



FACHHOCHSCHUL-BACHELORSTUDIENGANG
AUTOMATISIERUNGSTECHNIK

GESTENERKENNUNG UNTER VERWENDUNG EINER TOF-KAMERA

ALS BACHELORARBEIT EINGEREICHT

zur Erlangung des akademischen Grades

Bachelor of Science in Engineering

Von

SANDBERGER GEORG

Juni 2011

Betreuung der Bachelorarbeit durch

Prof. (FH) Dipl. Ing. Kurt Niel

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt, die den benutzten Quellen entnommenen Stellen als solche kenntlich gemacht habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

.....
Georg Sandberger

St. Agatha, 30. Juni 2011

KURZFASSUNG

Die vorliegende Arbeit befasst sich mit kamerabasierter Gestenerkennung unter Verwendung einer Time-of-Flight (ToF) Kamera und dem Ziel, einen Roboterarm durch Handbewegungen zu steuern. Dabei wird die Bewegung der menschlichen Hand in einem kleinen, dreidimensionalen Raum von einer ToF-Kamera erfasst. Die Position eines fixen Punktes der Hand wird im entsprechenden Verhältnis in den Arbeitsraum des Roboterarms umgerechnet und als Tool-Center-Point vorgegeben. Dadurch kann die Gelenkstellung des Roboterarms berechnet werden.

Die Gestenerkennung der Hand erfolgt durch eine Tiefensegmentierung der 3D-Bilder der ToF-Kamera, wodurch die menschliche Hand vom Hintergrund getrennt und als Punktwolke im Raum dargestellt wird. Mit Hilfe von mathematischen Algorithmen werden die Daten analysiert. Zur Steuerung des Roboterarms wird die Principal Component Analyse verwendet, um die Hauptachsen der Punktwolke und somit die Ausrichtung der Hand zu ermitteln. Durch Translation, Rotation und Skalierung kann ein 3D-Handmodell mit Hilfe der Iterativ-Closest-Point-Methode und den zuvor ermittelten Hauptachsen über die 3D-Punktwolke der Hand gelegt werden. Dadurch kann ihre Position im Raum bestimmt werden.

INHALTSVERZEICHNIS

1 MOTIVATION	1
1.1 HINTERGRUND	1
1.2 ABGRENZUNG	3
2 TIME OF FLIGHT KAMERAS	4
2.1 LAUFZEITVERFAHREN	4
2.1.1 <i>Direkte Messung</i>	4
2.1.2 <i>Amplitudenmodulation</i>	5
2.1.3 <i>Frequenzmodulation</i>	6
2.2 TECHNISCHER AUFBAU	6
2.3 FUNKTIONSWEISE	7
2.5 HERSTELLER UND KOSTEN	8
3 GESTENERKENNUNG	10
3.1 DEFINITION	10
3.2 OPTISCHE GESTENERKENNUNG.....	10
3.3 ANFORDERUNGEN	11
3.4 TOF-KAMERADATEN	12
3.4.1 <i>Kalibrierung</i>	12
3.4.2 <i>Tiefensegmentierung</i>	12
3.4.3 <i>Principal Component Analysis (PCA)</i>	13
3.5 MODELLIERUNG	17
3.5.1 <i>Betrachterorientierte Modellierung</i>	17
3.5.2 <i>Objektorientierte/geometrische Modellierung</i>	17
3.6 KLASSIFIKATION	18
3.6.1 <i>Iterativ-Closest-Point Methode</i>	18
3.7 AUSWERTUNG	20
4 AUSBLICK	21
5 FAZIT UND DANK	22
6 VERZEICHNISSE	23
6.1 ABBILDUNGSVERZEICHNIS.....	23
6.2 TABELLENVERZEICHNIS.....	23
6.3 LITERATURVERZEICHNIS.....	24

Kapitel 1

Motivation

„Zu wissen, was man weiß, und zu wissen, was man tut, das ist Wissen.“

Konfuzius (*479 v. Chr.), chinesischer Philosoph

Kamerabasierte Gestenerkennung ist die Grundlage der Gestensteuerung, die in den letzten Jahren eine immer häufiger verwendete Form der Mensch-Maschine-Interaktion geworden ist. Mit Hilfe der Gestensteuerung ist es möglich eine einfache und intuitive Interaktionsform zu erschaffen, die es erlaubt, ohne zusätzliche Hilfsmittel zahlreiche Anwendungen zu bedienen. Durch diese natürliche Form der Kommunikation hat die kamerabasierte Gestensteuerung eine hohe Akzeptanz und wird z.B. bei verschiedenen Spielekonsolen, wie der Xbox360 von Microsoft Inc.¹, verwendet.

1.1 Hintergrund

Im Zuge des Berufspraktikums an der FH-Wels, Fakultät für Technik und Umweltwissenschaften, wurde in Zusammenarbeit mit dem RoboRacingTeam² der FH Wels nach einer Lösung gesucht, eine intuitive Steuerung für einen Roboterarm zu entwickeln (siehe Abb. 1.1). Diese sollte portabel und platzsparend sein, da der Roboterarm weltweit in verschiedenen Wettkämpfen eingesetzt wird.

¹ www.microsoft.com

² www.rrt.fh-wels.at



Abbildung 1.1: Roboterarm der FH-Wels

Von der Gestensteuerung einer Spielekonsole inspiriert, wurde die Idee geboren, den Roboterarm durch eine kamerabasierte Gestensteuerung zu manipulieren. Durch Gestenerkennung sollte die Bewegung der menschlichen Hand in einem kleinen, dreidimensionalen Raum von einer ToF-Kamera erfasst werden. Die Position eines fixen Punktes auf der Hand wird im entsprechenden Verhältnis in den Arbeitsraum des Roboterarms umgerechnet und als Tool-Center-Point (TCP) vorgegeben. Dabei werden Translation (x , y , z) und Rotation (α , β , γ) der Handstellung gemessen. Diese ergeben den TCP:

$$TCP = (x \ y \ z \ \alpha \ \beta \ \gamma)^T$$

Mit Hilfe der inversen Kinematik kann durch Vorgabe des Tool-Center-Points auf die Stellung der Gelenke des Roboterarms zurückgerechnet werden (Abb. 1.2).

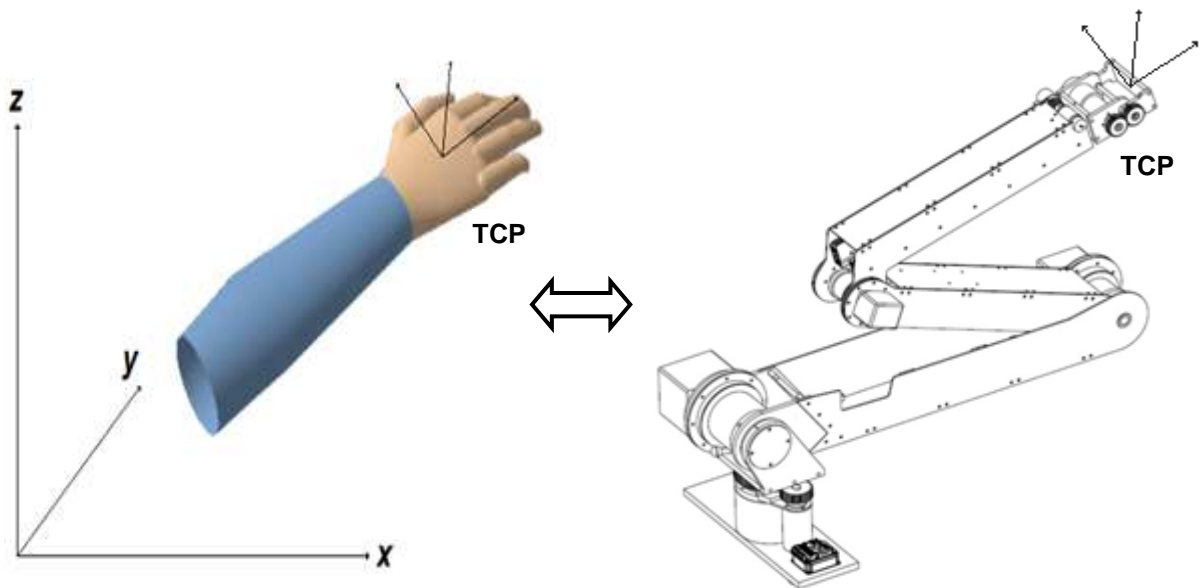


Abbildung 1.2: Manipulation des Roboterarms durch Armbewegungen

Eine Arbeit mit ähnlicher Motivation jedoch komplett unterschiedlicher Technik wurde bereits an der FH-Wels realisiert. Dabei wurde mit dem 3D-Verfahren der Stereoskopie gearbeitet (vergleiche [1] und [2]).

1.2 Abgrenzung

Die vorliegende Arbeit befasst sich mit dem Thema der kamerabasierten Gestenerkennung und dient als theoretische Grundlage und Recherche für die Umsetzung des Manipulators für den Roboterarm. Die Umsetzung des Projektes ist in dieser Arbeit nicht enthalten.

Kapitel 2

Time-of-Flight Kamera

Time-of-Flight (ToF) Kameras sind 3D-Kamerasysteme, die in Echtzeit Distanzen zu einem Objekt messen. Die Distanzmessung wird mit Hilfe des Laufzeitverfahrens durchgeführt. Die Kamera sendet einen Lichtimpuls und misst die Zeit, die das Licht benötigt um wieder zur Kamera zurück zu gelangen. Dadurch kann die Tiefeninformation gewonnen werden und die Objekte können dreidimensional dargestellt werden.

2.1 Laufzeitverfahren

Das Laufzeitverfahren beschreibt eine Technik zum Messen von Entfernungen. Dabei wird ein Referenzsignal gesendet, welches vom Objekt reflektiert wird. Das Echo wird von einem Detektor erfasst und verarbeitet. Neben der direkten Laufzeitmessung gibt es auch Laufzeitmessungen mit Hilfe der Frequenz- und Amplitudenmodulation.

2.1.1 Direkte Messung

Bei der direkten Laufzeitmessung geht man von der Beziehung

$$s = v * t \quad (1.1)$$

aus. Für die Tiefenmessung wird ein Signal gesendet, welches in der Zeit t eine Strecke vom Sender zum Objekt und wieder retour zurücklegt. Bei bekannter Signalgeschwindigkeit v ergibt sich folgende Beziehung für die Tiefe R :

$$R = \frac{v*t}{2} \quad (1.2)$$

Typischerweise werden bei Tiefensensoren Radiowellen, Ultraschall sowie Lichtimpulse als Signalarten verwendet. Dabei gilt, je schneller die Signalart, desto höhere Anforderungen ergeben sich für die Zeitmessung. Wegen der guten Signaleigenschaften und Echtzeitfähigkeit werden in der Bildverarbeitung meist Lichtimpulse eingesetzt. [3]

Bei einer Lichtgeschwindigkeit von ca. $c = 3 \cdot 10^8$ m/s ist für eine Tiefenauflösung im mm-Bereich eine extrem genaue Zeitmesselektronik erforderlich, die mit einer Genauigkeit von wenigen Picosekunden messen kann.

2.1.2 Amplitudenmodulation

Bei der Laufzeitmessung mit Hilfe der Amplitudenmodulation wird die Laufzeit indirekt ermittelt. Als Signalart wird ein Laser verwendet, dessen Lichtleistung mit einer Frequenz f_{AM} sinusförmig moduliert wird. Das reflektierte Licht hat die gleiche Frequenz, ist aber aufgrund der zurückgelegten Distanz um Winkel φ phasenverschoben.

Daraus ergibt sich folgende Formel für die Tiefe R:

$$R = \frac{c \cdot \varphi}{4 \cdot \pi \cdot f_{AM}} \quad (1.3)$$

Der Vorteil besteht darin, dass die Elektronik nicht die direkte und extrem kurze Laufzeit, sondern lediglich die Phasenverschiebung φ messen muss. Nachteil dieser Methode ist, dass die Phasenverschiebung nur relativ gemessen werden kann. Dadurch entsteht bei Tiefensensoren ein Eindeutigkeitsbereich R^* von einer halben Wellenlänge:

$$R^* = \frac{c}{2 \cdot f_{AM}} = \frac{\lambda_{AM}}{2} \quad (1.4)$$

Bei einer Modulationsfrequenz von 20MHz entsteht nach Formel 1.4 ein Eindeutigkeitsbereich von ca. 7,5m. Die Tiefenauflösung lässt sich erhöhen, indem eine hohe Frequenz oder eine hohe Phasenauflösung verwendet wird. [3]

2.1.3 Frequenzmodulation

Die Laufzeitmessung mit Hilfe der Frequenzmodulation arbeitet mit einer sinusförmigen oder triangulären Modellierung f_{FM} der Frequenz des Signals. Wie bei der Amplitudenmodulation werden meist Laserdioden verwendet, bei denen man durch Veränderung von Strom und Temperatur die Frequenz einfach verändert kann. Das verwendete Frequenzband bezeichnet man als Δf . Durch die Laufzeit ist die detektierte Frequenz f_d ungleich der referenzierten Frequenz f_r . Aus der Differenz ergibt sich die Schwebungsfrequenz f_s , mit der die Tiefe R aus Formel 1.5 ermittelt werden kann. [3]

$$R = \frac{c * f_s}{4 * f_{FM} * \Delta f} \quad (1.5)$$

2.2 Technischer Aufbau



Abbildung 2.1: ToF Kamera [4]

Emitter

Der Emitter sendet einen Lichtimpuls oder ein moduliertes Signal aus, das vom Objekt reflektiert wird. Die derzeit mögliche Beleuchtungsstärke beträgt ca. 1 Watt. Für ToF-Kamerasysteme werden überwiegend Infrarot-LEDs benutzt. [5]

Optik

Durch eine Optik wird das reflektierte Licht auf dem Sensor abgebildet. Ein optischer Bandpassfilter lässt nur die Wellenlänge durch, mit der auch die Beleuchtung arbeitet. Somit wird ein großer Teil des störenden Hintergrundlichtes eliminiert.

Sensor

Jedes Pixel des Sensors misst die Laufzeit bzw. Phasenverschiebung des Signals. Durch den komplizierteren Aufbau sind die Pixel im Vergleich zu Digitalkameras groß. Sie erreichen Seitenlängen bis zu 100 μm . Die bisher höchste Auflösung von ToF-Sensoren beträgt derzeit 204 \times 204 Pixel bei einer Kantenlänge von 45 μm . [5]

Elektronik

Anhand der analogen Daten, die der Sensor liefert, führt sie alle Berechnungen für die Abstandsdiagramme durch und steuert die Kommunikation mit dem angeschlossenen Rechner. Zusätzlich regelt sie die Ansteuerung der Beleuchtung und der Sensoren. Verschieben sich die Ansteuersignale nur minimal, bedeutet das eine erhebliche Verfälschung des Messergebnisses.

2.3 Funktionsweise

Derzeitiger Stand der Technik für die Funktionsweise von ToF-Kameras ist die Laufzeitmessung mit Hilfe der Amplitudenmodulation. Indem das Referenzsignal mit dem reflektierten Signal verglichen wird, wird die Phasenverschiebung φ ermittelt. Mit Formel 1.3 erhält man die Tiefe.

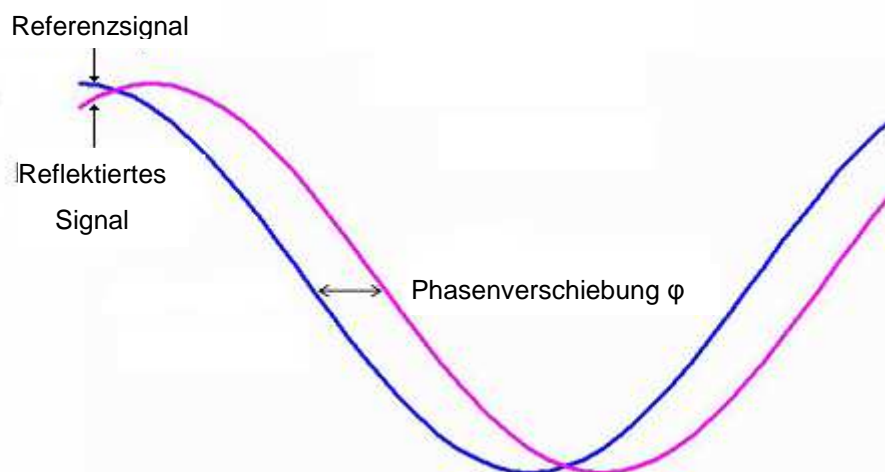


Abbildung 2.2: Amplitudenmodulation

Um nicht nur einen Punkt, sondern die gesamte Szene auf einmal zu vermessen, misst jeder Bildpunkt des Sensors die Laufzeit bzw. Phasenverschiebung des Signals. Durch diese Tiefeninformationen entsteht eine 3D-Punktwolke (Abbildung 2.3).

Zusätzlich wird noch ein dimensionsloser Wert - der sogenannte Amplitudenwert - in jedem Pixel berechnet. Dieser Wert quantifiziert grob, wie viel des aktiv ausgesandten Signales das Pixel erreicht hat. Dieser Anteil hängt im Wesentlichen von der Entfernung des Objektes und seinen Reflektionseigenschaften ab. In Abbildung 2.4 ist ein beispielhaftes 2D-Grauwertbild dargestellt, das mit Hilfe der Amplitudendaten gewonnen wurde.

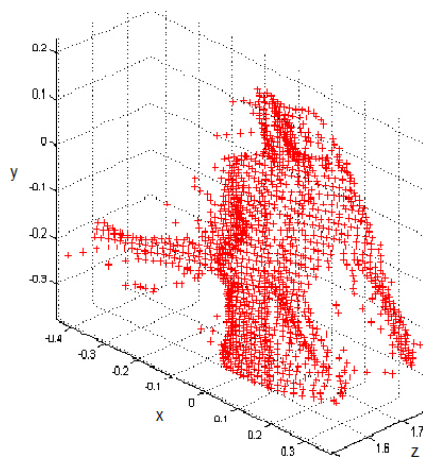


Abb. 2.3: 3D-Punktwolke [7]



Abb. 2.4: 2D-Grauwertbild [6]

Da jeder Grauwert und jeder 3D-Punkt eindeutig einem Pixel zugeordnet sind, kann bei der Entwicklung von Algorithmen übergangslos zwischen 2D- und 3D-Verfahren gewechselt werden. [6]

2.5 Hersteller und Kosten

In Tabelle 1.1 werden die wichtigsten Hersteller und deren Produkte verglichen. Da die Konsolensteuerung Kinect der Firma Microsoft Inc. nicht für gewerbliche Zwecke gedacht ist, sind auch nur wenige Daten vorhanden.

Hersteller	Produkt	Details	Kosten
MESA Imaging AG ³	SR4000 [4]	<ul style="list-style-type: none"> • Auflösung: 176x144 Pixel • Messbereich: 0,8m bis 8m • Bis zu 50 Bilder pro Sekunde • USB und Fast Ethernet Anschluss 	ab ca. 5000 €
PMD Technologies GmbH ⁴	PMD[vision]® CamCube 3.0 [8]	<ul style="list-style-type: none"> • Auflösung: 204x204 Pixel • Messbereich von 0,3m bis 7m • API and MATLAB Interface • USB Anschluss 	ab ca. 6500€
IFM Inc. ⁵	Efector PMD 3D [9]	<ul style="list-style-type: none"> • Auflösung: 64x50 Pixel • PMD 3D-Chip • Ethernet Anschluss 	ab ca. 800 €
Microsoft Inc. ⁶	Kinect	<ul style="list-style-type: none"> • Motion Controller für die Spielekonsole Xbox 360 • gewerbliche Nutzung nicht erlaubt 	ab ca. 130 €

Tabelle 1.1: Hersteller ToF-Kamerasysteme

Durch die schnelle technische Entwicklung und den ausgezeichneten Eigenschaften, wie z.B. der Echtzeitfähigkeit und dem kleinen Platzbedarf, finden ToF-Kameras immer mehr Verwendung. Man findet sie unter anderem in Spielekonsolen, der Robotik oder der Medizintechnik wieder, wo sie vielseitig eingesetzt werden.

³ www.mesa-imaging.ch

⁴ www.pmdtec.com

⁵ www.ifm.com

⁶ www.mircosoft.com

Kapitel 3

Gestenerkennung

3.1 Definition

Das Wort Geste kommt von dem lateinischen Wort „gestus“, was soviel wie „*Mienenspiel, Gebärdenspiel*“ bedeutet. Für das Wort gibt es viele Definitionen, wobei die meisten eine Geste als nonverbale, kommunikative Bewegung des Körpers beschreiben. [10]

Im Zuge dieser Arbeit bezieht sich die Gestenerkennung auf die menschliche Hand, welche kamerabasiert erkannt werden sollte.

3.2 Optische Gestenerkennung

Neben der taktilen Gestenerkennung, die zur Erkennung der Geste technische Hilfsmittel wie z.B. einen Datenhandschuh oder Infrarotkontroller verwenden, gibt es auch berührungslose bzw. optische Gestenerkennung. Diese Form der Gestenerkennung wird z.B. bei der Spielekonsole Xbox360 der Firma Microsoft Inc. verwendet, wo die Gesten nur mit Hilfe einer ToF-Kamera oder anderen, rein optischen Sensoren erkannt werden.

Bei der optischen Gestenerkennung unter Verwendung einer ToF-Kamera gibt es verschiedenste Techniken, um die Gesten in den Kameradaten zu erkennen. Durch die Tiefeninformation der Bilder kann eine Tiefensegmentierung durchgeführt werden, bei der die menschliche Hand vom Hintergrund getrennt und als Punktwolke dargestellt wird. Mit Hilfe von mathematischen Algorithmen werden die Daten analysiert. In dieser Arbeit wird die Principal Component Analysis verwendet, um die Hauptachsen der Punktwolke und somit Ausrichtung der Hand zu ermitteln. Durch Translation, Rotation und Skalierung kann ein 3D-Handmodell mit Hilfe der Iterativ-Closest-Point-Methode und den zuvor ermittelten Hauptachsen über die 3D-Punktwolke der Hand gelegt werden. Dadurch können der Handmittelpunkt sowie die Position im Raum bestimmt werden.

3.3 Anforderungen

Hardware

Da ToF-Kameras bis zu 50 Bilder pro Sekunde liefern können, muss auch die Hardware hohe Anforderungen erfüllen. Für die Anwendung der Gestenerkennung als Manipulator für einen Roboterarm genügen hingegen ca. 3fps, da die Stellgeschwindigkeit des Roboterarms ohnehin sehr klein ist und auf schnelle und kleine Positionsveränderungen nicht reagiert werden kann. Diese Bildrate kann mit einem handelsüblichen PC mit 4GB RAM und einem Prozessor mit ca. 2,5GHz Taktfrequenz erreicht werden.

Beleuchtung

Um Fehler in der Aufnahme der Bilder zu verhindern, muss eine zu starke Hintergrundbeleuchtung vermieden werden. Da die Roboterarmsteuerung meist in einem Gebäude betrieben wird, sind jedoch keine Störungen durch zu dominante Hintergrundbeleuchtung, wie z.B. Sonnenlicht, zu erwarten.

Software

Die meisten Anbieter von 3D-Kamerasystemen stellen zur Verarbeitung der 3D-Daten eigene Programme zur Verfügung. Da diese meist in Ihrem Umfang eingeschränkt sind, empfiehlt sich zur Analyse und Weiterverarbeitung eine Software wie MatLAB der Firma Mathworks Inc.⁷, welche auch .stl-Daten verarbeiten und darstellen kann. Dabei werden Volumenkörper durch ein Oberflächenmodell ersetzt, um die Datenmengen zu verringern (vergleiche Kapitel 3.5.2). MatLAB-Befehle können auch in Programmiersprachen wie C oder C++ eingebunden werden. Zur Modellerstellung können alle CAD-Programme, wie z.B. CATIA V5 der Firma Dassault Systemes Inc.⁸, verwendet werden, die die .stl-Schnittstellen unterstützen.

⁷ www.mathworks.de

⁸ www.3ds.com

3.4 ToF-Kameradaten

3.4.1 Kalibrierung

Um korrekte 3D-Daten zu bekommen, muss die ToF-Kamera kalibriert werden. Dazu müssen die x-, y- und z-Werte der ToF-Kamera mit einem Skalierungsfaktor auf eine reale Strecke umgerechnet und Verzerrungen korrigiert werden. Die Kalibrierung von ToF-Kameras stellt eine sehr komplexe Aufgabe dar und wird im Rahmen dieser Arbeit nicht näher behandelt. Jedoch soll bemerkt werden, dass neben falscher Kalibrierung auch durch äußere Einflussfaktoren, wie z.B. Reflektivitätsunterschiede oder veränderte Umgebungstemperaturen, Abweichungen von wenigen Millimetern entstehen können. Bei Verwendung als Manipulator sind diese Abweichungen nicht relevant.

3.4.2 Tiefensegmentierung

Für die Gestenerkennung muss die menschliche Hand vom Hintergrund unterscheiden werden. Dazu wird eine Region-of-Interest (ROI) festgelegt. Diese bezieht sich allerdings nicht auf eine Region in einem 2D-Bild sondern auf die Tiefe. Dabei wird ein Tiefenbereich festgelegt, in dem sich die Hand bewegen kann. Abbildung 3.1 zeigt ein ToF-Bild, in dem die Hand vom Hintergrund getrennt wurde und in gelb angezeigt wird. Die Hand kann dadurch vom restlichen Bild gelöst und als eigene Punktwolke dargestellt werden.



Abbildung 3.1: Tiefenbild [11]

Eine Schwierigkeit liegt darin, die ideale Entfernung der Hand zu definieren. Um eine genaue Positionierung zu ermöglichen, sollte die Hand möglichst viele Bildpunkte ausfüllen, d.h. möglichst nahe an der ToF-Kamera sein. Dabei können jedoch überbelichtete Bilder entstehen, da das entstehende Streulicht bei der Reflektion des Lichts in der Nähe größeren Einfluss hat als in der Entfernung. Bei zu großer Entfernung kann die Position der Hand nicht mehr identifiziert werden. [12]

3.4.3 Principal Component Analysis (PCA)

Die PCA wird in der Bildverarbeitung auch als Hauptkomponentenanalyse oder als Karhunen-Loeve-Transformation bezeichnet und wird zur Analyse der Punktwolke, die die ToF-Kamera liefert, eingesetzt. In der PCA werden Richtungen in den Daten gesucht, entlang derer die Daten eine maximale oder minimale Varianz haben. Durch Weglassen der Richtungen mit der kleinsten Varianz kann eine Dimensionsreduktion durchgeführt werden, d.h. es werden orthogonale Richtungen gesucht, die die Punktemenge am besten repräsentiert. Diese Richtungen werden als Hauptkomponenten oder Eigenvektoren bezeichnet. Abbildung 3.2 zeigt die Principal Component Analyse einer 2D-Punktwolke mit den Hauptkomponenten v_1 und v_2 .

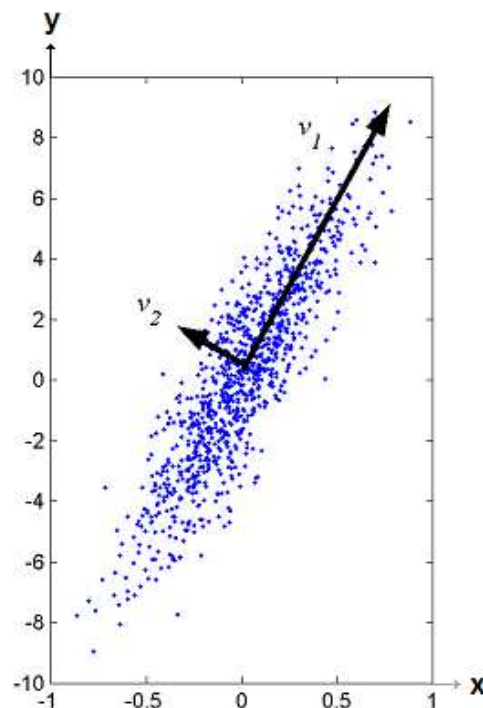


Abbildung 3.2: Principal Component Analyse einer 2D-Punktwolke [14]

Die PCA wird in der Bildverarbeitung unter anderem auch bei der Gesichts- und der Schriftenerkennung eingesetzt. Dabei wird ein zweidimensionales Bild ($N \times M$) mit der Karhunen-Loeve-Transformation in einen Vektor der Größe ($NM \times 1$) umgewandelt. Jedes Bild beschreibt dann einen Punkt im ($N \times M$)-dimensionalen Raum, den sogenannten Bildraum. Mehrere Testbilder vom selben Gesicht treten dann gehäuft in einem Bereich des Raumes auf. Mit der PCA wird nun dieser Bereich ermittelt und zu sogenannten Eigengesichtern zusammengefasst (Abb. 3.3). Dabei werden die Testbilder mit der größten Übereinstimmung, also den größten Eigenwerten der Eigenvektoren, am stärksten gewichtet. Liegt nun ein zu identifizierendes Bild vor, kann die Übereinstimmung mit dem Eigenbild und der PCA überprüft werden. [13]



Abbildung 3.3: Eigengesichter [14]

Eine Variante der Schriftenerkennung funktioniert nach demselben Prinzip. Dabei wird von verschiedenen Bildern eines Buchstabens ein Mittelwertbild erstellt, das wiederum als Vorlage für die Identifizierung dient. Das Bild mit dem größten Eigenwert des Eigenvektors kommt der Vorlage am nächsten.



Abbildung 3.4: Mittelwertbild und Bilder mit den größten Eigenvektoren [14]

Für die Anwendung der Roboterarmsteuerung wird die PCA zur Bestimmung der Zeigerichtung der Punktwolke der Hand verwendet, indem man versucht, die Richtung mit der maximalen Varianz zu finden.

Dabei ist die Punktefolge der Hand $x = x_1, \dots, x_n$ als Element von \mathbb{R}^m von m -dimensionalen Datenpunkten gegeben. Die aus den Punkten berechnete $m \times m$ -Matrix

$$CovMat(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) * (x_i - \bar{x})^T \quad (1.6)$$

bezeichnet man als Kovarianzmatrix von x , wobei der Vektor $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ der Vektor der Mittelwerte der einzelnen Komponenten von x_1, \dots, x_n ist. Da hier im dreidimensionalen Raum gearbeitet wird, besteht jeder Vektor x aus drei Werten, der Pixelposition x und y und dem Tiefenwert z der Kamera. Dadurch sieht die Kovarianzmatrix folgendermaßen aus:

$$CovMat = \begin{pmatrix} Cov_{x,x} & Cov_{x,y} & Cov_{x,z} \\ Cov_{y,x} & Cov_{y,y} & Cov_{y,z} \\ Cov_{z,x} & Cov_{z,y} & Cov_{z,z} \end{pmatrix} \quad (1.7)$$

wobei $Cov_{a,b} = \frac{1}{n} \sum_{i=1}^n (a_i - \bar{a}) * (b_i - \bar{b})^T$ die Kovarianzen sind. [12]

Nach Berechnung der Kovarianzmatrix erhält man durch Berechnung der Eigenvektoren und deren Eigenwerten die Hauptachsen.

Wird die Principal Component Analyse bei Time-of-Flight Kameradaten eingesetzt, zeigt die 1. Hauptachse immer in die Tiefe, da durch die Schattenbildung und einem daraus resultierenden „Kometenschweif“ in diese Richtung immer die größte Varianz der Punktwolke entsteht. Die 2. Hauptachse zeigt immer in Richtung der zweitgrößten Varianz, also in Zeigerichtung des Arms.

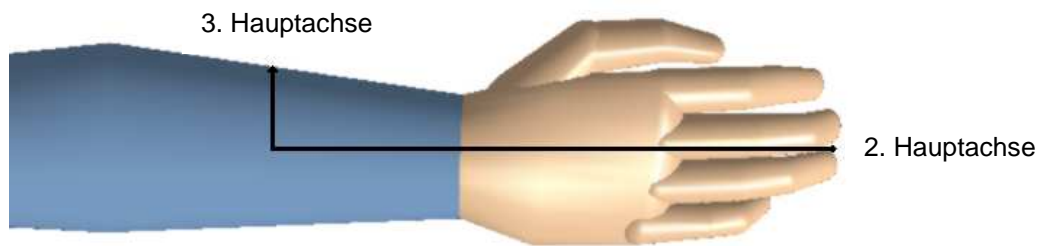


Abbildung 3.5: Hauptachsen der Hand

Liegt eine Drehung der Hand vor zeigt die 3. Hauptachse entweder in Richtung der Breite oder der Höhe des Arms, was für die Positionsbestimmung ein Problem darstellt (vergleiche Abb. 3.5 und Abb. 3.6).

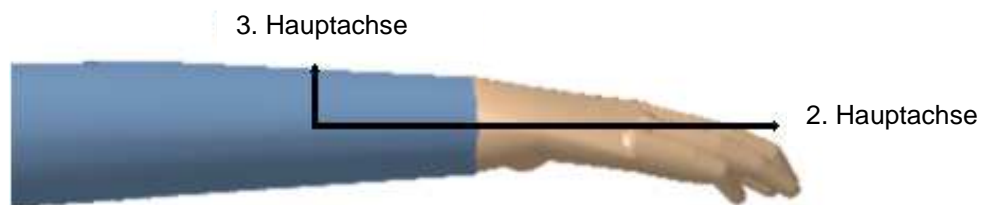


Abbildung 3.6: Hauptachsen der Hand bei Drehung

Zusätzlich ist es problematisch einen fixen Punkt in der Punktwolke zu bestimmen, der den Tool-Center-Point repräsentiert. Um dieses Problem der Drehung zu lösen und nicht den Mittelpunkt der Punktwolke, sondern einen definierten Punkt auf der Hand zu erhalten, muss ein Modell über die Punktwolke gelegt werden.

3.5 Modellierung

Zur Positionsermittlung der Hand muss ein Modell einer menschlichen Hand mit der Punktwolke überlagert werden. Dazu muss ein Modell erstellt werden, wobei es mehrere Möglichkeiten zur Modellerstellung gibt.

3.5.1 Betrachterorientierte Modellierung

Bei der betrachterorientierten Modellierung werden 2D-Referenzbilder aus verschiedenen charakteristischen Ansichten erstellt. Durch diese Bilder ist es möglich, Objekte in den Kameradaten zu erkennen. Nach [15] bietet die betrachterorientierte Modellierung den Vorteil, einfache und gut untersuchte Verfahren aus der 2D-Objekterkennung verwenden zu können. Hat man jedoch mehrere Modelle, ist das zu verarbeitende Datenvolumen enorm.

3.5.2 Objektorientierte/geometrische Modellierung

Bei der objektorientierten Modellierung wird eine vollständige 3D-Rekonstruktion des Modells vorgenommen. Eine Möglichkeit stellt dabei die Verwendung von CAD-Programmen dar. In CAD-Programmen können Modelle gezeichnet und das erstellte 3D-Modul als .stl-Datei abspeichern werden. Eine .stl-Datei reduziert die zu verarbeitende Datenmenge erheblich. Dabei wird ein Oberflächenmodell erstellt, bei dem die exakten Oberflächen durch Dreiecke angenähert werden (Abb. 3.7). In der .stl-Datei werden dabei die drei Eckpunkte und die Flächennormale der Dreiecke abgespeichert. Dieses Format kann zur weiteren Verarbeitung als Punktwolke geladen werden, in der alle Eckpunkte dargestellt werden.

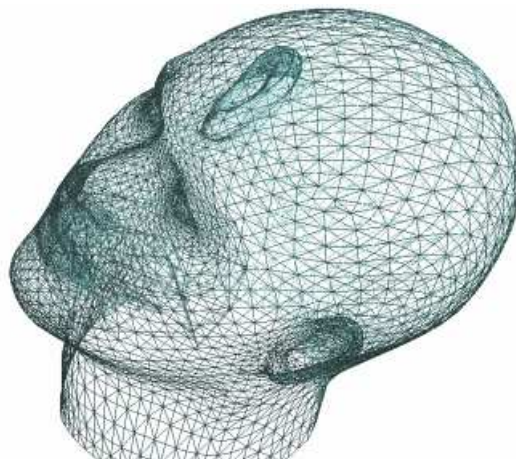


Abbildung 3.7 Oberflächenmodell eines Kopfes als .stl-Datei [16]

Da für die Verwendung als Steuerung für den Roboterarm nur die relative Armbewegung gemessen werden muss, muss das Modell nicht exakt mit dem menschlichen Arm übereinstimmen. Deshalb empfiehlt sich eine geometrische Modellierung mit Hilfe eines CAD-Programmes. Abbildung 3.8 zeigt ein vorgefertigtes Handmodell in High-End-CAD-Programm CATIA V5 von Dessault Systems Inc⁹.

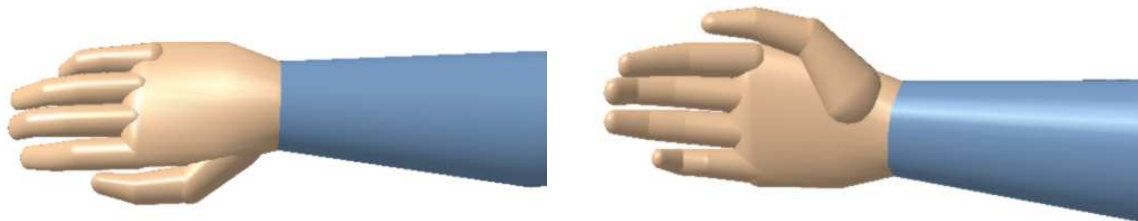


Abbildung 3.8: Handmodell

3.6 Klassifikation

Einer der wesentlichsten Punkte der Gestenerkennung ist die Klassifikation, d.h. das Matching anhand des Modells. Dabei werden Modell und Sensordaten verglichen und mit Hilfe eines Algorithmus überlagert. Eine Methode, die sich in diesem Fall anbietet, ist die Iterativ-Closest-Point Methode. Hier wird versucht zwei Punktwolken so aufeinander zu legen, dass die geringste Abweichung zwischen allen Punkten entsteht.

3.6.1 Iterativ-Closest-Point Methode

Eine Punktmenge stammt hierbei aus dem Datensatz des Referenzbildes, die andere aus dem Datensatz des zu transformierenden Bildes. Man sucht eine Transformationsmatrix, die, angewandt auf die Punktmenge D, aus dem Datensatz

⁹ www.3ds.com

des zu transformierenden Bildes die Abstandskquadrate zwischen beiden Punktmengen minimiert:

$$\min_{R,t} \|M_i - (RD_i + t)\|^2 \quad (1.8)$$

D: Punktmenge des zu transformierenden Bildes

M: Punktmenge des Referenzbildes

R: Rotationsmatrix

t: Translationsvektor

Index i: Punkte aus den Punktmengen

Man bestimmt dazu zunächst zu jedem Punkt aus dem Referenzbild den nächsten Punkt im zu transformierenden Bild. Danach berechnet man die oben genannte Transformationsmatrix und wiederholt den Vorgang so lange, bis eine Terminierungsbedingung erreicht ist. [17]

Da die Laufzeit dieser Methode sehr stark vom vorgegebenen Startwert abhängig ist, muss das Modell mit einer Transformationismatrix zuerst an der in Punkt 3.4 errechneten 2. Hauptachse ausgerichtet werden. Dadurch reduzieren sich die iterativen Schritte.

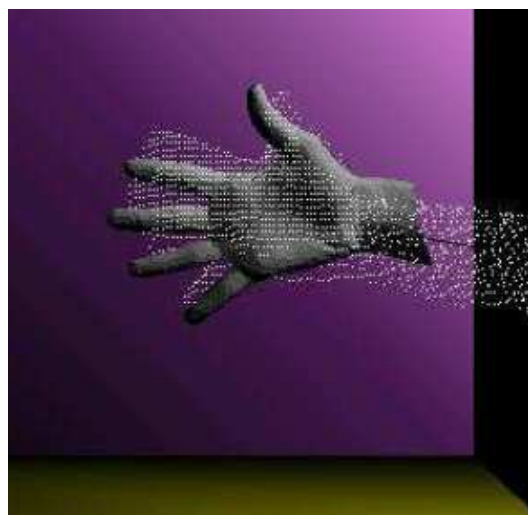


Abbildung 3.9: Überlagerung von Modell- und Sensordaten [12]

3.7 Auswertung

Durch den berechneten Translationsvektor t und der Rotationsmatrix R aus Formel 1.8 erhält man die Position des Modells und somit auch die sechs Freiheitsgrade des Tool-Center-Points im Raum. Damit kann durch inverse Kinematik die Gelenkstellung des Roboterarms berechnet werden (vergleiche [19]).

Bei einer geforderten Framerate von ca. 3fps bedeutet das, dass für die Klassifizierung eine Rechenzeit von ca. 0,33s zur Verfügung steht. Bei Problemen mit der Laufzeit könnte diese verringert werden, indem man die Terminierungsbedingung des Iterativ-Closest-Point-Verfahrens erhöht und somit weniger Durchläufe benötigen würden. Dabei muss jedoch beachtet werden, dass dadurch die Reproduzierbarkeit des Ergebnisses ebenfalls gesenkt wird und es zu leichten Sprüngen des Modells kommen könnte.

Man sieht jedoch, dass der Einsatz von Gestenerkennung unter Verwendung einer Time-of-Flight Kamera eine gute Möglichkeit darstellt, eine intuitiv bedienbare Roboterarmsteuerung zu realisieren.

Kapitel 4

Ausblick

Im Zuge dieser Arbeit wurde die Positionserfassung der menschlichen Hand mit den sechs Freiheitsgraden des Raums thematisiert. Falls gefordert, könnte man diese Positionserfassung um eine Handzeichenerkennung erweitern, die z.B. das Öffnen und Schließen des Greifers steuert (Abb. 4.1).

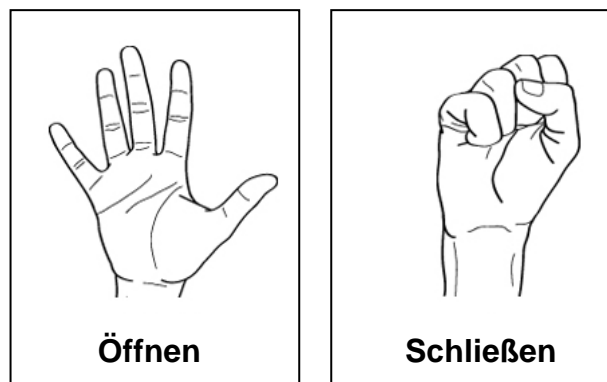


Abbildung 4.1: Beispiel für Handzeichen zum Steuern des Greifers [18]

Um zu wissen ob Handzeichen vorliegen, könnte man die gelieferten Grauwertbilder der ToF-Kamera nutzen und eine klassische, zweidimensionale Bildverarbeitung einsetzen, die erkennt, ob Handzeichen gegeben werden. Dabei muss jedoch bedacht werden, dass bei mehreren Handzeichen auch mehrere Modelle erstellt werden müssen, um die Zuordnung mit dem Iterativ-Closest-Point Verfahren (siehe Kapitel 3.7) zu gewährleisten.

Zusätzlich könnte die Gestenerkennung unter Verwendung einer ToF-Kamera als Basis für viele weitere Anwendungen, wie z.B. einer intuitiven Eingabeform für Programme, verwendet werden.

Kapitel 5

Fazit und Dank

Im Rahmen dieser Arbeit wurde das Thema der Gestenerkennung mit Hilfe einer Time-of-Flight Kamera behandelt. Die größten technischen Herausforderungen der Gestenerkennung sind dabei die mathematischen Analysen der aufgenommenen Bildpunkte, welche in dieser Arbeit mit der Principal-Component-Analyse zur Bestimmung der Zeigerichtung des Arms und dem Iterativ-Closest-Point-Verfahren zur Klassifikation durchgeführt wurden. Im weiteren Verlauf wird die Robotersteuerung mit Hilfe der kamerabasierten Gestenerkennung umgesetzt und in Betrieb genommen.

Abschließend möchte ich mich bei Herrn Prof. (FH) Dipl. Ing. Kurt Niel für die Betreuung der Bachelorarbeit, und bei Herrn DI(FH) Raimund Edlinger und Herrn Ing. Michael Zauner, BSc. für die technische Unterstützung bedanken.

Ich denke, dass ich durch diese Bachelorarbeit viel Wissen und viele Erfahrungen gesammelt habe, die mir in meinem künftigen Berufsleben sicher von Nutzen sein werden.

Kapitel 6

Verzeichnisse

6.1 Abbildungsverzeichnis

ABBILDUNG 1.1: ROBOTERARM DER FH-WELS	2
ABBILDUNG 1.2: MANIPULATION DES ROBOTERARMS DURCH ARMBEWEGUNGEN	3
ABBILDUNG 2.1: TOF KAMERA [4].....	6
ABBILDUNG 2.2: AMPLITUDENMODULATION.....	7
ABBILDUNG 2.3: 3D-PUNKTWOLKE [7].....	8
ABBILDUNG 2.4: 2D-GRAUWERTBILD [6]	8
ABBILDUNG 3.1: TIEFENBILD [11].....	12
ABBILDUNG 3.2: PRINCIPAL COMPONENT ANALYSE EINER 2D-PUNKTEWOLKE [14].....	13
ABBILDUNG 3.3: EIGENGESICHTER [14].....	14
ABBILDUNG 3.4: MITTELWERTBILD UND BILDER MIT DEN GRÖßTEN EIGENVEKTOREN [14]..	14
ABBILDUNG 3.5: HAUPTACHSEN DER HAND	16
ABBILDUNG 3.6: HAUPTACHSEN DER HAND BEI DREHUNG	16
ABBILDUNG 3.7 OBERFLÄCHENMODELL EINES KOPFES ALS .STL-DATEI [16].....	17
ABBILDUNG 3.8: HANDMODELL.....	18
ABBILDUNG 3.9: ÜBERLAGERUNG VON MODELL- UND SENSORDATEN [12]	19
ABBILDUNG 4.1: BEISPIEL FÜR HANDZEICHEN ZUM STEUERN DES GREIFERS [18].....	21

6.2 Tabellenverzeichnis

TABELLE 1.1: HERSTELLER TOF-KAMERASYSTEME	9
---	---

6.3 Literaturverzeichnis

- [1] R. Gelhart: *Hochleistungsbildverarbeitung zur Ansteuerung von Antriebsachsen*, Diplomarbeit an der FH-Wels, 2006
- [2] C. Bauer: *Optisch gesteuerte Werkzeugmaschinen*, Diplomarbeit an der TU Wien, 2007
- [3] Jiang X., Bunke H.: *Dreidimensionales Computersehen*, Springer Verlag, Heidelberg 1997, ISBN 3-540-60797-8, S. 60 ff
- [4] MESA GmbH, <http://www.mesa-imaging.ch/MesaTof.php>, abgerufen am 07.05.2011
- [5] TOF-Kamera, <http://de.wikipedia.org/wiki/TOF-Kamera>, abgerufen am 07.03.2011
- [6] C. Heckenkamp: *Das magische Auge - Grundlagen der Bildverarbeitung: Das PMD Prinzip*. In: *Inspect*. Nr. 1, 2008, S. 25
- [7] 3D-Punktewolke, <http://www.ruhr-uni-bochum.de/geodaesie/3dkamera.html>, abgerufen am 26.06.2011
- [8] PDM CamCube 3.0, http://www.pmdtec.com/fileadmin/pmdtec/downloads/documentation/datenblatt_camcube3.pdf, abgerufen am 07.05.2011
- [9] IFM GmbH, <http://ifm-electronic.com.tr/ifmat/web/dsfs!O3D201.html>, abgerufen am 08.05.2011
- [10] P. Harling, A. Edwards: *Progress in Gestural Interaction*, Springer Verlag, 1997, ISBN 3-540-76094-6, S. 75ff

- [11] Tiefensegmentierung, <http://www.artts.eu/events/ict2008>, abgerufen am 02.06.2011
- [12] P. Bräuer: *Entwicklung einer prototypischen Gestenerkennung*, Universität Koblenz, 2005
- [13] PCA, <http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-5059/anhang.pdf>, abgerufen am 10.06.2011
- [14] Bilder PCA, http://www-home.fh-konstanz.de/~mfranz/muk_files/lect11_unsup.pdf, abgerufen am 10.06.2011
- [15] G. Hetzel: *Ansichtsbasierte Erkennung komplexer 3D-Objekte aus Tiefenbildern*, IPVS, Wien 2005, ISBN 3-8325-0733-7, S. 8ff
- [16] .stl-Datei, http://www.tenlinks.com/news/PR/field_precision/070609_geometer_stl.htm, abgerufen am 02.06.2011
- [17] Iterativ-Closest-Point, <http://ias.in.tum.de/seminare/hs.WS02.intrnav/ausarbeitungen/Registrierung.pdf>, abgerufen am 02.06.2011
- [18] Handzeichen, <http://www.fingeralphabet.org/downloads>, abgerufen am 26.06.2011
- [19] T. Penkner: *Inverse Kinematik eines Roboterarms*, Bachelorarbeit an der FH-Wels, 2011