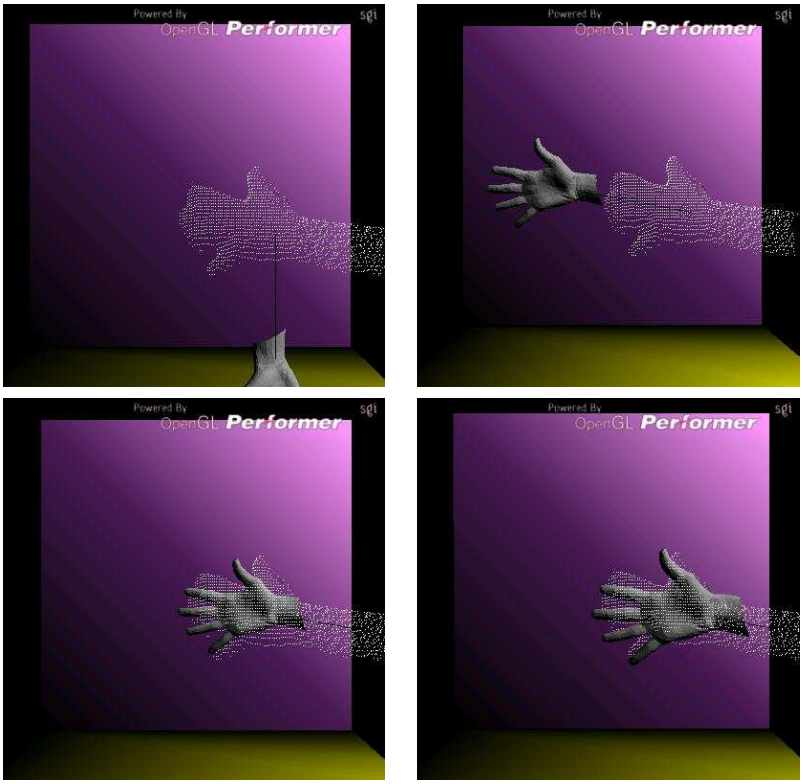

Abbildung 16: **Verarbeitungsfolge** links oben: zum Schwerpunkt verschoben; rechts oben: auf zweite Hauptachse rotiert; links unten: in die Wolke geschoben; rechts unten: skaliert

6 Ergebnisse

Um direkt einen Eindruck von den Ergebnissen zu erhalten, sind im Folgenden Bildsequenzen zur Veranschaulichung dargestellt:

Abbildung 17: **Bildsequenz 1 (Frontalansicht):** zeigt eine kleine Abwärtsbewegung der Hand, die von rechts ins Bild ragt

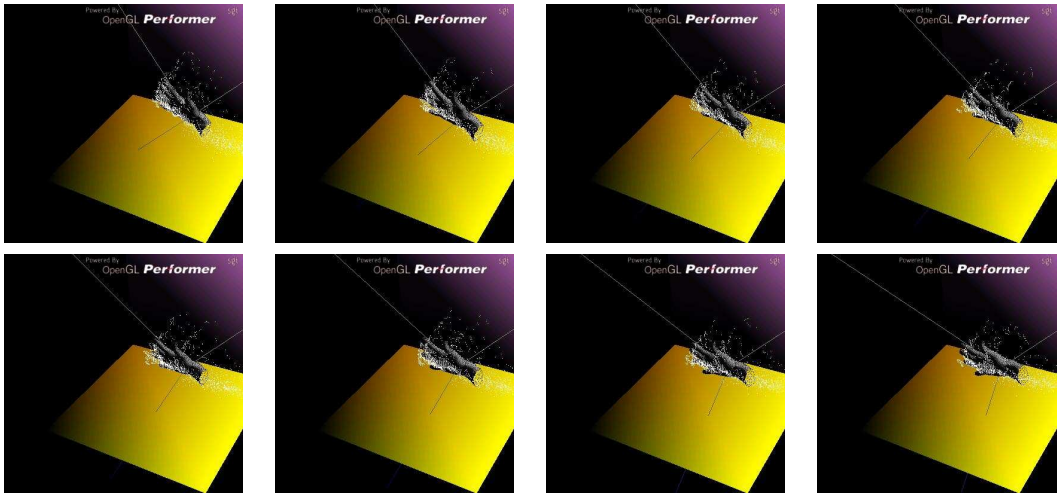

Abbildung 18: **Bildsequenz 1 (seitliche Ansicht):** zeigt die selbe Bewegung wie Abb. 17 aus einem anderen Blickwinkel

Abbildung 19: **Bildsequenz 2:** zeigt das Eintreten der Hand ins Sichtfeld und eine darauf folgende Abwärtsbewegung

Abbildungen 17 und 18 dokumentieren eine gelungene Erkennung. Das Handmodell liegt gut in der Punktwolke und folgt ihr nach. Dies ist sowohl in der Frontal- als auch in der seitlichen Ansicht deutlich zu erkennen. In Abbildung 19 zeigen sich noch bestehende Probleme. An den ersten beiden Bildern wird die Schwierigkeit erkennbar, die Hand beim Eintritt in die Szene richtig zu erfassen. Bei Bild 1 wurde die Eintrittsrichtung nicht korrekt erkannt, wodurch das Modell nicht richtig orientiert im Raum liegt. In Bild 2 versucht der Algorithmus durch Schrumpfen des Modells, dieses in die gesamte Punktwolke einzupassen. Hier brachte die zusätzliche Betrachtung der mittleren Distanz nicht den angenommenen Erfolg. Ähnliches lässt sich auch in den Bildern 9 und 11 erkennen. Dort wird ebenfalls nicht richtig skaliert. Betrachtet man die restlichen Bilder, so wurde dort das Handmodell gut in die Punktwolke eingepasst.

Mit einem 3.00GHz Intel Pentium 4 mit 2GByte Speicher wurde bei dieser Erkennung von 7 Freiheitsgraden (Translation (t_x,t_y,t_z), Rotation (ϕ_x, ϕ_y, ϕ_y) und Skalierung) eine Frame-Rate von 2-3fps erreicht, wobei die Messungen jedoch unter dem Vorbehalt betrachtet werden müssen, dass der Algorithmus noch nicht optimiert wurde.

Das optische Feedback war zwar auch recht zeitaufwändig, zunächst in seiner Konzipierung und dann ebenso durch das Mehr an Rechenaufwand/-leistung für die zusätzlichen Berechnungen und die Darstellung, jedoch diente dies der Überprüfung der gemachten Berechnungen. Es kann später einfach abgeschaltet werden, wenn es nicht mehr vonnöten ist. Da sich ohne optisches Feedback jedoch nur eine fast nicht messbare Verbesserung von durchschnittlich 16,5ms pro Frame ergibt, zeigt sich, dass die Darstellung rechentechnisch förmlich nicht ins Gewicht fällt.

Eine der Hauptaufgaben war es, herauszufinden, ob und inwieweit man Gesten aus Tiefenbildern erkennen kann.
Im Laufe dieser Arbeit sind dabei viele Vorteile aber auch Nachteile der Tiefendaten deutlich geworden, die im Folgenden nun erläutert werden sollen.

Der grundlegende Vorteil besteht darin, dass man mit nur einer Aufnahme einer einzigen Kamera ein mit Tiefeninformationen versehenes Bild der Umgebung erhält. Mit anderen Verfahren ist dies nur mit größerem Zeitaufwand oder mehreren Kameras, also auch größerem Rechen- und Zeitaufwand, möglich.

Außerdem ist bei der Erkennung aus Tiefenbildern kein spezieller Hintergrund nötig. Die Situationen „blauer Pullover vor blauem Hintergrund“ oder bei der Gestenerkennung „Hand vor hautfarbenem Hintergrund“ stellen die Methoden der Bildverarbeitung vor eine schwierige Aufgabe, wohingegen dieser Umstand die Messung der Infrarot-Tiefenkamera nicht beeinflusst. Auch schlecht oder gar nicht beleuchtete Umgebungen sind für die Tiefenkamera kein Problem. Will man beispielsweise mit einer Anwendung durch

Gesten interagieren und der Raum ist zur besseren Sichtbarkeit der Anwendung abgedunkelt, so kann eine Tiefenkamera ohne Weiteres eingesetzt werden.

Andersherum, wenn extremer Sonneneinfall herrscht, wird auch die Tiefenkamera gestört. Zwei grundsätzliche Auswirkungen sind zu beobachten. Zum Einen verursachen die durch das starke Hintergrundlicht erzeugten, zusätzlichen Elektronen ein höheres „photon shot noise“ (Photonenrauschen) und beeinflussen somit negativ die Genauigkeit der gemessenen Tiefenwerte. Zum Anderen überfluten die aus dem konstanten Hintergrundlicht photogenerierten Elektronen die Pixel und überlasten deren Kapazität somit völlig. Daran, diese Auswirkungen zu minimieren, wird jedoch zur Zeit schon gearbeitet (vgl. Kapitel 2.4).
Das Problem der „Überstrahlung“ besteht zudem nicht allein für die time-of-flight Tiefenkamera, sondern auch farbbasierte Verfahren sind davon betroffen. Bei enormer Sonneneinstrahlung bzw. übermäßigem Hintergrundlicht ergeben sich ganz neue Farben und Farbabstufungen, sowie unbrauchbare Reflexionen können auftreten, was dann die Verarbeitung und Erkennung beeinflusst.

Ein nicht unerhebliches Problem ist zudem die Korrespondenzfindung. Dabei gibt es zwei wesentliche Punkte, die zu beachten sind:

1. *Sichtbarkeit bzw. Verdeckung:* Zunächst muss die *Selbstverdeckung* berücksichtigt werden. Immer ein Teil der Punkte des Modells sind durch den Rest des Modells verdeckt, nämlich diese, deren Normalen (vom Dreiecksnetz ausgehend) von der Kamera weg zeigen. Außerdem können sie am Rand des Sichtfeldes aus dem sichtbaren Bereich herausragen und somit den *Range of View* verlassen. Diese Punkte sind aus diesem Grund für die „Verheiratung“ des Modells mit der Punktwolke irrelevant geworden, da die Kamera auch nur die „Vorderseite“ der Szene sieht (sozusagen $2\frac{1}{2}$D). Sie müssen allerdings zunächst auch als solche erkannt werden. Bei dem vorliegenden Ansatz wird die Sichtbarkeit im Randbereich berücksichtigt, die Orientierung der Normalen jedoch noch nicht betrachtet und ausgenutzt.

2. *der Kometenschweif:* Durch das Auftreten des Kometenschweifs - die an den Objekträndern nach hinten ausstrahlenden Tiefenwerte (vgl. 2.3) - ergeben sich Punkte, die nicht dem Modell entsprechen. Der Versuch, Diesem mittels eines Medianfilters entgegenzuwirken, brachte keine wesentliche Verbesserung. Es stellte sich allerdings heraus, dass man durch die Festlegung - 1.Hauptachse zeigt in die Tiefe, 2.Hauptachse bestimmt die Zeigerichtung des Arms und 3.Hauptachse die „Dicke“ - die Auswirkungen des Kometenschweifs auf die Erkennung minimieren bzw. vernachlässigen konnte.

Des Weiteren ist die Wahl eines guten Modells wichtig. Das hier verwendete Handmodell ist besonders geeignet, da es einer anatomisch richtig strukturierten Hand entspricht und trotzdem dank seiner geringen Auflösung für Echtzeitanwendungen nutzbar ist. Es kann animiert und somit in die gemessene Punktwolke eingepasst werden.

Sind einmal die restlichen (oder zumindest einige weitere) Gelenkwinkel herausgefunden, so ist die genaue Erkennung, welche Geste (eventuell aus einem vorgegebenen Katalog) gezeigt wurde, nicht mehr so kompliziert. Doch mit dem schwierigen Teil, ein Skelett oder gar ganzes Modell in eine Punktwolke zu legen und daraus Gelenkwinkel zu berechnen, damit hat sich diese Arbeit beschäftigt.

7 Fazit und Ausblick

Im Rahmen dieser Arbeit wurde eine prototypische Gestenerkenneung unter Verwendung von Tiefenbildern konzipiert und umgesetzt. Dabei bedeutete Gestenerkennung nicht, eine spezifische Geste als solche zu identifizieren, sondern sich die Hand mit ihren Bewegungen für die Mensch-Maschine-Interaktion nutzbar zu machen. Das hier entwickelte System ist in der Lage 7 DOFs (Translation (t_x,t_y,t_z), Rotation (ϕ_x, ϕ_y, ϕ_y) und Skalierung) einer menschlichen Handbewegungen modellbasiert zu erkennen. Dabei wurde mit einem 3.00GHz Intel Pentium 4 mit 2GByte Speicher eine Frame-Rate von 2-3fps erreicht.

Nach eingehender Beschäftigung mit der verwendeten Kamera und dem Studium vieler im Bereich Gestenerkennung veröffentlichter Arbeiten, konnte ein geeignetes Konzept entwickelt werden, um die gestellte Aufgabe zu realisieren. Die Umsetzung stellte dabei eine große Herausforderung dar. Die Kamera musste nicht nur kalibriert werden, damit brauchbare Messwerte erlangt werden konnten, aus diesen mussten darüber hinaus noch reale mm-Werte und dadurch 3D-Positionen ermittelt werden. Durch die Interpretation der Messwerte als Punktwolke konnten mittels der Hauptkomponentenanalyse der Schwerpunkt und die Hauptausdehnungsachsen der gemessenen Tiefendaten ermittelt werden, was wiederum einen guten Anhaltspunkt für die genauere Positionierung des Modells bot. Die Illustrationen der Zwischenergebnisse - die Darstellung der Messwerte als Punktwolke zusammen mit den berechneten Hauptachsen, sowie die Visualisierung als Mesh - zeigten noch vorhandene Ungenauigkeiten der Messwerte, aber auch Lösungschancen auf. Durch Beispiele wurde dabei deutlich gemacht, welche Interaktionsmöglichkeiten auch zu diesem Zeitpunkt schon vorstellbar waren. Daraufhin folgte das modellbasierte Feinmatching. Ein dreidimensionales, naturgetreues Handmodell sollte in die Punktwolke eingepasst werden. Die daraus sich ergebenen Fragestellungen (Korrespondenz- sowie Sichtbarkeitsproblem) mussten analysiert und gelöst werden. Da der Abstand des Modells zur Punktwolke mininiert werden sollte, wurden die reinen Punktabstände als Maß dafür festgelegt und eine zugehörige Kostenfunktion erstellt. Außerdem wurden genaue Kriterien für die Sichtbarkeit der Modellpunkte definiert. Schließlich konnten globale Translation, Rotation und Skalierung ermittelt werden, um das Modell bestmöglich in die Punktwolke hineinzulegen.

Betrachtet man die Frage, ob man eventuell andere Sensoren zur Unterstützung einsetzen kann oder muss, so kommt man zu dem Schluss, dass man natürlich immer andere Sensoren zusätzlich einsetzen kann, dies bis zu diesem Zeitpunkt jedoch eindeutig (noch) nicht nötig war.
Die Verwendung eines Datenhandschuhs würde ohnehin keine Vereinfachung der Erkennung ergeben. Zunächst einmal müsste der Handschuh genau ka-

libriert werden, um brauchbare Werte zu erlangen, was jedoch nicht trivial ist. Des Weiteren erhält man nach der Kalibrierung zwar einige Winkel der Fingergelenke, die sechs Freiheitsgrade des Handgelenks können damit allerdings nicht ermittelt werden. Dafür müsste ein zusätzlicher Tracker angebracht werden. Außerdem würde der Anwender dann notwendigerweise verkabelt, was ihn in seinen Aktionen (weiter) einschränkt. Die zusätzliche Nutzung einer optischen Kamera und Methoden der Farbbildverarbeitung würden an dieser Stelle hingegen einen gewissen Vorteil erbringen. Würde man mit Hilfe der Tiefendaten die Hand entsprechend aus dem Farbbild extrahieren, könnten durch die Bildverarbeitung beispielsweise Falten in der Hand erkannt werden, die Rückschlüsse auf eine Krümmung des jeweiligen Gelenks zulassen.

Bei weiterführenden Schritten müsste der Hauptaugenmerk auf die Bestimmung weiterer Gelenkwinkel gerichtet werden. Dabei ist es denkbar, dass Verfahren wie ICP oder die Verwendung der Hausdorff-Distanz zum Einsatz kommen, oder aber auch der zusätzliche Einsatz einer Farbkamera und bildverarbeitender Methoden. Außerdem müssten bei der Bestimmung der Sichtbarkeit der Modellpunkte die Normalen noch mit einbezogen werden, wodurch sich zusätzlich die Anzahl der Abstandsberechnungen noch einmal reduzieren würde. Des Weiteren ist es erdenklich, dass bei der näheren Bestimmung der anderen Gelenke und ihrer Winkel nicht bei jeder Iteration die Abstände aller sichtbaren Modellpunkte von sämtlichen Punkten der Wolke berechnet werden, sondern bei der Feinabstimmung nur noch die Messpunkte in die Berechnung eingehen, die laut den vorherigen Abstandsbestimmungen nahe der Hand waren.

Will man zukünftig die Gelenkwinkel für das Handgelenk und die Finger ermitteln, so benötigt man auch ein artikuliertes Modell bzw. darunterliegendes Skelett. In diese Richtung wurden schon Vorarbeiten geleistet, welche jedoch aus Zeitgründen nicht abgeschlossen werden konnten. Zunächst wurde ein passendes Skelett erstellt. Durch Vermessung des eigenen Unterarms und der Hand konnten Diese millimetergenau nachgebildet werden. Das entstandene Skelett besteht aus den einzelnen Geometrien für Unterarm, Handfläche und Finger, die wiederum durch DCS-Knoten, die die Gelenke repräsentieren, verbunden sind (vgl. Abb. 20, 21 und 22).

Die DCS-Knoten enthaltend die Matrizen, die sich entsprechend der Gelenkwinkel auf die darunterliegenden Geometrien auswirken. Die Anzahl der möglichen Freiheitsgrade (degrees of freedom, DOFs) und ihrer Größenordnung je Gelenk müssen dabei gut überlegt sein. So dürfen sich die Finger beispielsweise nicht zum Handrücken hin „verbiegen“ lassen, da dies anatomisch auch nicht möglich ist. Für eine erste Erkennung mit Hilfe eines artikulierten Modells werden dabei nicht sämtliche Freiheitsgrade einer realen Hand (26 DOFs, ohne die möglichen Gelenkwinkel zwischen den einzelnen Knochen der Handfläche) betrachtet, sondern nur die 6 DOFs im

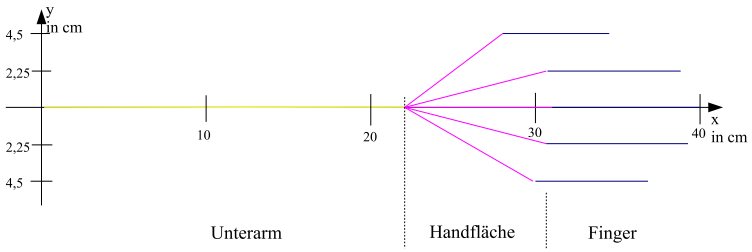

Abbildung 20: **Skelett (Zeichnung)**

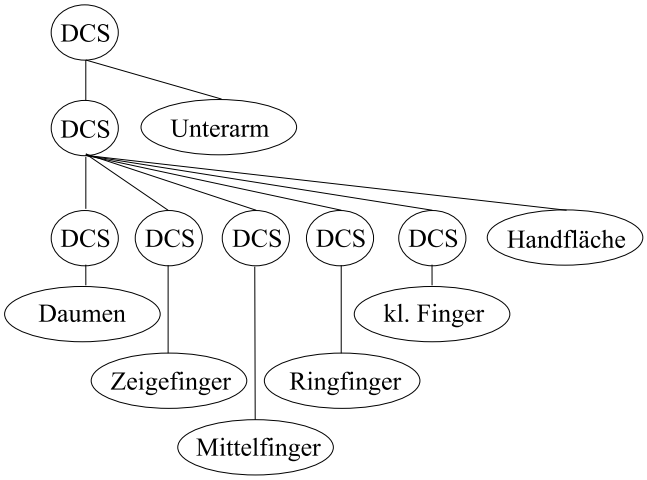

Abbildung 21: **Szenengraph der artikulierten Arm-Hand-Geometrie**

Handgelenk (+ 1 DOF für die Skalierung) sowie jeweils ein Freiheitsgrad pro Finger. Eine geeignete Festlegung dieser 12 Freiheitsgrade kann Tabelle 2 entnommen werden.

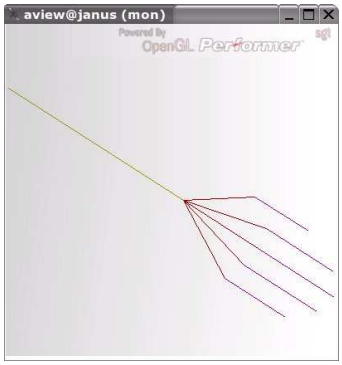

Abbildung 22: **Skelett im Programm**

Tabelle 2: **Freiheitsgrade je Gelenk**

Gelenk	Handgelenk	Finger
Transl. in X-Richtung	beliebig	-
Transl. in Y-Richtung	beliebig	-
Transl. in Z-Richtung	beliebig	-
Rot. um X-Achse	-90° bis +90°	-
Rot. um Y-Achse	-90 bis +90°	0° bis +90°
Rot. um Z-Achse	-45° bis +45°	-
Skalierung	beliebig	-

A Berechnungen

Für alle im Folgenden angestellten Berechnungen gilt:

$s = \begin{pmatrix} u \\ v \end{pmatrix}$	Kamera-Koordinaten
$s' = \begin{pmatrix} x \\ y \end{pmatrix}$	3D-Koordinaten
f	Brennweite der Kamera
d	gemessene Distanz
$k = \sqrt{s^2 + f^2}$	Abstand der Pixel-Position zum Brennpunkt, wobei $s^2 = u^2 + v^2$

Herleitung von z:

$$
\begin{aligned}
\tfrac{z-f}{f} &= \tfrac{d-k}{k} && | \cdot f \\
z - f &= \left(\tfrac{d-k}{k} \cdot f\right) && | + f \\
z &= \left(\tfrac{d-k}{k} \cdot f\right) + f && | k = \sqrt{s^2 + f^2} \\
&= \left(\tfrac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}} \cdot f\right) + f \\
&= \left(\left(\tfrac{d}{\sqrt{s^2+f^2}} - 1\right) \cdot f\right) + f && | s^2 = u^2 + v^2 \\
&= \left(\left(\tfrac{d}{\sqrt{u^2+v^2+f2}} - 1\right) \cdot f\right) + f \\
&= f \cdot \left(\tfrac{d}{\sqrt{u^2+v^2+f^2}} - 1 + 1\right) \\
z &= f \cdot \tfrac{d}{\sqrt{u^2+v^2+f^2}} \quad q.e.d.
\end{aligned}
\tag{5}
$$

Herleitung von x und y:

$$
\begin{aligned}
\tfrac{-s'}{d-k} &= \tfrac{s}{k} && |\cdot(d-k) \\
-s' &= \tfrac{s\cdot(d-k)}{k} \\
&= \tfrac{sd-sk}{k} \\
&= \tfrac{sd}{k} - s && |k = \sqrt{s^2+f^2} \\
&= s \cdot \tfrac{d}{\sqrt{s^2+f^2}} - s && |s^2 = u^2+v^2 \quad |s = \begin{pmatrix} u \\ v \end{pmatrix} \\
&= \begin{pmatrix} u \\ v \end{pmatrix} \cdot \tfrac{d}{\sqrt{u^2+v^2+f^2}} - \begin{pmatrix} u \\ v \end{pmatrix} && |\cdot(-1) \quad |s' = \begin{pmatrix} x \\ y \end{pmatrix} \\
\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} -u \\ -v \end{pmatrix} \cdot \tfrac{d}{\sqrt{u^2+v^2+f^2}} \quad q.e.d.
\end{aligned}
\tag{6}
$$

genaue Bestimmung des Fehlers ϑ_z für z:

$$
\begin{aligned}
&\text{Formel für z:} && z &&= f \cdot \tfrac{d}{\sqrt{u^2+v^2+f^2}} \\
&\text{Näherung (für } u^2+v^2 \ll f^2\text{):} && z_n &&\approx d \\
&\text{relativer Fehler für z:} && \vartheta_z &&:= \tfrac{|z-z_n|}{z} \\
& && &&= \frac{\left|\frac{f\cdot d}{\sqrt{u^2+v^2+f^2}} - d\right|}{\frac{f\cdot d}{\sqrt{u^2+v^2+f^2}}} \\
& && &&= \left|\frac{\sqrt{u^2+v^2+f^2}}{f\cdot d} \cdot \left(\frac{f\cdot d}{\sqrt{u^2+v^2+f^2}} - d\right)\right| \\
& && &&= \left|1 - \frac{d\cdot\sqrt{u^2+v^2+f^2}}{f\cdot d}\right| \\
& && &&= \left|1 - \frac{\sqrt{u^2+v^2+f^2}}{f}\right|
\end{aligned}
\tag{7}
$$

genaue Bestimmung des Fehlers ϑ_{xy} für x und y:

Formel für x und y:

$$\begin{aligned}
\frac{s}{k} &= \frac{s'}{d-k} \\
s' &= s \cdot \frac{d-k}{k} && |k = \sqrt{s^2+f^2} \\
&= s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}}
\end{aligned}$$

Näherung (für $u^2+v^2 \ll f^2$):

$$\begin{aligned}
\frac{s}{f} &\approx \frac{s'_n}{z-f} \\
s'_n &\approx s \cdot \frac{z-f}{f} && |Ann.: z \approx d \\
&\approx s \cdot \frac{d-f}{f}
\end{aligned}$$

relativer Fehler für x und y:

$$\begin{aligned}
\vartheta_{xy} &:= \frac{|s'-s'_n|}{s'} \\
&= \frac{\left| s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}} - s \cdot \frac{d-f}{f} \right|}{s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}}} \\
&= \left| 1 - \frac{s \cdot \frac{d-f}{f}}{s \cdot \frac{d-\sqrt{s^2+f^2}}{\sqrt{s^2+f^2}}} \right| \\
&= \left| 1 - s \cdot \frac{d-f}{f} \cdot \frac{\sqrt{s^2+f^2}}{s \cdot (d-\sqrt{s^2+f^2})} \right| \\
&= \left| 1 - \frac{s \cdot (d-f) \cdot \sqrt{s^2+f^2}}{f \cdot s \cdot (d-\sqrt{s^2+f^2})} \right| \\
&= \left| 1 - \frac{(s \cdot d - s \cdot f) \cdot \sqrt{s^2+f^2}}{d \cdot f \cdot s - f \cdot s \cdot \sqrt{s^2+f^2}} \right| \\
&= \left| 1 - \frac{s \cdot d \cdot \sqrt{s^2+f^2} - s \cdot f \cdot \sqrt{s^2+f^2}}{s \cdot d \cdot f - s \cdot f \cdot \sqrt{s^2+f^2}} \right| \\
&\approx \left| 1 - \frac{s \cdot d \cdot \sqrt{s^2+f^2}}{s \cdot d \cdot f} \right| \\
&\approx \left| 1 - \frac{\sqrt{s^2+f^2}}{f} \right| = \left| 1 - \frac{\sqrt{u^2+v^2+f^2}}{f} \right|
\end{aligned} \tag{8}$$

B Visualisierungen

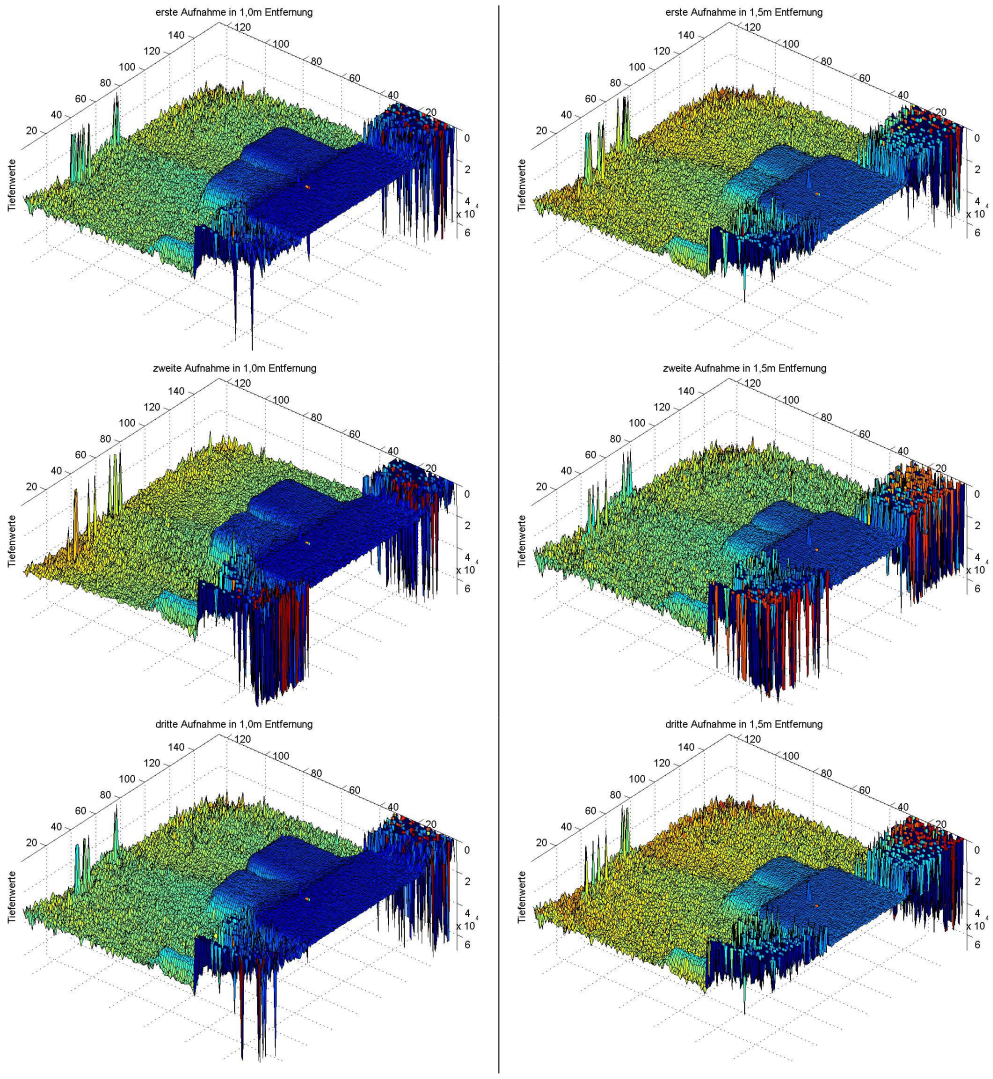

Abbildung 23: **Messwerte in MatLab visualisiert:** Messergebnisse bei 1,0m Entfernung (links) und bei 1,5m Entfernung (rechts)

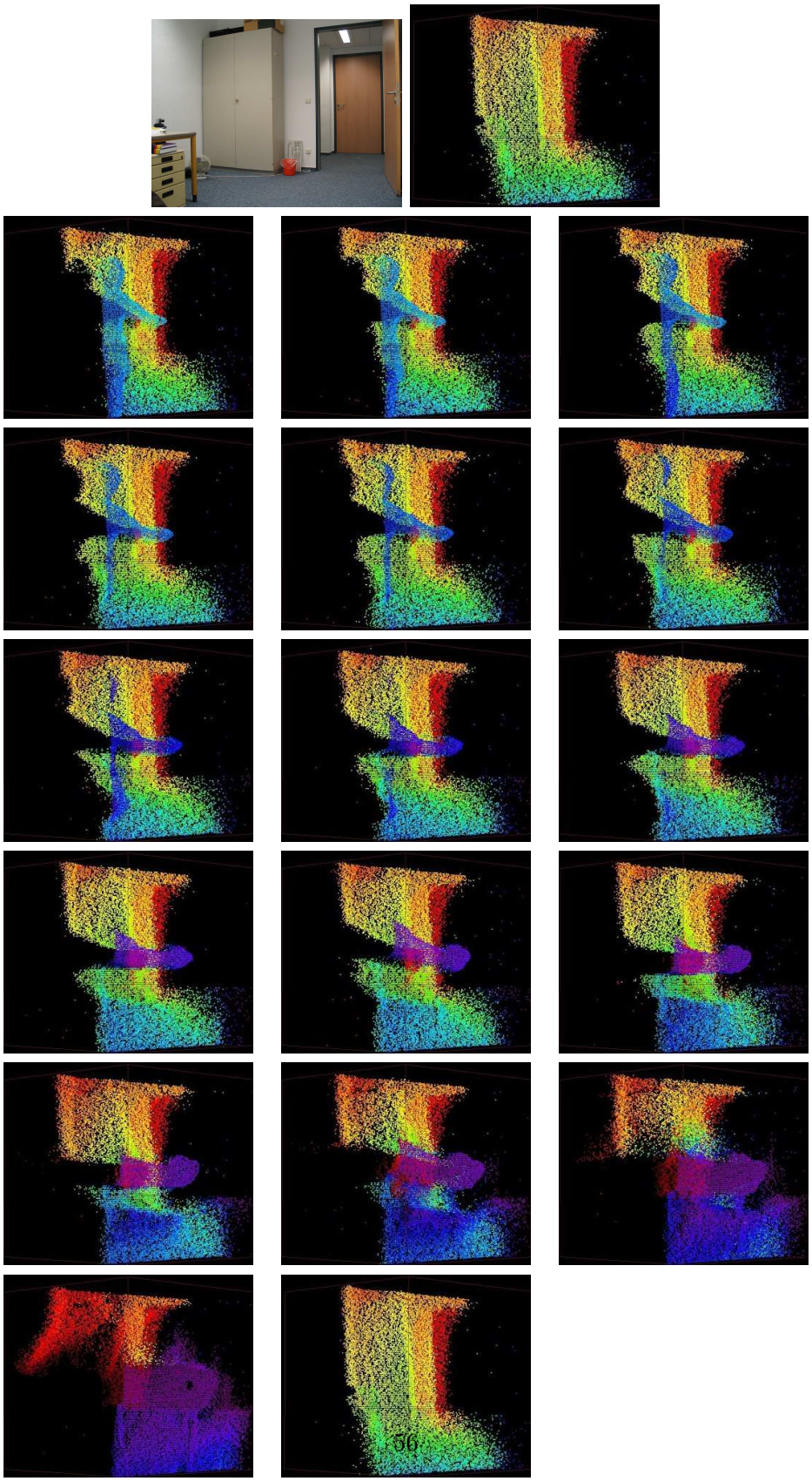

Abbildung 24: **versch. Tiefen - mit dem SR2-Tool aufgenommen:** Die oberste Zeile zeigt den Ausgangszustand. Die folgenden Aufnahmen zeigen die Hand in einer Entfernung von 2,0m bis 0,5m (0,1m-Schritte). Zu Vergleichszwecken ist am Ende noch einmal der Ausgangszustand visualisiert.

Literatur

[AHS03] Albrecht, Irene ; Haber, Joerg ; Seidel, Hans-Peter: Construction and animation of anatomically based human hand models. In: *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, 2003, S. 98–109

[AS03] Athitsos, Vassilis ; Sclaroff, Stan: Estimating 3D hand pose from a cluttered image. In: *CVPR 2003: Computer Vision and Pattern Recognition Conference, Madison, WI, USA, 18-20 June 2003, vol.2*, 2003, S. II–432–9

[BHE+02] Bebis, Dr. G. ; Harris, Dr. F. ; Erol, Dr. A. ; Yi, Beifang ; Martinez, Javier ; Hernandez-Usabiaga, Jorge ; Fritzinger, Scott: Development of a Nationally Competitive Program in Computer Vision Technologies for Effective Human-Computer Interaction in Virtual Environments / ComputerVisionLab, University of Nevada Reno. 2002. – Forschungsbericht

[BI05] Borst, Christoph W. ; Indugula, Arun P.: Realistiv Virtual Grasping. In: *Proceedings, IEEE Virtual reality 2005, Bonn, Germany, 12-16 March 2005* IEEE Computer Soc., 2005, S. 91–8

[BIF05] Benko, Hrvoje ; Ishak, Edward W. ; Feiner, Steven: Cross-Dimensional Gestural Interaction Techniques for Hybrid Immersive Environments. In: *Proceedings, IEEE Virtual reality 2005, Bonn, Germany, 12-16 March 2005* IEEE Computer Soc., 2005, S. 209–16

[BKK04] Bendels, Gerhard H. ; Kahlesz, Ferenc ; Klein, Reinhard: Towards the next generation of 3D content creation. In: *AVI '04: Proceedings of the working conference on Advanced visual interface*, 2004, S. 283–89

[BKMG04] Bray, Matthieu ; Koller-Meier, Esther ; Gool, Luc van: Smart particle filtering for 3D hand tracking. In: *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 17-19 May 2004*, 2004, S. 675–80

[BLL02] Bretzner, Lars ; Laptev, Ivan ; Lindeberg, Tony: Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In: *Proceedings Fifth IEEE*

International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 20-21 May 2002 IEEE, 2002, S. 423–8

[BOL+05] Büttgen, Bernhard ; Oggier, Thierry ; Lehmann, Michael ; Kaufmann, Rolf ; Lustenberger, Felix: CCD/CMOS Lock-In Pixel for Range Imaging: Challenges, Limitations and State-of-the-Art. In: *Proceedings, 1st Range Imaging Research Day, September 8/9, 2005 at ETH Zurich Switzerland*, 2005, S. 21–32

[BPV04] Buades, Jose M. ; Perales, Francisco J. ; Varona, Javier: Real time segmentation and tracking of face and hands in VR applications. In: *Articulated Motion and Deformable Objects. Third International Workshop, AMDO 2004. Proceedings, Palma de Mallorca, Spain, 22-24 Sept. 2004*, 2004, S. 259–68

[CA03] Couturier, Jean-Michel ; Arfib, Daniel: Pointing Finger:Using Multiple Direct Interactions with Visual Objects to Perform Music. In: *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada*, 2003, S. 184–7

[CBB+98] Corradini, Andrea ; Braumann, Ulf-Dietrich ; Brakensiek, Anja ; Krabbes, Markus ; Boehme, Hans-Joachim ; Gross, Horst-Michael: Visual person localization with dynamic neural fields: towards a gesture recognition system. In: *Proceedings of the 10th Italian Workshop on Neural Nets WIRN VIETRI-98, Salerno, Italy, 21-23 May 1998*, 1998, S. 201–6

[CBH03] Corso, Jason J. ; Burschka, Darius ; Hager, Gregory D.: The 4D Touchpad: Unencumbered HCI With VICs. In: *Proceedings of CVPRHCI*, 2003

[CGA05] *The CGAL home page.* URL http://www.cgal.org/, 2005

[CS04] Cui, Jinshi ; Sun, Zengqi: Visual Hand Motion Capture for Guiding A Dexterous Hand. In: *Proceedings, Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul,Korea*, 2004, S. 729–34

[CSE05] CSEM: *Swiss Ranger SR-2 Datasheet.* www.swissranger.ch/pdf/SR-2_DataSheet.pdf, 2005

[Dre05] Dreiseitl, Stephan: *Skriptum zu MAT3 WS2004/05.* 2005

[DWT04] Derpanis, Konstantinos G. ; Wildes, Richard P. ; Tsotsos, John K.: Hand Gesture Recognition within a Linguistics-Based Framework. In: *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, 2004, S. 282–96

[EGG+03] Eisenstein, Jacob ; Ghandeharizadeh, Shahram ; Golubchik, Leana ; Shahabi, Cyrus ; Yan, Donghui ; Zimmermann, Roger: Device independence and extensibility in gesture recognition. In: *Proceedings IEEE Virtual Reality 2003, Los Angeles, CA, USA, 22-26 March 2003 * Los Alamitos, CA, USA: IEEE Comput. Soc, 2003, p 207-14* IEEE Comput. Soc, 2003, S. 207–14

[EGH+01] Eisenstein, J. ; Ghandeharizadeh, S. ; Huang, L. ; Shahabi, C. ; Shanbhag, G. ; Zimmermann, R.: Analysis of clustering techniques to detect hand signs. In: *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001, Hong Kong, China, 2-4 May 2001*, 2001, S. 259–62

[Eye05] *EyeToy Homepage*. http://www.eyetoy.com/, 2005

[Gut04] Gut, Oliver: *Untersuchungen des 3D-Sensors SwissRanger*, ETH Zrich, Diplomarbeit, 2004

[HB01] Hardenberg, Christian von ; Bérard, François: Bare-Hand Human-Computer Interaction. In: *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, 2001, S. 1–8

[HJ01] Huang, Chung-Lin ; Jeng, Sheng-Hung: A model-based hand gesture recognition system. In: *Machine Vision and Applications* 12 (2001), Nr. 5, S. 243 – 258

[Jr.99] Jr., Joseph J. L.: *Whole-Hand and Speech Input in Virtual Environments*, Brown University, Diss., 1999

[KI05a] Kahlmann, Timo ; Ingensand, Hilmar: Calibration and improvements of the high-resolution range-imaging camera SwissRanger. In: *Proceedings of SPIE Vol. 5665; Videometrics VIII; 16-20 January 2005, San Jose, California, USA*, 2005

[KI05b] Kahlmann, Timo ; Ingensand, Hilmar: Range Imaging Sensor Properties and Calibration. In: *Proceedings, 1st Range Imaging Research Day, September 8/9, 2005 at ETH Zurich Switzerland*, 2005, S. 71–80

[KR04] Kong, W. W. ; Ranganath, Surendra: 3-D Hand Trajectory Recognition for Signing Exact English. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, 2004, S. 535–540

[KT04] Kölsch, Mathias ; Turk, Matthew: Robust Hand Detection. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, 2004, S. 614–619

[KZK04] Kahlesz, Ferenc ; Zachmann, Gabriel ; Klein, Reinhard: Visual-fidelity dataglove calibration. In: *Computer Graphics International 2004 (CGI 2004)* IEEE Comput. Soc, 2004, S. 403–10

[LBT02] Lenman, Søren ; Bretzner, Lars ; Thuresson, Bjørn: Using marking menus to develop command sets for computer vision based hand gesture interfaces. In: *NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction*, 2002, S. 239–42

[LC04] Lee, Sung U. ; Cohen, Isaac: 3D Hand Reconstruction from a Monocular View. In: *ICPR (3)*, 2004, S. 310–313

[LF04] Liu, Xia ; Fujimura, Kikuo: Hand gesture recognition using depth data. In: *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 17-19 May 2004* IEEE Comput. Soc, 2004, S. 529–34

[Lie05] Lien, Cheng-Chang: A scalable model-based hand posture analysis system. In: *Machine Vision and Applications* 16 (2005), Nr. 3, S. 157 – 169

[LRKL04] L.Hernandez-Rebollar, Jose ; Kyriakopoulos, Nicholas ; Lindeman, Robert W.: A new instrumented approach for translating American Sign Language into sound and text. In: *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 17-19 May 2004 * Los Alamitos, CA, USA: IEEE Comput. Soc, 2004, p 547-52*, 2004, S. 547–52

[LS04] Licsar, Attila ; Sziranyi, Tamas: Hand gesture recognition in camera-projector system. In: *Computer Vision in Human-Computer Interaction. ECCV 2004 Workshop in HCI. Proceedings, Prague, Czech Republic, 16 May 2004*, 2004, S. 83–93

[LWH04] Lin, John Y. ; Wu, Ying ; Huang, Thomas S.: 3D Model-Based Hand Tracking Using Stochastic Direct Search Method. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, 2004, S. 693–8

[MCA02] Mammen, James P. ; Chaudhuri, Subhasis ; Agrawal, Tushar: A Two-stage Scheme for Dynamic Hand Gesture Recognition / Department of Electrical Engg. Indian Institute of Technology, Bombay, India-400076. 2002. – Forschungsbericht

[MHRS+04] McGuire, R. M. ; Hernandez-Rebollar, Jose ; Starner, Thad ; Henderson, Valerie ; Brashear, Helene ; Ross, Danielle S.: Towards a One-Way American Sign Language Translator. In: *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 17-19 May 2004*, 2004, S. 620–5

[MTM+01] Malassiotis, S. ; Tsalakanidou, F. ; Mavridis, N. ; Giagourta, V. ; Grammalidis, N. ; Strintzis, M. G.: A face and gesture recognition system based on an active stereo sensor. In: *Proceedings 2001 International Conference on Image Processing, Thessaloniki, Greece, 7-10 Oct. 2001, vol.3*, 2001, S. 955–8

[NW96] Nam, Yanghee ; Wohn, KwangYun: Recognition of Space-Time Hand-Gestures using Hidden Markov Model. In: *Proceedings of the ACM VRST96*, 1996, S. 51–8

[OB04] Ong, Eng-Jon ; Bowden, Richard: A Boosted Classifier Tree for Hand Shape Detection. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, 2004, S. 889–94

[OINO02] Ozeki, Motoyuki ; Itoh, Masatsugu ; Nakamura, Yuichi ; Ohta, Yuichi: Tracking hands and objects for an intelligent video production system. In: *Proceedings of 16th International Conference on Pattern Recognition, Quebec City, Que., Canada, 11-15 Aug. 2002, vol.3* IEEE Comput. Soc, 2002, S. 1011–14

[OMLR+03] Oggier, Thierry ; Michael Lehmann, Matthias S. ; Richter, Michael ; Metzler, Peter ; Lang, Graham ; Lustenberger, Felix ; Blanc, Nicolas: An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In: *SPIE, conference on optical system design, St. Etienne, September 2003*, 2003

[OR04] Ong, Sylvie C. W. ; Ranganath, Surendra: Deciphering Gestures with Layered Meanings and Signer Adaptation. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, 2004, S. 559–64

[PCI04] Perrin, Stephane ; Cassinelli, Alvaro ; Ishikawa, Masatoshi: Gesture recognition using laser-based tracking system. In: *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 17-19 May 2004* IEEE Comput. Soc, 2004, S. 541–6

[SLM+03] Starner, Thad ; Leibe, Bastian ; Minnen, David ; Westyn, Tracy ; Hurst, Amy ; Weeks, Justin: The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks. In: *Mach. Vis. Appl.* 14 (2003), Nr. 1, S. 59–71

[SS01] Shamaie, Atid ; Sutherland, Alistair: Graph-Based Matching of Occluded Hand Gestures. In: *30th Applied Image Pattern Recognition Workshop (AIPR 2001), Analysis and Understanding of Time Varying Imagery, 10-12 October 2001, Washington, DC, USA, Proceedings*, 2001, S. 67–73

[SS03] Shamaie, Atid ; Sutherland, Alistair: *Accurate Recognition of Large Number of Hand Gestures.* 2003

[STTC04] Stenger, B. ; Thayananthan, A. ; Torr, P.H.S. ; Cipolla, R.: Hand pose estimation using hierarchical detection. In: *Computer Vision in Human-Computer Interaction. ECCV 2004 Workshop in HCI. Proceedings, Prague, Czech Republic, 16 May 2004*, 2004, S. 105–16

[TAH+04] Theobalt, Christian ; Albrecht, Irene ; Haber, Jörg ; Magnor, Marcus ; Seidel, Hans-Peter: Pitching a baseball: tracking high-speed motion with multi-exposure images. In: *ACM Trans. Graph.* 23 (2004), Nr. 3, S. 540–547

[WADP97] Wren, Christopher R. ; Azarbayejani, Ali ; Darrell, Trevor ; Pentland, Alex P.: Pfinder: Real-Time Tracking of the Human Body. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (1997), Nr. 7, S. 780–5

[WB99] Wilson, Andrew D. ; Bobick, Aaron F.: Parametric Hidden Markov Models for gesture recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, USA * vol 21 (Sept. 1999), no 9*, 1999, S. 884–900