

Ein wissensbasiertes Rahmensystem zur merkmalsbasierten Gestenerkennung für multimediale Anwendungen

von

Martin Fröhlich

1999

Ein wissensbasiertes Rahmensystem zur merkmalsbasierten Gestenerkennung für multimediale Anwendungen

Eine zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Universität Bielefeld
von der Technischen Fakultät
genehmigte

Dissertation

von

Dipl.-Inform. Martin Fröhlich

Gutachter:
Prof. Dr. Ipke Wachsmuth
Prof. Dr. Helge Ritter
Prof. Dr.-Ing. Gerhard Sagerer

Sommer 1999

Tag der Promotion: 3. September 1999

Dieses Werk wurde mittels Adobe FrameMaker 5.5.6 auf einer Sun ULTRASparc 10 gesetzt und die Bilder mit dem GNU Image Manipulation Program (gimp) überarbeitet. Und auf einem druckbaren Bereich von 15,3 cm × 24,2 cm ausgelegt, der sich sowohl auf DIN A4 wie auch auf US Letter Papier drucken läßt und bei Verkleinerung auf 80% für ein Buch-Format geeignet ist.

In seiner elektronisch gespeicherten Form ist es eine Datei im Portable Document Format und enthält Hypertextverweise auf interne Ziele wie Sprünge von Zitatsmarken auf die Literaturangaben und auch Verweise auf externe Ziele, die in Form von Uniform Recource Locations (URLs, Web-Links) angegeben sind. Dabei ist sich der Autor bewußt, daß diese Informationen nicht persistent sind und bestenfalls den Stand des Frühjahres 1999 wiedergeben. Leider gibt es noch keine *etablierte* Methode auf Daten im Internet über persistente Indizes zuzugreifen.

Die Referenzexemplare sind auf alterungsbeständigem Papier nach nach ∞ DIN-ISO 9706 gedruckt.

Alle genannten Firmen- und Produktnamen sind eingetragene oder nicht eingetragene Warenzeichen der betreffenden Unternehmen.

Die Bibliothek der Universität Bielefeld, Die Deutsche Bibliothek und die DFG-Sondersammelgebietsbibliothek haben das Recht diese Dissertationsschrift in elektronischer Form in Wissenschaftsnetzen zugänglich zu machen sowie zusätzliche gedruckte Exemplare herzustellen oder zu verbreiten

Danksagung

Eine Dissertationsschrift, als eine Prüfungsleistung in einem Promotionsverfahren, kann nur mit guter Betreuung in einem motivierten Team ein Erfolg werden. Daher möchte ich mich an dieser Stelle bei all denen bedanken, die mir mit Rat und Tat zur Seite standen. Besonderer Dank gilt meinem Betreuer Prof. Dr. Ipke Wachsmuth, der dies Thema auslobte und der mich schon bei meiner Diplomarbeit und in der Zeit der Erstellung dieser Arbeit, stets unterstützte. Auch meine Gutachter Prof. Dr. Helge Ritter und Prof. Dr.-Ing. Gerhard Sagerer sollen an dieser Stelle hervorgehoben sein; ich bedanke mich für die Mühe, die Gutachten zu schreiben. Ebenso möchte ich mich bei den anderen Antragstellern des Graduiertenkollegs „Aufgabenorientierte Kommunikation“ bedanken, die nicht nur, indem sie mir ein Stipendium zuerkannten, diese Arbeit ermöglichten, sondern auch immer mit tätigem Rat zur Seite standen.

Weiterhin bleibe ich meinen Kollegen im Graduiertenkolleg wie in der Arbeitsgruppe dankbar verbunden: Unserem ehemaligen Diplomanden, jetzigen Kollegen und treuen Mitstreiter Timo Sowa, meinem stets hilfsbereiten Zimmergenossen Martin Hoffhenke, meinem Projektmitstreiter in SGIM und ehemaligen Diplomanden Marc Erich Latoschik, dem um keine Antwort verlegenen Bernhard Jung, dem stets aufgeschlossenen Stefan Kopp und der mittlerweile nicht mehr in der Arbeitsgruppe beschäftigten Britta Lenzmann sowie den Kollegiaten der ersten Stunde, insbesondere Kerstin Fischer, Katrin Kirchhoff, Brigitte Schaffranietz, Martina Hasenjäger, Peter Meinicke, Hans Brandt-Pook und Katrin Lehnen. Sie beantworteten mir eine Unzahl von Fragen und halfen durch gemeinsame Diskussion von Problemstellungen die Übersicht zu behalten. Ein weiterer Dank gebührt Karl-Heinrich Schmidt, Medienzentrum Essen, für die vermittelten Einsichten bei der Systematik der Wissensrepräsentation, Gunnar Brinkmann, Fakultät für Mathematik, für etwas Nachhilfe in Graphentheorie und den Teilnehmern der drei internationalen Gestenworkshops, die teilweise auch über die Workshops hinaus Diskussions- und Lösungsanreize gaben. Nicht zuletzt unterstützte mich Peter Serocka, VisLab des FSP Strukturbildungsprozesse, mit gutem Rat bei vielen technischen „Kleinigkeiten“. Der Rechnerbetriebsgruppe, personifiziert durch Rainer Orth und Peter Koch, verdankt die Fakultät stabile Arbeitswerkzeuge, die mir in der ganzen Zeit immer zuverlässig zur Verfügung standen.

Martin Fröhlich
Im Frühjahr 1999

Meiner mich beständig unterstützenden Familie

Inhaltsverzeichnis

	<i>Danksagung</i>	v
	<i>Inhaltsverzeichnis</i>	ix
	<i>Abbildungsverzeichnis</i>	xiii
	<i>Tabellenverzeichnis</i>	xvii
Kapitel 1	Einleitung	1
1.1	Hintergrund	2
1.2	Motivation	6
1.3	Ziele	7
1.4	Überblick	8
Kapitel 2	Theoretische Grundlagen	11
2.1	Gesten aus Sicht des Produzenten	12
2.2	Gesten aus Sicht des Rezipienten	13
2.3	Innere Struktur von Gesten	21
2.4	Verhalten zum Sprachkanal	23
2.5	Repräsentation	24
2.6	Zusammenfassung	30
Kapitel 3	Zeitbegriffe	31
3.1	Gleichzeitigkeit – Ungleichzeitigkeit	31
3.2	Ordnungsschwelle	32
3.3	Drei-Sekunden-Phänomen	32
3.4	Zeitbänder	35
3.5	Zusammenfassung	36
Kapitel 4	Gestennotationssysteme	37
4.1	Labanotation (Kinetographie Laban)	39
4.2	Berner System zur Untersuchung nonverbaler Interaktion	43
4.3	Hamburger Notationssystem	47

4.4	Zusammenfassung	55
Kapitel 5	Gestenbehandelnde Systeme	57
5.1	Klassische Systeme	57
5.2	ICONIC	58
5.3	Gebärdenerkennung mit Sensorhandschuhen	60
5.4	Viena	62
5.5	SignRec	65
5.6	SGIM	66
5.7	Zusammenfassung	68
Kapitel 6	Konzeption eines Gestenerkenners	71
6.1	Systemgrundlagen	71
6.2	Schicht 1: Datenaufnahme	78
6.3	Schicht 2a: Vorverarbeitung	82
6.4	Schicht 2b: Körpermodell	82
6.5	Schicht 2c: Merkmalerkennung	91
6.6	Schicht 3: Integration	93
6.7	Architektur eines Gestenerkennungssystems	97
6.8	Zusammenfassung	99
Kapitel 7	Vorschlag einer Realisation des Gestenerkenners	101
7.1	Systemgrundlagen	101
7.2	Datenaufnahme	105
7.3	Vorverarbeitung	109
7.4	Körpermodell	109
7.5	Merkmalerkennung	119
7.6	Integrationsverfahren	126
7.7	Implementierungsaspekte	138
7.8	Zusammenfassung	142
Kapitel 8	Anwendung	143
8.1	Einfache Gestenerkennung	143
8.2	Einfache multimodale Integration	147

8.3 Zusammenfassung	149
Kapitel 9 Résumé	151
9.1 Wesentliche Systemeigenschaften	151
9.2 Lösungen	152
9.3 Kritik	153
9.4 Abschließende Bemerkungen	153
<i>Anhänge</i>	155
1.1 Multimodal versus Multimedial	155
1.2 Prinzipien der Softwareentwicklung	155
1.3 Agentenorientierte Programmierung	156
1.4 HamNoSys und SGIM	158
1.5 Tabelle der Faktoren im Körpermodell	164
1.6 Die Klassen in COOL	166
1.7 Thread-sichere Programmierung	167
1.8 Technik der Datenaufnahme	171
1.9 Störungen bei der Zuweisung von Zeitstempeln	175
1.10 Näherung der Handspannungsfunktion	177
<i>Literaturverzeichnis</i>	183
<i>Index</i>	203

Abbildungsverzeichnis

Abbildung 1: Kommunikationsmodell	2
Abbildung 2: Vorschlag für eine Klassifizierungshierarchie.	20
Abbildung 3: Regeln für Bewegungsphasen nach [McN92].	21
Abbildung 4: Syntagmatische Organisation von Gesten nach [KGH98].	22
Abbildung 5: Glyphen und Notationssysteme	28
Abbildung 6: Verkehrszeichen.	29
Abbildung 7: Der Necker-Würfel	33
Abbildung 8: Reproduktion von Zeitstrecken nach [Poe97].	34
Abbildung 9: SignStream: Eingabe- & Videofenster.	38
Abbildung 10: SyncWriter.	39
Abbildung 11: Labanotation: Zeichenerklärungen	40
Abbildung 12: Labanotation: Einführungsbeispiel	41
Abbildung 13: Labanotation: bekannte Tanzschritte	42
Abbildung 14: Komplexe Bewegung aus [Hut77], Seite 397	44
Abbildung 15: HamNoSys: Grundformen	49
Abbildung 16: HamNoSys: Modifikationen der Grundformen	50
Abbildung 17: HamNoSys: Verbindungsformen	50
Abbildung 18: HamNoSys: Fingeransatzrichtungen	50
Abbildung 19: HamNoSys: Handflächenorientierung.	51
Abbildung 20: HamNoSys: Aufenthaltsort der Hand in der Vertikalen.	52
Abbildung 21: HamNoSys: Einteilung der transversalen Ebene	53
Abbildung 22: HamNoSys: Beidhändige symmetrische Notation	54
Abbildung 23: Das „Iconic“ System nach [Wex94].	59
Abbildung 24: Die Agentur des Viena Systems nach [CJW95].	63
Abbildung 25: Die erweiterte Agentur des Viena Systems (nach [WLJ+97])	64
Abbildung 26: SGIM: schematische Sicht und Architektur, nach [LFJ+98]	67
Abbildung 27: Schichtenmodell.	72
Abbildung 28: Schichtenmodellvergleich, eigenes und [Rad96].	73

Abbildung 29: vertikale und horizontale Ordnung.	75
Abbildung 30: Das erweiterte Schichtenmodell	76
Abbildung 31: Das menschliche Skelett: Schultergürtel aus [RY94]	82
Abbildung 32: Bewegung des Schulterblattes nach [RY94]	83
Abbildung 33: Thoraxial- und Armskelett bei Elevation nach [RY94]	83
Abbildung 34: Supinationsstellung der Hand.	84
Abbildung 35: Armmodell aus [LU94]	86
Abbildung 36: Armmodell nach [Ko94]	87
Abbildung 37: Geometrisches Körpermodell nach [FW98]	88
Abbildung 38: Integrationshierarchie nach [FW98]	94
Abbildung 39: Vorschlag einer Systemarchitektur	98
Abbildung 40: Hierarchie der Symbolklassen.	99
Abbildung 41: Skizze der Integrationshierarchie	100
Abbildung 42: Das WBS Multiagentensystem	103
Abbildung 43: Flock of Birds und einzelner Sensor	106
Abbildung 44: Labor für Künstliche Intelligenz und Virtuelle Realität in Bielefeld .	107
Abbildung 45: Das „Mean of Multiple Computations“-Netz.	111
Abbildung 46: Armmodell für MMC-Netz.	112
Abbildung 47: Grundgleichungen des MMC-Netzes.	113
Abbildung 48: Benötigte Gleichungen für eine Variable	115
Abbildung 49: Anwendung des MMC auf das Körpermodell.	117
Abbildung 50: Einteilung der transversalen Ebene	122
Abbildung 51: Bewegung: Einteilung der Hauptebenen	123
Abbildung 52: Handflächenorientierung: Einteilung der Ebene	126
Abbildung 53: Hand mit Sensoren.	127
Abbildung 54: Abklingfunktionen	133
Abbildung 55: Fingermodell nach [HE96]	134
Abbildung 56: Prototyp A: Systemkonfiguration	143
Abbildung 57: Symboldefinition zur Laufzeit.	145
Abbildung 58: Integrator Visualisierung	146
Abbildung 59: Prototyp B: Systemkonfiguration	148

Abbildung 60: Wortklassen und ihre Instanzen	149
Abbildung 61: Prozeßflußdiagramm mit Threads nach [KSS96]	169
Abbildung 62: Struktur und Klassenhierarchie der „Basic Objects“	170
Abbildung 63: Klassenhierarchie eines „Locked Basic Objects“	171
Abbildung 64: Labor für Künstliche Intelligenz und Virtuelle Realität in Bielefeld .	172
Abbildung 65: Graph der Ausgleichsfunktion nach [FW98]	174
Abbildung 66: Meßdaten von ruhenden Birds, 3D-Sicht..	174
Abbildung 67: Meßdaten von ruhenden Birds, 2D-Projektion.	176
Abbildung 68: Fehler des Zeitstempels	177
Abbildung 69: Güte der Näherungsfunktionen für den Zeigefinger	178
Abbildung 70: Güte der Näherungsfunktionen der universellen Fingerspannung.	180

Tabellenverzeichnis

Tabelle 1: Zusammenfassung von Kategorisierungssystemen nach [Wex95]	15
Tabelle 2: AHIG System	17
Tabelle 3: WBS System nach [LW98]	17
Tabelle 4: <i>Versuch</i> der Klassifizierung einiger Gestenarten	20
Tabelle 5: Apexabstände und Regressionsgewichte nach [Rui98], Seite 56.	23
Tabelle 6: Zeitbänder der menschlichen Kognition nach [New90].	35
Tabelle 7: Zusammenfassung des Berner Notationssystems	45
Tabelle 8: Vergleich der Notationen	55
Tabelle 9: SignRec Ergebnisse, vereinfacht nach [HBK99].	65
Tabelle 10: Gestenbehandelnde Systeme	68
Tabelle 11: Einordnung des Schichtenmodells in die Zeitmodelle	74
Tabelle 12: Vergleich der Sensortypen (Körper)	81
Tabelle 13: Bewegungsumfänge des Schulterbereichs aus [Ben94].	85
Tabelle 14: Bezeichner im Körpermodell.	118
Tabelle 15: Höheneinteilung im Gestikulationsraum	120
Tabelle 16: Abstandseinteilung im Gestikulationsraum	121
Tabelle 17: Positionen im Gestikulationsraum.	121
Tabelle 18: Bewegungsrichtungen im Gestikulationsraum	124
Tabelle 19: Geschwindigkeit.	125
Tabelle 20: Fingerlängen	135
Tabelle 21: Fehler der approximierten Polynome (pro Finger).	136
Tabelle 22: Fehler der approximierten Polynome (Fingerlänge 1)	137
Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente	159
Tabelle 24: Beispielwerte für konsistente Belegung des Körpermodells	165
Tabelle 25: Koeffizienten der Ausgleichsfunktion	175
Tabelle 26: Koeffizienten aller Polynome für die Zeigefingerspannung	179
Tabelle 27: Koeffizienten aller Polynome für die universelle Fingerspannung	179

Kapitel 1

Einleitung

Als am Anfang der Computerentwicklung Maschinen wie ENIAC¹⁾, Colossus²⁾ und EDVAC³⁾ gebaut wurden, war die „Kommunikation“ mit diesen riesigen Apparaten auf die teilweise Neuverdrahtung der Maschinen und das Ablesen von Indikatorlampen beschränkt. Mit dem Stanzen und Lesen von Lochkarten war zwar danach die prinzipielle Möglichkeit einer dem Menschen verständlichen Kommunikationsform in Form einer Schrift gegeben, doch hatten diese etwa geldscheingroßen Kartonstreifen so gut wie keine Aussagekraft für jeden, der nicht zuvor eine anspruchsvolle Ausbildung durchlief. Als Fernschreiber zur Bedienung eingeführt wurden, war die Schnittstelle zwischen Mensch und Maschine auf eine für den Menschen bekannte Form, die Telegraphie, zurückgeführt. Doch erst dann, als auf Bildschirmen zuerst alphanumerische Glyphen und dann auf Graphikterminals Bilder möglich waren, wurde der Bedarf an angenehmer Darstellung und effizienter Eingabe der Daten offensichtlich. Schon Anfang der sechziger Jahre, mit der Etablierung der Forschungsrichtung „Künstliche Intelligenz“, wurde Kommunikation, als ein mit dem Intelligenzbegriff eng verwobenes Thema, wissenschaftlich untersucht. Als 1956 im Dartmouth College Hanover, New Hampshire, während einer Tagung der Begriff „Artificial Intelligence“ von John McCarthy, Marvin Minsky, Allen Newell und anderen Wissenschaftlern geprägt wurde, ging man von Allan M. Turings logischem Beweis aus, daß eine „Universelle Rechenmaschine“ (Turing-Maschine) möglich ist. Man

-
1. In dem Projekt ENIAC (*Electronic Numerical Integrator and Computer*) bauten J. Presper Eckert und John W. Mauchly an der Moore School of Engineering einen Rechner in Röhrentechnologie, um ballistische Tafeln für die Artillerie zu berechnen. Der ENIAC-Rechner wurde 1946 fertiggestellt. [Tan94]
 2. Colossus war ein Projekt in Bletchley Park während des Zweiten Weltkrieges. Es ging darum, den Code der deutschen Enigma-Codiermaschinen, der insbesondere von der U-Bootwaffe eingesetzt wurde, zu entschlüsseln. Der englische Mathematiker und Logiker Allan M. Turing hat während seiner Zeit am Colossus-Projekt grundlegende Vorarbeiten für die Anfänge des Rechnerbaus geleistet. [Tan94]
 3. Der EDVAC-Rechner, der von dem ungarischen Mathematiker Johann von Neumann am *Institute for Advanced Studies* in Princeton in den Jahren von 1944 bis 1946 entwickelt wurde, gilt als der erste Computer in heutigem Sinne. Das Akronym EDVAC steht für *Electronic Discrete Variable Automatic Computer*. [Tan94]

suchte Wege, mit Computern, die eigentlich ausschließlich zur syntaktischen Manipulation fähig sind, semantische Informationsverarbeitung zu ermöglichen.

Heutzutage sind Computer nicht nur Geräte des Alltags geworden, sie werden auch zu immer komplexeren Aufgaben herangezogen. Um so wichtiger wird dadurch die Forschung nach neuen, „intelligenten“ Kommunikationsformen. Idealerweise sollte eine Maschine das gleiche Kommunikationsphänomen liefern wie ein Mensch. Obwohl der Stand der Forschung davon noch weit entfernt ist, scheint es trotzdem sinnvoll, den Menschen als Vorbild zu betrachten.

Im Alltag ist zu beobachten, daß zwischenmenschliche Kommunikation sich verschiedener Kanäle bedient. Neben der gesprochenen Sprache werden Gestik und Mimik des Gegenübers ausgewertet. Auch für die Mensch–Maschine–Kommunikation kristallisieren sich multimodale Kommunikationsformen, besonders die Kanäle Sprache und Gestik heraus. Im folgenden soll ein Beitrag zur Erschließung einer natürlichen Kommunikationsform — der Gestik — für die Interaktion des Menschen mit komplexen rechnerbasierten Systemen geleistet werden.

1.1 Hintergrund

KOMMUNIKATION. Für die Kommunikation von Mensch zu Mensch stehen uns verschiedene Kanäle zur Verfügung. Solche Kanäle sind zum Beispiel die verschiedenen Ebenen der Sprache, der Mimik und der Gestik sowie die Gebärden. Auf diesen Kanälen sind wir in der Lage, verschiedene, teilweise unabhängige und sogar widersprüchliche Informationen auszutauschen, die erst zusammen die eigentliche Botschaft vermitteln. Kommunikationstheoretisch betrachtet, wird also eine Nachricht in mehrere, oft ungleich große Teile aufgespalten, die dann jeweils über verschiedene, unterschiedlich verrauschte Kanäle übertragen, und dann so vom Empfänger wieder zusammengesetzt wird. Fehlerquellen ergeben sich hierbei dadurch, daß die Fähigkeit des Senders, die ursprüngliche Nachricht so in Teile aufzuspalten, daß die Summe dieser Teile wieder die Nachricht ergibt, von seinem Ausdrucksvermögen abhängt. Andererseits müssen die Kanäle so gewählt werden, daß der Empfänger sie auch bei der Datenfusion berücksichtigen kann (siehe Abbildung 1).

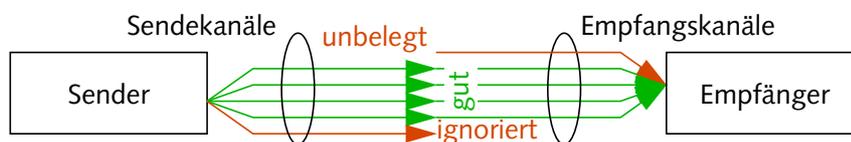


Abbildung 1: Kommunikationsmodell

Definition: Geste: Vorerst soll unter dem Begriff „Geste“ eine Bewegung verstanden werden, die für einen Betrachter interpretierbar ist, ungeachtet der Intention, die der sie äußernde hat. Somit subsumiert der hier eingeführte Begriff die klassischen Begriffe „Geste“ und „Gebärde“ und lehnt sich in seiner Bedeutung dem englischen Begriff „gesture“ an. Im Abschnitt „Theoretische Grundlagen“ auf Seite 11 werden weitere und verfeinerte Definitionen vorgestellt werden.

Definition: Gestenart: Eine Gestenart sei eine Menge von Gesten, die in möglichst vielen *Merkmale* übereinstimmen.

Definition: Gestentyp: Ein Gestentyp ist eine Menge (Cluster) von Gesten, die durch einen typischen *Vertreter* beschrieben ist.

Merkmale im Sinne obiger Definition können sich aus der funktionalen Rolle der Geste in der Kommunikation oder Interaktion ergeben (z.B. Fühlen), aus Kategorisierungssystemen hergeleitet werden oder direkt als physikalische oder biomechanische Parameter der Bewegung entnommen werden (z.B. ist schnell; beugt das Handgelenk).

MENSCH–MASCHINE–KOMMUNIKATION. Die Mensch–Maschine–Kommunikation (MMK, engl.: Human Computer Interaction; HCI) als Teilgebiet der Informatik untersucht, wie die beiden verschiedenen Weltbilder, die sich aus der Sicht von Mensch und Maschine ergeben, einen Berührungspunkt finden können, der eine möglichst einfache Kommunikation ermöglicht. Dabei wird versucht, eine „natürliche“ Kommunikationsbasis zu schaffen, die einem Laien wie einem Experten ermöglichen soll, effizient mit der Maschine umzugehen. Unter diesen Bedingungen ist es eine besondere Aufgabe, die Kommunikation nicht nur effizient für die Menschen, sondern auch effizient für die Maschine zu gestalten. Die Begriffe Kommunikation und Interaktion sowie Intelligenz sind eng miteinander verknüpft, wenngleich sie auch nicht das gleiche bedeuten. Intelligenz kann sich in Form von Kommunikation und Interaktion äußern, überspitzt gesagt: Intelligenz ist erst durch Kommunikation und Interaktion wahrnehmbar.

DAS PROBLEM DER „NATÜRLICHEN“ KOMMUNIKATION. Ein großes Problem liegt in dem Begriff „natürliche Kommunikation“ verborgen: Was macht Kommunikation oder Sprachen „natürlich“? Die gängigste Definition sagt, daß natürliche Sprachen die sind, die Menschen untereinander zur Kommunikation entwickelt haben. Also werden auch natürliche Sprachen erlernt. Menschen erlernen Sprachen typischerweise in der frühen Kindheit oder — formalisierter — während ihrer Schulausbildung. Offen-

sichtlich gibt es mehrere natürliche Sprachen, die in Alphabet, Vokabular und Syntax voneinander abweichen; auch in geographisch und kulturell ähnlichen Gebieten, wie z.B. Westeuropa. Es existieren also verschiedene Sprachen, die zur Beschreibung und Interaktion in einer im wesentlichen gleichen Welt entstanden sind. Daraus folgt direkt, daß die natürliche Kommunikationsform der z.B. Franzosen für einen Deutschen fremd ist, solange jedenfalls, bis er französisch erlernt hat. Dieses Erkenntnis läßt sich auf verschiedene Ebenen übertragen, so z.B. auf Leute verschiedener Berufe oder Wissenschaftsrichtungen, die jeweils eine Fachsprache innerhalb ihrer Landessprache entwickelt haben, die in großen Teilen nicht von Mitgliedern anderer Berufe oder Wissenschaftsrichtung verstanden werden. Anzunehmen, daß es *eine* universelle „natürliche Sprache“ oder „natürliche Kommunikation“ gibt, wird durch die angeführten Argumente nicht unterstützt. Jede Sprache hat ihre Stärken und Schwächen bei der Kommunikation gewisser Konzepte. Sprachen können deshalb natürlich oder auch fremdartig in Hinblick auf Konzepte und Sprecher sein (siehe auch: [Bux90]).

Eine im Wortsinn „natürliche Kommunikation“ kann nur in einem System erreicht werden, wenn die verwendete Sprache adäquat für die Aufgabe, den Sprecher und den Rezipienten ist, die beteiligten Kommunikatoren sich also in einer gemeinsamen Sprache bedienen, die der Aufgabe angemessen ist. Daraus folgt die Einsicht, daß natürliche Kommunikation in einem Gesamtkontext betrachtet werden muß.

Auf der anderen Seite wird in der MMK [Lau90] aber die Frage nach einer gemeinsamen Kommunikationsstruktur auch wie folgt beantwortet:

„But is there an untapped resource there, a language resource that users bring to the system on first encounter that could provide the basis for such ‚natural‘ dialogues?

Yes!“ Bill Buxton [Bux90], Seite 408.

Dabei bleibt Buxton eine direkte Definition dieser „language resource“ schuldig, führt aber eine Reihe von Beispielen an, wie die metaphorische Verwendung von Symbolen und Begriffen sowie Phrasenstrukturen diesen Zweck erfüllen können.

Durch diese Sichtweise ist die Interaktion eines Menschen mit einem Computer alles andere als natürlich, denn die „natürliche“ Kommunikationsform des Rechners ist ein Artefakt, also künstlich. Die Aufgabe ist also, eine dem menschlichen Kommunikator für ihn natürlich *erscheinende* Kommunikationsform mittels möglichst universeller Sprachmittel zu planen.

GESTEN IN DER MENSCHLICHEN KOMMUNIKATION. Wenn Menschen miteinander reden, benutzen sie verschiedene Typen von Gesten zu teilweise recht unterschiedlichen Zwecken. Wie von verschiedenen Seiten experimentell untersucht, scheinen Gesten aber einheitlich genug zu sein, um eine Kommunikation sogar zwischen Menschen verschiedener Kulturen zu unterstützen, auch wenn einzelne Gesten in verschiedenen Kulturen diametrale Bedeutungen haben. Andererseits aber sind Gesten komplex und vielschichtig genug, um eine eigene Sprache zu formen. Die syntaktische und semantische Beschaffenheit einer solchen formal zu beschreibenden Gestensprache zu ermitteln, ist gegenwärtig ein aktueller Forschungsgegenstand. Wie Menschen ihre Ausführungen mit Gesten untermalen, hat z.B. Wexelblat [Wex94] anhand der Beschreibung von zuvor betrachteten Spielfilmszenen untersucht. Dabei kommt er zu einigen auch im Betrachtungskontext wichtigen Ergebnissen. Zum einem laufen Sprache und Gestik selten synchron zueinander ab, d.h. eine Geste erscheint meist kurz vor oder seltener nach der sprachlichen Äußerung. Zum anderen werden in der gleichen Situation von verschiedenen Personen verschiedene Typen von Gesten benutzt, um den identischen Sachverhalt zu untermalen oder zu schildern. Auch der Grad des Gesteneinsatzes hängt signifikant von der beobachteten Person ab. Wexelblat macht eine Personengruppe mit 6-12 Gesten pro Szenenschilderung sowie eine andere mit 2-4 Gesten pro Szenenschilderung aus. Es läßt sich noch nicht vorhersagen, wann sprachliche Mittel allein oder gestenunterstützt verwendet werden.

GESTEN IN MENSCH–MASCHINE–SCHNITTSTELLEN. Heutzutage gibt es in einigen Bereichen große Probleme, adäquate Mensch–Maschine–Schnittstellen zur Verfügung zu stellen. Hier liegt ein Problem darin, komplexe Informationen dem Benutzer adäquat zu präsentieren und aus seinen kontextbezogenen, unscharfen Kommunikationen systemverwertbare Information zu bilden. Ein solcher Aufgabenbereich betrifft Systeme, aus dem CAD oder CIM Bereich, die professionellen Benutzern eine einfachere Schnittstelle präsentieren möchten, damit die tägliche Arbeit leichter oder effektiver wird. Für derartige Anwendungen der multimodalen Mensch–Maschine–Kommunikation kann die hier vorgestellte Arbeit von Nutzen sein. So ist es z.B. denkbar, aus Gesten Objektreferenzen oder spatiale Referenzen zu extrahieren, die sprachlich kommunizierte Eingaben ergänzen. Eine weitere breite Anwendung ist in immersiven Systemen der virtuellen Realität absehbar, wo Maus oder Tastatur sich nicht mehr in Reichweite oder Blickfeld befinden. Sprachliche wie auch gestische Eingabeformen sind dort vielversprechende Alternativen. In den letzten Jahren werden auch zunehmend Systeme entwickelt, die computergraphische Darstellung virtueller Figuren benutzen um dem Benutzer zu gestikulieren. Dabei werden Darstellungen von Körpergestik, Arm- und Handbewegungen sowie Mimik benutzt [Thó96].

1.2 Motivation

Gestenerkennung ist Grundlage einer weiterentwickelten Interface-Technik [ABF+94]. Insbesondere die Betrachtung einer instruktionsbasierten, mittelbaren Bedienung — im Gegensatz zum Paradigma der direkten Manipulation virtueller Objekte — erweist sich als immer bedeutsamer. Zhai et. al. [ZMR97] belegen experimentell, daß viele Menschen große Probleme haben, Objekte in sechs Freiheitsgraden zu handhaben. Im weitem basieren die meisten Interfaces auf einer Metapher, die eine direkte Manipulation nur von Ikonen (engl.: Icons) usw. — also metaphorischer Repräsentation von Dateneinheiten oder Funktionsabläufen — zuläßt und somit die Anzahl der Freiheitsgrade durch die Metapher auf einige wenige beschränkt. Diese beschränkte Anzahl an Freiheitsgraden (z.B. 2) wird dann meist direkt auf ein Eingabemedium (z.B. „Maus“) abgebildet. Gestenerkennung beinhaltet aber das Potential, die Anzahl der Freiheitsgrade nicht mehr direkt vom Eingabegerät abhängig zu machen, sondern sie fallweise von der Kommunikationssituation und der Interaktionsmetapher zu wählen.

KRITIK. Ein häufige Kritik der gestenbasierten Mensch-Maschine Kommunikation formuliert Wexelblat [Wex98], in der er ausführt:

„It is common in the literature to see systems reported which achieve recognition rates of 60-80% and this is reported as success. [...] When the error rate for an input mode becomes significant, it is hard to convince people to use it. After all, the recognition rate for keystrokes is 100%“ Alan Daniel Wexelblat, [Wex98], Seite 10.

Der Benutzer eines Systems, das neue Schnittstellen benutzt, schreibt diesem typischerweise eine „höhere Intelligenz“ in Form einer allgemein höheren Funktionalität zu, was häufig zu Enttäuschungen führt. Bei Sprach- oder gestenverarbeitenden Systemen wird meist implizit vorausgesetzt, daß sie die Wünsche des Benutzers verstehen können. Bei einem fairen Vergleich ist unter dieser Voraussetzung betrachtet die Erkennungswahrscheinlichkeit der Tastatur alles andere als 100%. Dies kann in einem Selbstversuch jeder schnell herausfinden: Diese Seite ist fehlerfrei abzutippen, ohne die Lösch- oder Rückschrittstaste zu benutzen. Danach wird die Fehlerrate anhand der Anzahl der abweichenden Wörter bewertet. Auch wird gerne vergessen, daß ein Tastendruck auch bei einer WIMP¹⁾- oder alphanumerischen Schnittstelle im Kontext

1. WIMP: Windows, Icons, Mouse, Pointer, die gebräuchlichste graphische Schnittstellenmetapher aber auch engl. Feigling

mit der Position der Eingabemarke und vorhergehenden Tastendrücken steht, also ebenfalls interpretiert werden muß. Die vordergründig so sichere Tastatureingabe hat die prinzipiell gleichen Probleme wie viele andere Eingabemodi auch. Die Basissensorik — Taste — hat tatsächlich eine Erkennungsrate von fast 100%. Leider ist dies nicht auf das Gesamtsystem „Tastatur-basierte Eingabe“ übertragbar.

1.3 Ziele

In dieser Arbeit soll ein Beitrag zur Erschließung einer natürlichen Kommunikationsform — der Gestik — für die Interaktion des Menschen mit komplexen rechnerbasierten Systemen geleistet werden. Deswegen wird hier nun ein Ansatz verfolgt, der als Ziel hat, eine instruktionsbasierte Schnittstelle mit Information über Konfigurationen der oberen Gliedmaßen zu versorgen. Aus informationstechnischer Sicht ist eine hohe Abhängigkeit von einer Aufgabe für eine Schnittstelle unerwünscht. Vorzuziehen sind — nach dem generellen Prinzip der minimalen Festlegung — immer die allgemeineren Lösungen. Im Kontext der Gestenerkennung stellt sich also die Frage, inwieweit es möglich ist, sinnvoll zwischen anwendungsdomänenspezifischeM Wissen und Methoden und allgemeineM Wissen und Methoden zur Gestenerkennung zu unterscheiden.

Dabei soll das globale Fernziel verfolgt werden, einen Rahmen zu definieren, der durch geeignete Parametrisierung an verschiedene Anwendungsdomänen angepaßt werden kann. Mit Blick auf dieses Fernziel muß konkrete Arbeit aber auch einen konkreten Rahmen haben, der als prototypische Arbeitsumgebung dient. Dieser Rahmen wird durch ein virtuelles Konstruktionsszenario gegeben, auf das sich die Beispiele beziehen werden.

In dieser Arbeit wird eine Aufstellung von Ergebnissen verschiedener Forschungsgebiete zu einer einheitlichen Sichtweise integriert, und ein darauf basierendes informationstechnisches System vorgestellt. Dabei wird ein kognitionswissenschaftlich motivierter Ansatz benutzt. Die Konzepte der menschlichen Kognition, sofern erforscht, werden aus zwei Gründen benutzt. Zum einen ist bekannt, daß Menschen Gesten verstehen können, also ist die menschliche Kognition ein Vorbild des angestrebten Systems. Zum anderen ermöglicht ein solcher Ansatz den Erkenntnisaustausch mit anderen Disziplinen, namentlich der Computer- und Psycholinguistik, der Biokybernetik und der medizinischen Psychologie. Ein solch interdisziplinäres Vorgehen bereichert die Einzelwissenschaften und fördert ein wissenschaftliches Gesamtbild.

Die Beschäftigung mit der menschlichen Gestik als Eingabemodalität für ein technisches System ist eine vielversprechende und wenig erforschte Technologie, die wissenschaftlich und technisch herausfordert. Zum einen ist die Erforschung der menschlichen Gestik ein aktuelles Thema der Linguistik, zum anderen ist das maschinelle Verstehen menschlicher Äußerungen ein Kernthema der Künstlichen Intelligenz. Die maschinelle Nutzung der menschlichen Gestik erfordert eine eigene Sensorik und Sensorauswertung, Wissen über den menschlichen Körper, Wissen über adäquate Repräsentationsformen dieser Information und Wissen über Form, Struktur und Bedeutung von Gesten. Der Kern der Herausforderung liegt darin, einen Bogen zu spannen, von der vermuteten Intention des Anwenders über das Messen der geäußerten Bewegung bis zur Interpretation der Bewegung als Geste in einem Kontext. Diese Arbeit wird nicht die letztendliche Lösung dieser Frage präsentieren können, wohl aber einen Ansatz beschreiben, der sich dieser Herausforderung annimmt.

Die *Simulation* der menschlichen Kognition hingegen ist ausdrücklich kein Ziel, das in dieser Arbeit verfolgt wird. Viel mehr steht eine informationstechnisch adäquate Umsetzung der gefundenen Konzepte im Vordergrund, so daß eine *Anwendung* des Systems in einem größeren Forschungskontext möglich wird. Diesem Anwendungssystem sollen Symbole, deren Semantik theoretisch fundiert ist, in einer internen Repräsentation geliefert werden, so daß es in der Lage ist, gemäß dem so geäußerten Benutzerwunsch, Funktionsausführungen zu aktivieren.

Auch die *Produktion* von Gestik, z.B. mittels computergraphischer Darstellungen von Körpergestik, Arm- und Handbewegungen oder Mimik wird in dieser Arbeit nicht verfolgt.

1.4 Überblick

Im Kapitel „Theoretische Grundlagen“ auf Seite 11 werden verschiedene Grundlagen zum Verständnis von Gesten und Gebärden gelegt und eine Reihe von Fragen formuliert, die dann im weiteren Verlauf dieses und der nächsten beiden Kapitel beantwortet werden. Dabei werden die Rollen, Kategorien und Strukturen von Gesten sowie ihr Verhältnis zur Sprache betrachtet, um Gesten beschreiben zu können.

Da Zeitbegriffe neben den geometrischen Eigenschaften von Gesten eine wichtige Rolle spielen, werden diese im Kapitel „Zeitbegriffe“ auf Seite 31 behandelt. Um den Betrachtungsgegenstand „Geste“ adäquat niederschreiben zu können, wird im Kapitel „Gestennotationssysteme“ auf Seite 37 eine Übersicht über bekannte Notations-

systeme gegeben. Dabei wird eine Auswahl relevanter Systeme in diesem Themenbereich kurz in Hinsicht auf diese Arbeit diskutiert.

Im Kapitel „Konzeption eines Gestenerkenners“ auf Seite 71 werden Konzepte vorgestellt und diskutiert die dann in einem Architekturvorschlag für einen Gestenerkennungsrahmen münden. Dabei wird für die einzelnen Systemkomponenten die Fundierung in der Gesamtkonzeption und der bereits dargelegten Theorie beschrieben. Das Kapitel „Vorschlag einer Realisation des Gestenerkenners“ auf Seite 101 stellt dann die Realisierung des Rahmensystems auf den Grundlagen der diskutierten Konzepte vor. Dabei wird das Funktionsprinzip der einzelnen Systemkomponenten sowie ihres Zusammenspiels beschrieben.

Um einen Eindruck von der Funktionsweise von mit diesem Rahmensystem erstellter Systeme zu bekommen werden dann im Kapitel „Anwendung“ auf Seite 143 zwei Prototypen vorgestellt und diskutiert. In diesem Kapitel werden wie in den Anhängen ab Seite 155 auch technische Einzelheiten betrachtet.

Zuletzt findet sich im Kapitel „Résumé“ auf Seite 151 ein kritischer Rückblick mit einer Zusammenfassung, die noch einmal in kondensierter Form, die einzelnen Aspekte der Arbeit unter verschiedenen Gesichtspunkten aufgreift, sowie einen Ausblick auf neue Forschungsfragen und nun beschreibbar erscheinende Wege gibt.

Kapitel 2

Theoretische Grundlagen

Bevor argumentiert werden kann, wie ein gestenverarbeitendes System aussehen soll, müssen einige Grundlagen dargelegt werden. Diese stellen sich als eine Menge von theoretischen Fragen dar:

1. Welche Funktion haben Gesten?
2. Wie können Gesten kategorisiert werden?
3. Wie sieht die innere Struktur von Gesten aus, sofern sie überhaupt existiert?
4. Wie verhalten sich Gesten zur Sprache?
5. Wie kann diese Information repräsentiert werden?
6. Wie kann Zeit gefaßt bzw. modelliert werden?
7. Wie können gestische Äußerungen aufgeschrieben werden?

Und nicht zuletzt: Was sind Gesten und Gebärden eigentlich?

Diese letzte Frage ist immer wieder Gegenstand von Diskussionen in der Literatur und auf Workshops. Ein grundlegendes Problem beim Umgang mit den Begriffen *Geste* und *Gebärde* ist der Umstand, daß zwar Gebärden als intentionalisierte Gesten verstanden werden, aber der Begriff *Geste* keine allgemein akzeptierte Definition hat. Dennoch werden Gesten in verschiedenen Kontexten kategorisiert.

Für diese Arbeit seien zur Vereinfachung *Gesten* und *Gebärden* in dem Oberbegriff *Gesten* zusammengefaßt — womit er dem englischen Begriff „gesture“ weitgehend entspricht — und im weiteren die Bewegungen der oberen Gliedmaßen (Arme, Hände) gemeint. Dies stellt keine allgemeine Definition des Begriffes dar, liefert aber doch eine das Verständnis vereinfachende Konvention. Im weiteren werden noch zwei Verfeinerungen definiert:

Definition: coverbale Gesten: Als coverbale Gesten sollen hier „coverbal gesture“ [Wex94] verstanden werden. Jene Gesten also, die im Kontext multimedialer¹⁾ Äuße-

1. Siehe auch [Len98] und Anhang „Multimodal versus Multimedial“ auf Seite 155.

rungen auftreten und typischer Weise mit einer sprachlichen Äußerung zeitlich korreliert auftreten.

Definition: EH-Gesten: Verfeinernd soll der Begriff *Geste* hier nicht auf den Werkzeuggebrauch ausgedehnt werden, wie es in einigen Untergebieten der Gestenverarbeitung, so z.B. bei den „pen-based systems“, durchaus üblich ist. Hier wird sich auf die sog. „open / empty hand gesture“ bezogen. Sie sind insofern nicht mit den Elementen einer Gebärdensprache zu verwechseln, als daß ihnen der syntaktische Kontext zu andern Gebärden fehlt, der eine Gebärdensprache auszeichnet.

Soweit eine Unterscheidung nötig ist, werden die „open / empty hand gestures“ im weiteren als EH-Gesten und die „coverbal gestures“ als coverbale Gesten bezeichnet werden, ansonsten ist ihre Schnittmenge gemeint.

Die meisten der obigen Fragen beleuchten den Begriff „Geste“ aus verschiedenen Blickwinkeln, um trotz fehlender direkter Definition den Kern des Begriffes, wie er hier Verwendung findet, darzustellen.

In den nächsten Abschnitten werden die Fragen Nr. 1 bis Nr. 5 vom Anfang dieses Abschnittes behandelt. Die Frage Nr. 7 wird dann im Kapitel „Zeitbegriffe“ auf Seite 31 und die Frage Nr. 7 im Kapitel „Gestennotationssysteme“ auf Seite 37 behandelt werden.

2.1 Gesten aus Sicht des Produzenten

Die Funktion einer Geste liefert eine erste, grobe Unterteilung. In [CC95] werden drei grundlegende funktionale Rollen von Gesten dargelegt:

- Die *ergodische* Funktion ist mit der Darstellung von Arbeit verknüpft. Sie entspricht der Fähigkeit, die Umwelt zu verändern, Artefakte zu schaffen, also Dinge mittels direkter Manipulation zu formen oder zu bewegen.
- Die *epistemische* Funktion erlaubt es, die Umwelt mittels der Sinne wahrzunehmen. Die sinnliche Wahrnehmung erlaubt, Materialeigenschaften wie Temperatur, Rauheit usw. zu erfahren.
- Die *semiotische* Funktion ist es, sinnvolle Informationen zu kommunizieren. Die Struktur einer semiotischen Geste ist konventionalisiert und resultiert aus gemeinsamer kultureller Erfahrung. Typische Vertreter dieser Funktionsgruppe sind das Abschiedswinken, die Bewegungen von Flugzeugeinweisern auf Flughäfen oder „Daumen hoch“, um Einverständnis oder Gnade zu signalisieren.

Alle diese Funktionen können durch den Gebrauch von Werkzeugen verstärkt werden. Als Beispiel mag hier die Verwendung eines Taschentuches beim Abschiedswinken dienen. Doch wird — wie erwähnt — dieser direkte Werkzeuggebrauch im weiteren nicht behandelt.

Diese erste Unterteilung geht von der die Geste äußernden Person, dem *Produzenten* oder informationstheoretisch „*Sender*“, aus. Es ist somit leicht vorstellbar, Gesten auch aus weiteren Blickwinkeln zu betrachten.

2.2 Gesten aus Sicht des Rezipienten

Betrachten wir nun Frage Nr. 2, „Wie können Gesten kategorisiert werden?“ auf Seite 11. Die Anfänge der Untersuchung von Gestik mit wissenschaftlichen Methoden gehen auf D. Efron in der Mitte der 30er Jahre dieses Jahrhunderts zurück [Efr41],[Efr72]. Er untersuchte das Konversationsverhalten von jüdischen aus osteuropäischen Ländern stammenden und italienischen Einwanderern in New York City.

Kendon setzte diese Untersuchungen aus der Sicht des *Rezipienten* — informationstheoretisch „*Empfängers*“ — fort [Ken86]. Er zeigte, daß Rezipienten signifikante „Ausflüge“ der Hände des Sprechers, die ob ihrer Ausführung als geplant wahrgenommen werden, als Gesten erkannten, obwohl der Kommunikator sich einer dem Rezipienten nicht verständlichen Sprache bediente. Bei Kendon findet sich auch früh die Theorie, daß Sprache und Gestik stark verwoben sind und Gesten insofern zweckmäßig erscheinen, als sie Sprache begleiten und sie anscheinend verstärken.

Er formuliert daher sein sog. „Kontinuum“, eine Hierarchie von verwandten Begriffen: Gestikulation → sprachhafte Gesten → pantomimische Gesten → Embleme → Zeichensprachen. Von links nach rechts betrachtet, nimmt die Gegenwärtigkeit von Sprache ab, die Anzahl der Merkmale für eine Sprache nehmen zu und individuell geprägte Gesten werden durch gesellschaftlich geformte Zeichen ersetzt. Kendon definiert Gesten als die Phänomene am linken Rand seines Kontinuums.

Gestikulation in Kendons Sinne sind sprachbegleitende, individuell geprägte Bewegungen der Arme und Hände. Ein Beispiel ist das Heben der Hand während der Sprecher sagt: „Das Eichhörnchen läuft den Baum hoch.“ Gesten in Kendons Sinne treten fast nie ohne Sprachbegleitung auf. *Sprachhafte Gesten* in Kendons Sinne sind in Form und Auftreten der Gestikulation ähnlich, aber unterscheiden sich insofern, daß sie grammatikalisch in die Äußerung integriert sind. Ein Beispiel ist die Äußerung: „Der Professor ist in Ordnung, aber seine Arbeitsgruppe [Geste]“; wobei die Geste den

grammatikalischen Platz eines Adjektives einnimmt. Bei *pantomimischen Gesten* steht die Hand für Gegenstände oder Aktionen, aber Sprache ist nicht unbedingt notwendig. Die Geste kann hier mit Schweigen oder onomatopoetischen Klangeffekten („Hups“, „Klick!“, usw.) begleitet sein. Darüber hinaus können Abfolgen von pantomimischen Gesten Sequenzen in einer Vorführung sein. Dies unterscheidet sie von der Gestikulation, wo Gesten keine kombinatorischen Eigenschaften besitzen. *Embleme* sind wohlgeformte Gesten bei Kendon, die hauptsächlich darauf abzielen, das Verhalten anderer Leute zu beeinflussen. Die Wohlgeformtheit ist die Eigenschaft, die Embleme von sprachhaften und pantomimischen Gesten sowie der Gestikulation unterscheidet. Die *Zeichensprache*¹⁾ am rechten Rand von Kendons Kontinuum, wie z.B. die Amerikanische Gebärdensprache (ASL), sind vollständige linguistische Systeme mit Segmentierung, Kompositionalität, einem Lexikon, einer Syntax, Regeln zur wohlgeformten Definition sowie einer Gemeinschaft von Benutzern.

McNeill erweitert dies noch:

„Contrary to the assumption which seems widespread, that gestures are part of a separate system of ‚non-verbal communication‘, only incidentally connected to speech, we find that gesture correlates closely with meaning on several levels of language organization.“ [McN82], Seite 275

Und er führt in [McN92] eine Theorie vor, in dem Sprache und Gestik einen gemeinsamen Ursprung haben. Er faßt die Entstehung einer multimodalen Äußerung als einen mikrogenetischen Prozeß auf, bei dem sich aus einem Kristallisationspunkt („growth point“) sowohl die Sprache mit ihrer immer besser erforschten, grammatikalischen Struktur als auch die Gestik entwickelt. Dieser Punkt repräsentiert das als nächstes zu kommunizierende Konzept abstrakt, ist aber in sich instabil und emergiert in Sprache und Gestik, die somit unterschiedliche Aspekte des Konzeptes zum Ausdruck bringen. McNeill selbst vergleicht diesen Punkt insofern mit Vygotskys psychologischem Prädikat [Vyg62], dem Punkt größter Relevanz.

1. Hier ergibt sich ein Problem mit der Übersetzung des engl. Begriffs „sign language“. Dies kann im dt. durchaus „Zeichensprache“ bedeuten aber auch „Gebärdensprache“. Dabei bedeutet Zeichensprache eine Kommunikationsform die auf wenigen fest definierten Gesten für eine engumrissene Aufgabe beruht, wie z.B. die Kranführersprache oder die Tauchersprache. Als eine Gebärdensprache wird im allgemeinen die natürliche Kommunikationsform einer Gruppe von Gehörlosen bezeichnet, sofern sie einige spezifische linguistische Merkmale aufweist (siehe z.B. [ASW95]).

2.2.1 Psychologisch/linguistische Systeme

Alan D. Wexelblat hat eine Zusammenfassung von bekannten Forschern in diesem Feld und ihrer Kategorisierungen in [Wex95] aufgestellt. Kendon [Ken86] begann mit dem Versuch herauszufinden, was Menschen wahrnehmen, wenn sie Gesten betrachten. Er legt nahe, daß Äußerungen als integrale Einheiten geplant werden, die aus Sprech- und Gestenkomponenten bestehen. McNeill und Levy (und später McNeill alleine) basierten ihre Arbeit [McN82] auf einer Theorie, nach der Kommunikationspartner senso-motorische Modelle ihrer Umgebung benutzen, wobei Gesten die Formen transportieren, die in diesen Modellen benutzt werden. Auch betrachteten sie die Interdependenzen zwischen Gesten und Sprachen besonders. Dabei fanden sie, insbesondere aber McNeill [McN92] im Widerspruch zu anderen Forschern in dem Gebiet, daß Gesten hochgradig mit verschiedenen sprachlichen Ebenen korrelieren. Rimé und Schiaratura konnten darlegen [RS91], daß sich Gestenbildung und der Sprechvorgang gegenseitig beeinflussen und beziehen so eine starke Position gegen den behaviouristischen Ansatz bezüglich Sprache. Sie führen aus, daß, wenn die motorischen Effekte des Erzeugens einer Geste den Inhalt eines Sprechaktes beeinflussen, das Unterscheidungskriterium zwischen Stimulus und Antwort im behaviouristischen Sinne verlorengeht. Efron begann seine Arbeit [Efr41] mit Beobachtungen über das Verhalten jüdischer und italienischer Einwanderer in New York. Der bedeutendste Aspekt für die Klassifikation von Gesten in seiner Arbeit ist die Orientierung auf den „Empfänger“ der Geste.

Rimé & Schiaratura [RS91]	Kendon [Ken86]	McNeill & Levi [McN82]	Efron [Efr41]	Beschreibung
physiographisch	physiographisch	ikonisch	kinetographisch	Stellt den Inhalt der Sprache dar
ikonisch	ideographisch	metaphorisch	ideographisch	Stellt die Ideen des Sprechers, aber nicht direkt den Inhalt der Sprache dar
sprachmarkierend	Gestikulieren	Taktschläge	Taktstock	Markieren den Rhythmus der Sprache

Tabelle 1: Zusammenfassung von Kategorisierungssystemen nach [Wex95]

Rimé & Schiaratura [RS91]	Kendon [Ken86]	McNeill & Levi [McN82]	Efron [Efr41]	Beschreibung
symbolisch	autonom	symbolisch	symbolisch / emblematisch	Standardisierte Gesten, vollständig ohne Sprachanteil
deiktisch	— keine —	deiktisch	— keine —	Zeigen auf Sache oder Bereich

Tabelle 1: Zusammenfassung von Kategorisierungssystemen nach [Wex95] (Forts.)

Wexelblats Auflistung ist in Tabelle 1 auf Seite 15 zu sehen. Dazu kommen noch etliche Spezialkategorien, wie z.B. die nach Brian Butterworth [BB78] benannte „Butterworth'schen Gesten“, die nicht die herausragenden Teile von in der Sprache betonten, sondern ausgelassene Teile markieren. Oder die „Lakoff'schen Gesten“, nach Georg Lakoff benannt, der in [LJ80] notierte, daß wir häufig verbale Metaphern räumlich darstellen. Diese Geste repräsentiert die räumlich-metaphorische Manifestation eines internen Zustandes, also gewissermaßen Zeigen in einem „inneren Zustandsraum der Metaphern“. Ein Beispiel für eine Lakoff'sche Geste wäre daher: „*Ich bin am Boden zerstört.*“, begleitet von einer nach unten gerichteten Handbewegung.

2.2.2 Technisch fundierte Systeme

Unter den Systemen, die unter Einbeziehung technischer Anforderungen entstanden sind, läßt sich ein Spannungsfeld aus semiotischer und technischer Fundierung ausmachen. Um dies darzustellen, seien im folgenden einige Systeme vorgestellt.

INGENIEURSANSATZ. Im Umfeld der technisch ausgerichteten Soft- und Hardwareentwicklung werden sehr oft definierte Zeichensprachen benutzt, die ein festes Codebuch benutzen und die Zeichen damit direkt auf Kommandos des Systems abbilden. Dabei werden selten mehr als 30 verschiedene Gesten unterschieden, wenngleich die Autoren immer besonderen Wert darauf legen, daß eigentlich beliebig viele Gesten trainiert werden könnten. Auch wird die Hand häufig als 3D-Maus verwendet, um mehr oder minder direkte Manipulation von virtuellen Objekten zu ermöglichen. Dies gliedert sich somit als Erweiterung in das bekannte „direct manipulation interface“-Paradigma der Mensch–Maschine–Kommunikation (MMK) ein. Als Beispiel kann hier das „Finger-fliegen“ in VR-Systemen dienen [Bor94]. Dabei wird dem Rechner aus-

gehend von der Faust mit dem gestreckten Zeigefinger die Richtung und mit dem Krümmungswinkel des Daumens zu den Knöcheln hin die Geschwindigkeit einer gewünschten Bewegung mitgeteilt.

AHIG KLASSIFIKATION (NACH [WEX95]). In der „Advanced Human Interface Group“ (AHIG) des Media Laboratory des Massachusetts Institute of Technology wurde das in Tabelle 2 dargestellte System benutzt. Die dort dargestellten Kategorien werden als

AHIG Kategorie	Beschreibung
symbolisch / modalisierend	„direkte“ Gesten, da direkte Abbildung von Geste auf Bedeutung
pantomimisch	Hände bedienen imaginäres Objekt
ikonisch / Objekt	Hand/Arm nimmt die Stelle des darzustellenden Objektes ein
deiktisch / Lakoff'sch	Zeigen auf Sache oder Bereich, Benutzung des Umfeldes
Taktschlag / Butterworth	markiert den Sprachrhythmus
— keine —	Stellt Ideen/Metaphern dar

Tabelle 2: AHIG System

die nützlichsten im Kontext der Interpretation bezeichnet. Jede Klasse kann durch eine andere Methode interpretiert werden, d.h. eine andere Funktionalität der zu bedienenden Applikation kann angesprochen werden.

AG-WBS SYSTEM. Die Arbeitsgruppe „Wissensbasierte Systeme“ der Technischen Fakultät der Universität Bielefeld, hat ebenfalls ein System entwickelt (siehe Tabelle 3).

Kategorie	Einordnung	Beschreibung
I	1. deiktisch	Räumliche Referenz auf Ort oder Objekt

Tabelle 3: WBS System nach [LW98]

Kategorie	Einordnung	Beschreibung
II	1. mimetisch 2. ikonisch 3. objektbezogen	Benutzung der oberen Gliedmaße als Platzhalter für Aktionen oder Verhalten von Objekten oder Zuständen
III	1. physiographisch 2. kinetographisch 3. pantomimisch	Illustriert, wie etwas zu benutzen ist, ohne das reale Objekt zu verwenden
IV	1. symbolisch 2. modus-setzend 3. emblematisch	Enthält eine eindeutige Botschaft, ändert den Modus, unter dem die verbale Äußerung zu interpretieren ist.
V	1. ideographisch 2. metaphorisch 3. Lakoff'sch	Repräsentiert die räumlich-metaphorische Manifestation eines internen Zustandes.
VI	1. Taktschläge 2. Gestikulieren 3. sprachmarkierend 4. selbstregulierend	Markieren des Sprechrhythmus, beziehungsweise der Akzentuierung

Tabelle 3: WBS System nach [LW98] (Forts.)

Bei der Entwicklung dieses Systems wurde strikt der Betrachterstandpunkt eingenommen, um die Geste aus der Sicht des Rezipienten einzuordnen. Dies gebot sich, da dieses System als Grundlage eines technischen Systems zur Ausnutzung von Zeigegesten gedacht ist, also von der Sicht eines künstlichen Rezipienten aus zu einem fundierten Design führen soll. Dabei wurde in Abgrenzung zu [Efr41] und [Ken86] aber in Übereinstimmung mit [McN92] und [Wex95] der Standpunkt eingenommen, daß deiktische Gesten in einer eigenen Kategorie darzustellen sind. Im Vergleich zu der AHIG-

Klassifikation wird hier der Interpretationsstandpunkt noch konsequenter vertreten und auf Gesten fokussiert, die zur Objektmanipulation dienen können.

2.2.3 Probleme der Taxonomie und Systematik

Um die Gefahr von Mißverständnissen einzugrenzen, ist eine eindeutige, herleitbare Bezeichnung durch einen Fachausdruck wünschenswert. Diese allgemein gültige Forderung wird durch Taxonomie und Systematik entsprochen.

Definition: Taxonomie: (gr.: *taxis*, Ordnung und gr.: *nomos*, Gesetz) Taxonomie ist die Theorie und Praxis der Klassifikation

Definition: Systematik: (gr.: *systema*) Systematik ist die wissenschaftliche Untersuchung der Arten und der Vielgestaltigkeit der zu untersuchenden Entitäten und sämtlicher Beziehungen zwischen ihnen.

Eine Taxonomie stellt eine Prioritätsfolge der Dimensionen eines hochdimensionalen Bezeichnungsraumes auf. Als wesentliches Einordnungskriterium gilt eine Menge gemeinsamer Merkmale, wobei aber Hyponyme sich maximal unterscheiden und eine disjunkte Partitionierung aller Merkmale ihres Hyperonyms darstellen sollten. Es ist ausdrücklich nicht gefordert, daß die verschiedenen Kategorien eines Taxons die gleichen Unterscheidungsmerkmale haben, sie sollten nur von gleicher Priorität sein.¹⁾

Das grundlegende Problem der Taxonomie von Bewegungen ist, daß die Menge aller Merkmale nicht bekannt ist und es im weiteren auch nicht klar ist, ob eine Klassifikation von der Form, der Funktion oder der Bedeutung her erfolgen sollte. Diese Unsicherheit wird durch den Aufbau der oben vorgestellten Schemata belegt. In ihnen wird weder die Forderung nach maximal diskrepanten Hyponymen noch nach einer disjunkten Partitionierung immer erfüllt, meist nicht einmal thematisiert. Trotzdem sei hier ein *Versuch* gewagt, in dem nur die in dieser Arbeit relevanten Taxa ausgeführt werden. Aus oben genannten Gründen leidet auch diese Hierarchie unvermeidlich

1. Die Biologie wartet mit der vielleicht ältesten, zumindest der bekanntesten Taxonomie auf, die dort für die Erfassung der verschiedenen Organismen benutzt wird. Diese biologische Taxonomie wurde vom schwedischen Naturforscher Carl von Linné (Linnaeus) begründet. In der Linnaeischen Hierarchie werden vor allem folgende Taxa benutzt: Reich (*regnum*), Stamm (*phylum*), Klasse (*classis*), Ordnung (*ordo*), Familie (*familia*), Gattung (*genus*) und Art (*species*). Der Wolf wird demnach als *Animalia Chordata Mammalia Canivora Canidae Canis lupus* und die Honigbiene als *Animalia Anthropoda Insecta Hymenoptera Apidae Apis mellifera* bezeichnet

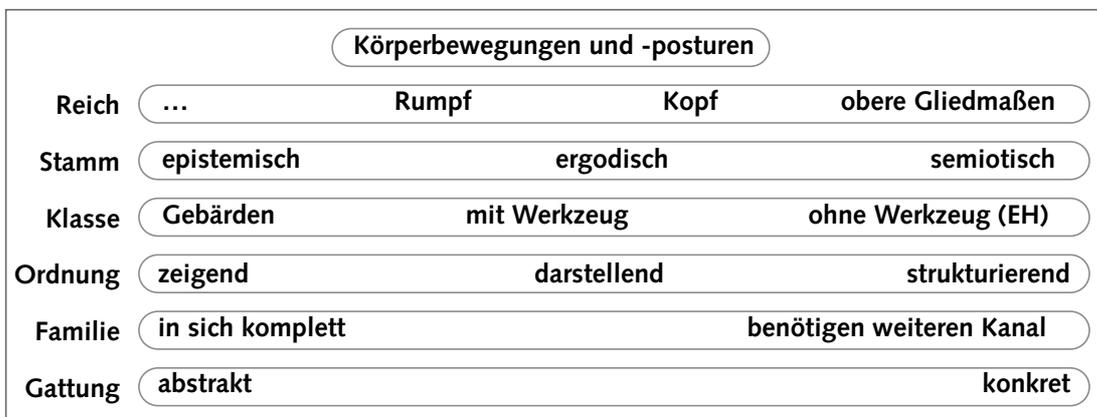


Abbildung 2: Vorschlag für eine Klassifizierungshierarchie

unter den dargestellten Unzulänglichkeiten. In Abbildung 2 ist eine Hierarchie und ihre Taxa angegeben, die zu einer Klassifizierung der Arten führt. Eine Klassifikation der schon erwähnten Begriffe ist in Tabelle 4, aufgeführt. Die Ebenen Klasse, Stamm

Art	Gattung	Familie	Ordnung
modus-setzende	konkret	komplett	darstellend
emblematisch	abstrakt	komplett	darstellend
mimetisch	konkret	weiterer Kanal	darstellend
pantomimisch	abstrakt	weiterer Kanal	darstellend
Taktschläge	konkret	weiterer Kanal	strukturierend
Butterworth'sch	abstrakt	weiterer Kanal	strukturierend
deiktisch	konkret	weiterer Kanal	zeigend
Lakoff'sch	abstrakt	weiterer Kanal	zeigend

Tabelle 4: Versuch der Klassifizierung einiger Gestenarten

und Reich sind dabei konstant mit den jeweiligen Werten „ohne Werkzeug“, „semiotisch“ und „obere Gliedmaßen“ belegt und werden deshalb nicht in der Tabelle aufgeführt.

Diese Taxonomie erlaubt es, bekanntes einzusortieren und unbekanntes zu benennen. Eine solche Taxonomie kategorisiert jeweils eine Geste, ohne explizit auf eine unter Umständen vorliegende innere Struktur einzugehen. Eine solche innere Struktur jedoch könnte auf einfach zu erkennende Elemente zurückzuführen sein.

2.3 Innere Struktur von Gesten

Neben dem semiotischen Gehalt besitzt eine Geste auch eine Form. Diese Form gliedert sich in der Zeit in verschiedene Phasen, die von verschiedenen Forschern untersucht und beschrieben wurden. Efron [Efr41] beschreibt die klassische Aufteilung in Vorbereitungs-, Ausführungs- und Rückstellungsphase. Diese Einteilung erweist sich bei näherer Betrachtung als zu grob und daher werden im weiteren einige Regelsysteme (Grammatiken) vorgestellt. Eine erste solche Grammatik schlug [McN92] vor, hier in Abbildung 3 dargestellt. Eine Grammatik für die syntagmatische Organisation in Bewegungsphasen von Gesten ist von Kita et. al. in [KGH98] erörtert und in Abbildung 4 auf Seite 22 dargestellt. Der Schlag oder Apex¹⁾ einer Geste ist durch die charakteristischen Merkmale einer Geste gekennzeichnet, eben jene Merkmale, die Kendon im folgenden meinte:

„...for an action to be treated as a 'gesture' it must have features which make it stand out as such.“ [Ken86], Seite 26²⁾

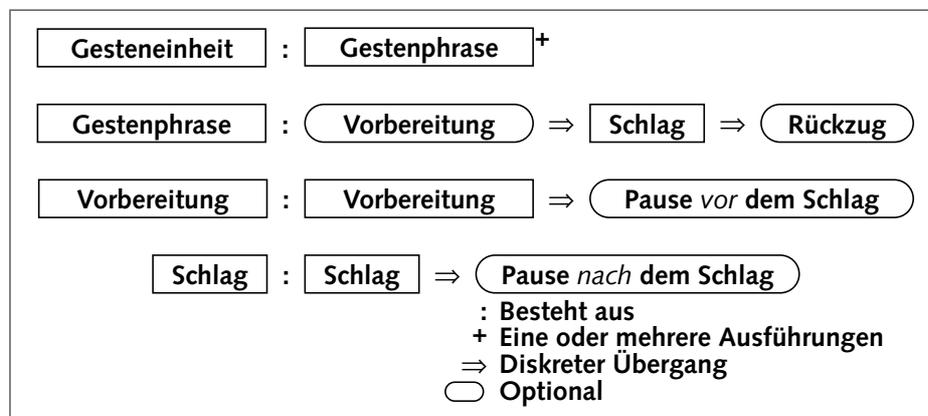


Abbildung 3: Regeln für Bewegungsphasen nach [McN92]

1. Hier und im weiteren wird Apex auch als Übersetzung des engl. peak verwendet.

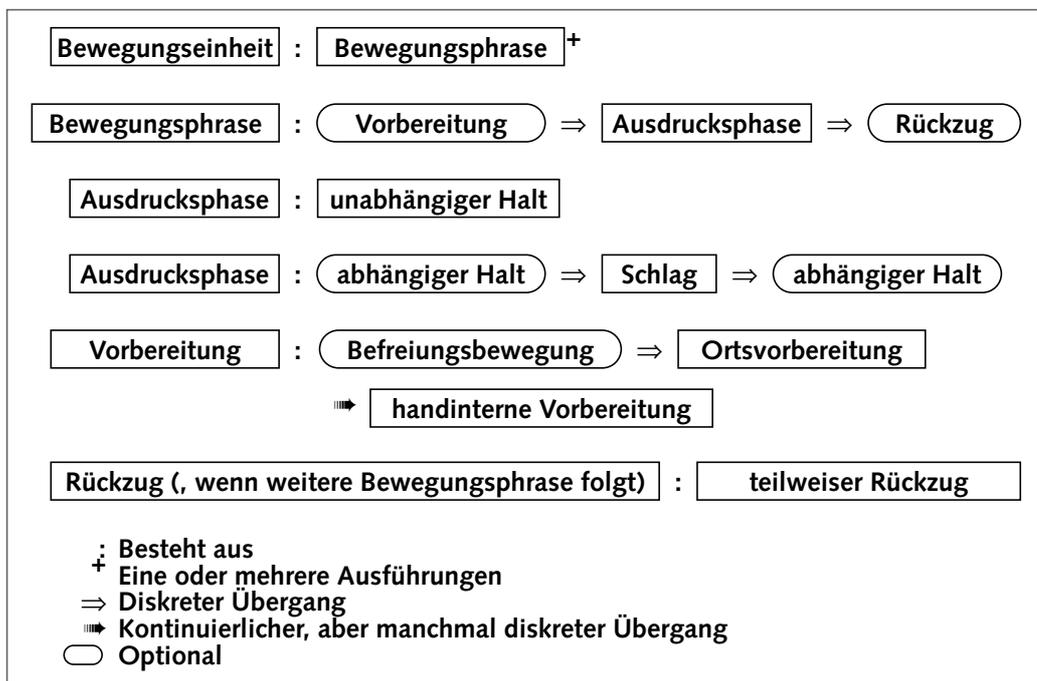


Abbildung 4: Syntagmatische Organisation von Gesten nach [KGH98]

McNeill stellte zudem fest, daß der Schlag den „conceptual focal point“ einer Äußerung hervorhebt.

Die Theorie der internen Struktur von Gesten liefert also eine Grundlage, Gesten merkmalsbasiert zu betrachten. Auch ist mit dieser Theorie die Existenz eines Apex, als Schlag oder unabhängiger Halt, begründet.

Die bislang dargestellten Theorien betrachten alleine Gesten, manchmal durchaus im Kontext mit anderen Äußerungsformen. Dabei ist die häufigste Äußerungsform, mit der Gesten kombiniert werden, die Sprache. Aus diesem Grund ist es sinnvoll, das Verhältnis von Geste und Sprache näher zu betrachten.

2. Diese Äußerung wurde des öfteren als zyklisch oder tautologisch kritisiert. Eine solche Kritik berücksichtigt hingegen nicht, daß diese Aussage ein notwendiges und nicht unbedingt ein hinreichendes Kriterium darstellt. Eine sinnvolle Interpretation dieser Äußerung ist hingegen die Erkenntnis, daß Gesten eine innere Struktur haben, die sich durch Merkmale beschreiben und somit unterscheiden läßt.

2.4 Verhalten zum Sprachkanal

Die Frage Nr. 4 vom Anfang dieses Kapitels thematisierte das Verhältnis von Gestik zum Sprachkanal. Bei der Schauspielausbildung wird der Lehrsatz verwendet: „Die Geste zieht die Sprache.“ Dies ist Ergebnis von Erfahrungswerten, die nahelegen, daß Geste und Sprache in einem gewissen temporalen Verhältnis zueinander stehen.

In [Rui98] gibt de Ruiter eine Übersicht über den Stand der Forschung der linguistischen Gestenanalyse und belegt in eigenen Experimenten Levels Beobachtungen [LRH85], daß sich die zeitliche Koordinierung der Sprache jener der Gestik anpaßt, sowie daß es *keine* vollständige Synchronisation zwischen Gestenapex und Apex der betonten Silbe in der Sprache gibt und das die Position der betonten Silbe keinen Effekt auf den Bewegungsbeginn hat. Jedoch konnte er einen Zusammenhang zwischen Gestenapex und Beginn des Nomens bei Zeigegesten nachweisen, die mit der Objektbenennung des bedeuteten Objektes einhergehen. In einem Experiment¹⁾, bei dem auf mit einem Lämpchen markiertes Bild gezeigt wurde und das abgebildete Objekt benannt wurde, ergaben sich die in Tabelle 5 dargestellten Ergebnisse einer Re-

Sprachereignis	Apex-abstand [ms]	β	s^2
<i>Beginn des Artikels</i>	-202	-0,12	0,25
<i>Beginn des Nomens</i>	-59	0,81	< 0,001
<i>Beginn der betonten Silbe</i>	71	0,07	0,11
<i>Ende der betonten Silbe</i>	360	0,04	0,25
<i>Ende des Nomens</i>	506	0,06	0,11

Tabelle 5: Apexabstände und Regressionsgewichte nach [Rui98], Seite 56

gressionsanalyse. β ist dabei der Regressionskoeffizient, ein Maß für die Stärke der Relation der Einflußgröße (hier: Sprachereignis) und der Zielgröße (hier: Gestenapex) und s^2 ist die Signifikanz gegeben durch die statistische Varianz der Einflußgröße.

1. Eine exakte Beschreibung des Versuchsaufbaues und der Versuchsbedingungen ist in [Rui98], Seite 46 ff. zu finden.

Allerdings bestätigt de Ruiter die schon von den Schauspielern (s.o.) und Gestenforschern (z.B. [McN92], [LRH85]) bekannte Regel, daß der Beginn des Gestenapex immer dem Apex der betonten Silbe in der Sprache vorausseilt und positiv mit dem Zeitpunkt dieser Silbe korreliert ist und zeigt, daß der Beginn der Bewegung auch bei einfachen Äußerungen z.B. 670 ms vor dem der Sprache liegen kann [Rui98].

Diese Erkenntnisse müssen bei einer gleichzeitigen Betrachtung beider Kanäle berücksichtigt werden und geben Aufschluß über Größenordnungen, die im Design eines technischen Systems berücksichtigt werden sollten.

2.5 Repräsentation

Da es beabsichtigt ist, Sachverhalte in dem Kontext der Gestenerkennung zu beschreiben, muß *Wissen* über Objekte und Effekte der Welt festgehalten oder dokumentiert werden, wie es konzentriert schon in Frage Nr. 5 vom Anfang des Kapitels angesprochen wurde.

Wissen wird in der Epistemologie mannigfaltig klassifiziert. Eine der Unterteilungen, die im Bereich der künstlichen Intelligenz ebenfalls gängig ist, unterscheidet zwischen *a priori* Wissen und *a posteriori* Wissen. Ersteres umfaßt universell wahres Wissen, das unabhängig von sinnlicher Wahrnehmung ist, wie mathematisch-logische Aussagen wie: „Die Winkelsumme eines ebenen Dreiecks ist 180°.“ *A posteriori* Wissen jedoch ist durch sensorische Eindrücke gewonnen, wie z.B.: „Die Ampel ist grün.“ Da sensorische Eindrücke nicht immer zuverlässig sind, trifft dies auch auf das *a posteriori* Wissen zu. So kann es unter Umständen notwendig sein, *a posteriori* Wissen auf Grund aktualisierter Eindrücke zu revidieren: „Nun ist die Ampel rot.“ Weiterhin kann Wissen in prozedurales, deklaratives oder stilles Wissen¹⁾ eingeteilt werden. Gekürzt dargestellt entspricht prozedurales Wissen Algorithmen und deklaratives Wissen Regelsätzen wie: „Fasse nicht in die Steckdose“. Stilles Wissen — auch unbewußtes Wissen genannt — umfaßt beispielsweise jene, scheinbar selbstverständlichen Dinge, wie die Fähigkeit Rad zu fahren. Die Grenze zwischen stillen Wissen und den anderen Wissensarten ist jedoch fließend, so lassen sich durchaus Regeln finden, mit denen Rad fahren erklärt werden kann.

Für die Repräsentation von stillem Wissen eignen sich z.B. künstliche neuronale Netze, da sie im Regelfall das in ihnen gespeicherte Wissen nicht direkt erklären können. Im weiteren können sie aus Beispielen lernen, kommen also auch ohne *a priori* Wissen

1. Im engl. Original: procedural, declarative and tacit knowledge

über die innere Struktur des Problems aus. Ihr größter Vorteil ist aber zugleich auch ihr größter Nachteil. Sie können zwar trainiert werden eine Aufgabe zu lösen, nachdem sie sie aber mit einer gewissen Fehlerrate erlernt haben, können sie ihre Struktur nicht erklären, noch kann Erkenntnis über die Problemstruktur aus den Gewichten der jeweiligen Neuronen abgelesen werden.

Um Wissen, insbesondere a priori Wissen, über Gesten und deren technische Verarbeitung zu beschreiben und zu diskutieren ist prozedurales oder deklaratives Wissen notwendig. Dies begründet den Vorzug eines *wissensbasierten Ansatzes* vor einem *konnektionistischen Ansatz*.

2.5.1 Interne symbolische Repräsentation

Kern des wissensbasierten Ansatzes ist die *interne symbolische Repräsentation* (ISR), die das Systemwissen darstellt und operationalisiert. Dabei müssen die Objekte und Effekte des Betrachtungsraumes in die *interne Repräsentation* überführt werden. Sei es nun, um sie aufzuschreiben oder sie in einem Rechensystem zu verarbeiten.

„Once a problem is described using an appropriate representation, the problem is almost solved.“ „The Representation Principle“ Winston [Win92], Seite 18

Eine interne symbolische Repräsentation besteht nach Winston [Win92] aus vier Teilen:

1. Einem *lexikalischen* Teil, der bestimmt, welche Symbole in dem *Vokabular* der Repräsentation erlaubt sind.
2. Einem *strukturellen* Teil, der einschränkende *Randbedingungen* beschreibt, in denen die Symbole angeordnet werden können,
3. Einem *prozeduralen* Teil, der *Zugriffsfunktionen* spezifiziert, die es erlauben, Beschreibungen zu erzeugen, zu ändern und Fragen zu beantworten, die diese Beschreibungen benutzen.
4. Einem *semantischen* Teil, der einen Weg eröffnet, *Bedeutung*, mit den Beschreibungen zu assoziieren.

SYMBOLSYSTEM. Es geht also um Symbole und ihre Beziehung zueinander und zu der Welt der Objekte und Ereignisse. Harnad faßt in [Har90] klassische Definitionen eines Symbolsystems von Newell, Pylyshyn, Fodor, Neuman, Gödel, Church und anderen zusammen. Demnach ist ein Symbolsystem eine Menge von beliebigen *physikali-*

schen Zeichen (Strichen, Bits), die auf der Basis von *expliziten Regeln*, die wiederum in physikalischen Zeichen oder *Aneinanderreihungen* von physikalischen Zeichen dargestellt sind, bearbeitet werden. Diese regelbasierte Zeichenmanipulation basiert ausschließlich auf der *Form* der Zeichen (nicht ihrer Bedeutung). Sie ist also rein *syntaktisch* und besteht aus *regelmäßiger Kombination* und Rekombination der Zeichen, die entweder *atomar* oder *zusammengesetzte* Aneinanderreihungen von physikalischen Zeichen sind. Das gesamte System, also die atomaren Zeichen, die zusammengesetzten Zeichen und die Regeln, ist *semantisch interpretierbar*. Der Syntax kann *systematisch* eine Bedeutung zugewiesen werden.

2.5.2 Schmidt'sche Hierarchie

Ein etwas formaleres und hierarchisches System stellt Schmidt in seiner Habilitationsschrift auf [Sch97]:

- Ein *Zeichen* Z ist eine Menge von *Marken*.
- Ein *Symbolschema* Σ ist eine Menge von *Zeichen*.
- Eine *Sprache* ist ein *Symbolschema*, das *syntaktisch disjunktiv* und *syntaktisch endlich differenziert* ist¹⁾.
- Ein *Symbolsystem* ist ein Tripel (Σ, R, O) eines *Symbolschemas* Σ , einer Menge von *Objekten* O und einer *Relation*²⁾ R zwischen dem *Markenvorrat*³⁾ (M) und O .
- Ein *Notationssystem* ist ein *Symbolsystem* S dessen *Symbolschema* Σ eine *Sprache* ist und das *unzweideutig*, *semantisch disjunktiv* und *semantisch endlich differenziert* ist⁴⁾.

1. syntaktisch disjunktiv: Zwei Zeichen enthalten nie die gleiche Marke.

syntaktisch endlich differenziert: Für syn. disjunktive in etwa der Entscheidbarkeitsbegriff der theoretischen Informatik für $m \in Z$ oder $m \in Z'$, Def. siehe [Sch97], Abschnitt 7.2.

2. Bei strikter Betrachtung handelt es sich nicht um eine Relation (die meist über nur *einer* Menge definiert ist), sondern um eine Korrespondenz zwischen M und O . Damit bezeichnet *Unzweideutigkeit*, daß R eine partielle Abbildung von Z nach O ist und die Bildmengen der Zeichen Z (Schmidt: Kongruenzklassen) disjunkt sind.

3. Der Markenvorrat ist die Vereinigungsmenge aller Zeichen in Σ .

4. semantisch disjunktiv: Zwei Objekte stehen nie gleichzeitig in Relation zur gleichen Marke. semantisch endlich differenziert: ähnlich wie oben, nur auf Objekte und die Relation R bezogen, Definitionen siehe [Sch97], Abschnitt 8.2.

Dabei kann die Grundeinheit „Marke“ frei gewählt werden. Eine Marke kann ein physikalisch meßbarer Effekt sein oder eine abstrakte Größe. Ob sich über einer gewählten Ansammlung von Marken die jeweils höheren Konstrukte finden lassen, und ob Konstituenten dieser Konstrukte wie die Konstrukte selbst eine hilfreiche Bedeutung haben, hängt von den gewählten Marken ab.

Im weiteren führt Schmidt aus, daß nur im Rahmen eines Notationssystems es überhaupt möglich ist, von der Gleichheit zweier Effekte zu reden. Die syntaktischen Anforderungen erlauben es, die Gleichheit oder Ungleichheit zweier Marken zu etablieren; die semantischen Anforderungen stellen sicher, daß eine Zuordnung zu Effekten prinzipiell durchgeführt werden kann und für Mengen von Effekten eindeutig ist.

Es ist festzustellen, daß Schmidts Begriff von Sprache und Notationssystem weit weniger einschränkend ist, als er normalerweise in der theoretischen Informatik aufgefaßt wird. Werden Winstons und Schmidts Äußerungen gemeinsam betrachtet, so stellt ein Schmidt'sches Notationssystem die Grundlage für den lexikalischen und semantischen Teil einer internen symbolischen Repräsentation dar.

2.5.3 Harnad'sche symbolische Repräsentation

Eine Fundierung in der menschlichen Wahrnehmung liefert Harnad [Har90]. Für ihn muß eine interne symbolische Repräsentation auf nichtsymbolischen Repräsentationen zweierlei Art fußen: Einer *ikonischen Repräsentation*, die Entsprechungen (Prototypen, Ikonen) der durch die Sensorik abgebildeten entfernten Objekte und Ereignisse enthält, sowie einer *kategorialen Repräsentation*, die Merkmalserkenner initiiert, die die invarianten Merkmale von Objekt- und Ereigniskategorien aus den sensorischen Abbildungen herauspicken.

Als Begründung dieser Zweischrittigkeit führt Harnad aus, daß in unserer unterbestimmten Welt mit ihrer Unendlichkeit an verwechselbaren potentiellen Kategorien, Entsprechungen (Prototypen, Ikonen) nutzlos für einen Identifikationsprozeß werden, da es von ihnen zu viele gibt und sie kontinuierlich von einer in die andere übergehen. Dies formt ein eigenständiges Problem der Identifizierung (Benennung), in dem entschieden wird, welche Prototypen Elemente einer Kategorie sind und welche nicht.

Werden nun Sensordaten als Schmidt'sche Marken aufgefaßt, sind diese ikonische Repräsentationen und die darauf fußenden Zeichen kategoriale Repräsentationen. Erst die semantische Zuordnung zwischen Zeichen und Bedeutung der grundlegenden Sensordaten formt dann ein Schmidt'sches Symbolsystem, das unter oben aufge-

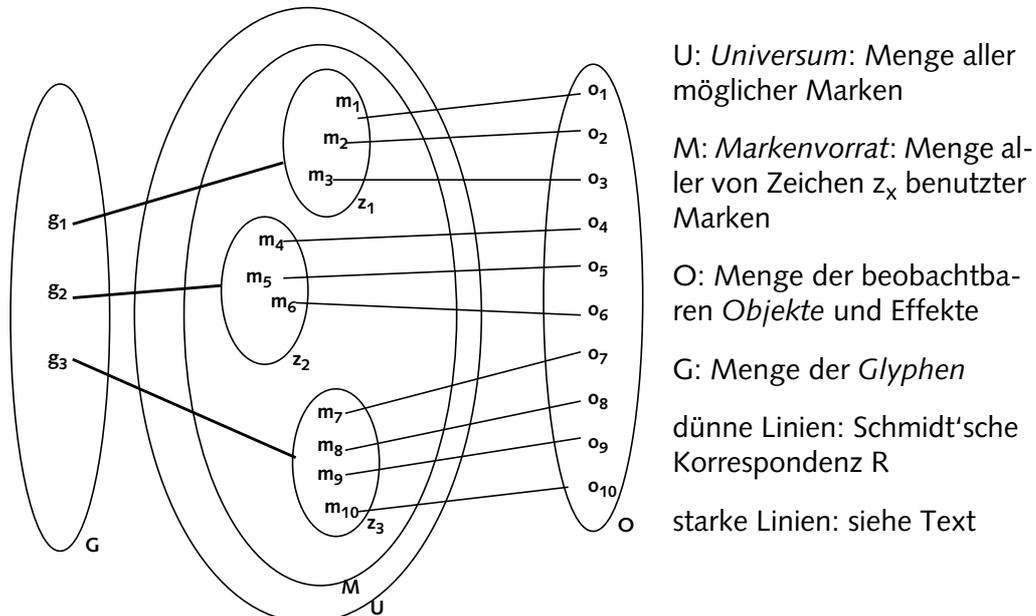


Abbildung 5: Glyphen und Notationssysteme

fürten Bedingungen ein Schmidt'sches Notationssystem und somit Grundlage einer internen symbolischen Repräsentation ist.

2.5.4 Glyphen

Es liegt nun am Standpunkt des Betrachters, ob er die niedergeschriebenen und damit physikalisch-konkreten Zeichen, Glyphen genannt, mit den Zeichen direkt identifiziert, oder aber Glyphen als eine Notation von inhärent abstrakten Zeichen versteht. Auf jeden Fall kann ohne eine Darstellung ein Zeichen nicht kommuniziert werden. Abbildung 5 soll die Thematik am Beispiel veranschaulichen. Die Frage lautet: Stellen die starken Linien eine Identität oder eine Abbildung dar? Nach Schmidt kann auch hier wieder nach den von ihm beschriebenen Strukturen gesucht werden, da er eine bewußt wahrgenommene, relevante Information als auf vielen geschachtelten Symbol- bzw. Notationssystemen beruhend ansieht.

Diese Ausführungen legen das Fundament für die im Abschnitt „Gestennotationssysteme“ auf Seite 37 behandelten Systeme und legen dar, welche Anforderungen an solche Systeme zu stellen sind. Neben diesen formalen Gesichtspunkten, gibt es aber noch welche, die sich mehr mit der Verständlichkeit solcher Notationssysteme für den Menschen beschäftigen.

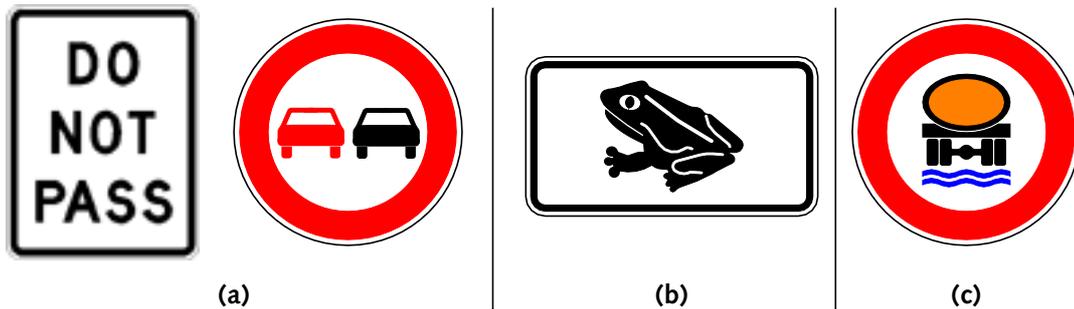


Abbildung 6: Verkehrszeichen

„**READILY INFERABLE MEANING**“ UND „**RICHLY GROUNDED SYMBOLS**“. Wie schon ausgeführt wurde, basieren Symbolsysteme auf physikalischen Zeichen, den Glyphen. In dem Fall, in dem diese Zeichen für einen Menschen wahrnehmbar sind, treten zwei Effekte auf, zum einen der von „Readily Inferable Meaning“ (RIM) und zum anderen der der „Richly Grounded Symbols“ (RGS) (siehe [Edw98]). Beide sind stark miteinander verwoben. Letztendlich bezeichnet die RIM jene Eigenschaft, die die Fundierung eines Glyphen reichhaltig macht. Um zu verstehen, was diese Begriffe bedeuten, seien das amerikanische und europäische Verkehrszeichen für „Überholverbot“ betrachtet (vgl. Abbildung 6 (a)). Das amerikanische Verkehrszeichen (links) ist rein symbolisch, seine Bedeutung ist beliebig aber konventionalisiert. Ohne die Konvention, hier die englische Sprache, könnte es jede Bedeutung haben. Doch das rechts daneben abgebildete europäische Verkehrszeichen hat als Glyphenbedeutung ebenso „Überholverbot“, doch trägt es auch zusätzliche Information, die wir aus bereits gemachten Erfahrungen ableiten. Die parallele Anordnung der autoförmigen Umrisse gibt einen Hinweis darauf, daß dieses Zeichen etwas über parallelfahrende Autos aussagt, der starke rote Rahmen wird im allgemeinen als Verbotssignal aufgefaßt, und die rote Farbe in dem linken Umriß legt nahe, daß es verboten ist, ein Auto links eines anderen anzuordnen. Damit ist die Glyphenbedeutung dem Sinn nach aus dem Alltagswissen und dem physikalischem Erscheinungsbild des Glyphen inferiert.

Ohne die definierte Bedeutung der Glyphen (b) und (c) in Abbildung 6 zu kennen, kann doch gefolgert werden, daß (b) etwas bezeichnet, das mit Kröten oder Fröschen zu tun hat, und daß (c) ein Verbot darstellt, das die Fracht von Tanklastzügen in Verbindung mit Wasser betrifft. Menschen versuchen also die Bedeutung eines Glyphen aus seiner physikalischen Form herzuleiten. Wenn dieses Phänomen sinnvoll ausgenutzt wird, kann es dazu beitragen, Symbolsysteme zu entwerfen, die nicht nur von

formal wohlgeformter Struktur, sondern auch für den Betrachter leicht verständlich sind.

2.6 Zusammenfassung

Der Begriff „Geste“ ist weit gefaßt und hat unscharfe Abgrenzungen. Daher wurde neben einer Arbeitsdefinition ein Beschreibungsschema — eine Taxonomie — für Gesten vorgestellt. Um die Taxa sinnvoll zu belegen, wurden Einteilungen von Gesten aus der Literatur nach dem Betrachtungspunkt von Produzenten und Rezipienten vorgestellt sowie die innere Struktur dieser Gesten betrachtet und ihr Verhalten zu einem weiteren Kanal — dem Sprachkanal — dargelegt.

Um diese Erkenntnisse nutzbar zu machen, wird eine interne symbolische Repräsentation benötigt. Um ihre Anforderungen zu charakterisieren sind Überlegungen zu Symbolsystemen und darauf aufbauenden Strukturen herangezogen worden. Weiterhin wird dargelegt, daß Notationssysteme nicht per se symbolische Repräsentationen sind, aber diese auf Notationssystemen, die auch in sich geschachtelt sein können, basieren. Im weiteren wurde dargestellt, das Unterschiede zwischen einem Objekt/Effekt, dem Zeichen und dessen Glyphen bestehen. Dabei werden Argumente aufgeführt die die Sicht motivieren, eine Menge von Objekten/Effekten mittels Merkmalen zu beschreiben und mittels eines Zeichens zu repräsentieren um dann darauf weitere Systeme aufzubauen, die z.B. das Zusammenspiel dieser Zeichen beschreiben. Die so verwendeten Zeichen sind durch diese Objekte/Effekte fundiert.

Durch diese Ausführungen sind die ersten fünf Fragen vom Anfang dieses Kapitels behandelt. Die Fragstellungen nach Zeit und Notation werden in den beiden folgenden Kapiteln thematisiert.

Kapitel 3

Zeitbegriffe

In den vorhergehenden Abschnitten wurden Gesten im Prinzip stationär betrachtet. Wie aber der Abschnitt „Innere Struktur von Gesten“ auf Seite 21 schon deutlich macht und in Frage Nr. 6, auf Seite 11 schon angesprochen wurde, sind Gesten ein Formphänomen in der Zeit. Dabei stellt sich „Zeit“ als ein komplexeres Phänomen dar, als es in naiver Betrachtung erscheint. Nicht nur die Zeitbegriffe der Physik, wie z.B. Einsteins „Raum-Zeit“ Begriff, sondern auch Zeit als Wahrnehmungsphänomen weichen von dem naiven Bild des linearen, streng monotonen, stetigen Flusses ab. Daher modellieren moderne Ansätze Zeit meistens über Intervalle oder Zeitpunkte und deren Relationen zueinander [HHP93].

Da es im Kontext dieser Arbeit aber um ein Wahrnehmungsphänomen geht, wird im folgenden auf einer Theorie zur zeitlichen Kognition von Pöppel und Koautoren [PS95], [PS93] aufgebaut, die hier kurz vorgestellt und eingeordnet sei.

Menschliches Zeiterleben läßt sich durch eine Zusammenstellung von fünf verschiedenen, elementaren Zeiterlebnissen beschreiben: *Gleichzeitigkeit*, *Ungleichzeitigkeit*, *Aufeinanderfolge*, *Gegenwart* und (Fort-) *Dauer*. Dies wird durch folgende Beobachtungen belegt:

3.1 Gleichzeitigkeit – Ungleichzeitigkeit

Wenn über einen Kopfhörer in beide Ohren kurze Reize gegeben werden, die etwa eine Millisekunde andauern, und diese beiden Reize gleichzeitig gegeben werden, so erlebt die Versuchsperson einen einzigen Ton, und zwar in der Mitte des Kopfes. Wird zwischen diesen Reizen eine zeitliche Verzögerung, z. B. von zwei Millisekunden, eingeschaltet, so erlebt die Versuchsperson ebenfalls nur einen Ton. Das heißt, die zwei Reize werden in der Wahrnehmung miteinander verschmolzen, obwohl objektiv betrachtet die beiden Reize ungleichzeitig sind. Erst dann, wenn die zeitliche Differenz zwischen den beiden akustischen Reizen etwa drei bis fünf Millisekunden beträgt, ist die Schwelle zur Ungleichzeitigkeit erreicht. Wird der analoge Versuch im visuellen System durchgeführt, dann ist feststellbar, daß die Verschmelzungsgrenze dort etwa bei 20 bis 30 Millisekunden liegt.

Die Unterscheidung von Gleich- und Ungleichzeitigkeit erfolgt also in jedem Sinnesbereich in unterschiedlichen Zeiträumen. Im weiteren bleibt festzuhalten, daß Gleichzeitigkeit im Erleben nichts Absolutes ist. Je nach Sinnesmodalität und weiterhin beeinflußt vom sonstigen Befinden und Zustand ist das *Gleichzeitigkeitsfenster* verschieden.

3.2 Ordnungsschwelle

Nun werden Versuchspersonen nicht mehr nur nach der Gleichzeitigkeit zweier Reize gefragt, sondern welches der *erste* und welches der *zweite* Reiz war. Durch die neue Frage wird die Aufmerksamkeit auf einen anderen Aspekt der Reizeanordnung gelenkt. Dabei zeigt sich, daß die Änderung der Frage zu einem anderen Ergebnis führt: Während die Schwelle zur Ungleichzeitigkeit beim Hören bei wenigen Millisekunden liegt, ist für das Erkennen der zeitlichen Ordnung bei identischen Reizbedingungen zu beobachten, daß dieser Wert bei etwa 30 bis 40 Millisekunden liegt. Diese Schwelle wird Ordnungsschwelle genannt. Es ist auffällig, daß die Ordnungsschwelle in den verschiedenen Sinnesbereichen alle gleich sind, während der Übergang von der Gleich- zur Ungleichzeitigkeit in den einzelnen Sinnesbereichen verschieden ist.

OSZILLATORISCHE NEURONALE PROZESSE. Pöppel und Mitarbeiter [PS93] [SZ95] führen diese Phänomene auf oszillatorische Nervenprozesse zurück, die eine getaktete innere Uhr zur Folge haben. Unter anderem wird mit der Wirkungsweise von Anästhetika argumentiert, die diese Oszillation unterbrechen und so jede zeitliche Wahrnehmung unterbinden. Tatsächlich fragen Patienten häufig *nach* einer Operation unter Vollnarkose, wann denn die Operation beginne, was eine völlige temporale Disorientierung des Patienten belegt.

SUBJEKTIVE GEGENWART. Pöppel betrachtet noch einen weiteren subjektiven Mechanismus des menschlichen Zeiterlebens. Er stellt fest, daß Ereignisse nicht für sich allein stehend wahrgenommen werden, sondern daß einzelne Ereignisse aufeinander bezogen werden und daß aufeinanderfolgende Ereignisse jeweils eine sog. *Wahrnehmungsgestalt*, eine in sich geschlossene Interpretation, haben. Dies ist nur dadurch möglich, daß das Gehirn einen zeitlichen Integrationsmechanismus bereitstellt, der dafür sorgt, daß solche Wahrnehmungsgestalten gebildet werden.

3.3 Drei-Sekunden-Phänomen

Eine Wahrnehmungsgestalt kann vom Gehirn nur etwa drei Sekunden lang festgehalten werden. Nach dieser Zeit ist die Integrationsfähigkeit des Gehirns gewissermaßen

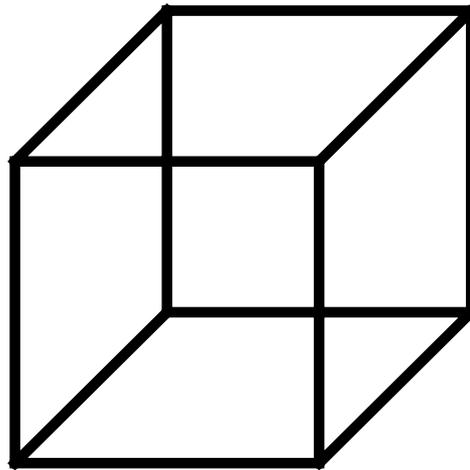


Abbildung 7: Der Necker-Würfel

„erschöpft“. Bietet die Reizkonfiguration die Möglichkeit, in mehreren Weisen interpretiert zu werden, dann kommt automatisch nach etwa drei Sekunden die jeweils andere Wahrnehmungsweise zur Geltung.

Zur Veranschaulichung führt Pöppel einige im Selbstversuch einfach zu beobachtende Phänomene auf:

- Wird ein Metronom betrachtet, das im Sekundentakt schlägt (oder eine langsam tickende Uhr) so kommt nach einer kurzen Zeit eine subjektive Akzentuierung in den Sinn: Etwa jedem zweiten Metronomschlag wird ein subjektiver Akzent gegeben. Es drängt sich das Gefühl auf, er sei ein wenig lauter als der subjektiv nichtakzentuierte Schlag (Tick-Tack Effekt). Es gelingt manchen Menschen auch noch, dem jeweils dritten Schlag subjektiv zu einer Wahrnehmungsgestalt zusammenzufassen, den wenigsten aber noch, dies für den vierten Schlag oder gar weitere Schläge zu erreichen.
- Ein Getränk, gemischt aus gleichen Teilen Kirsch- und Bananensaft, heißt auf den Karten einiger Gaststätten „KiBa“ bei anderen „BaKi“. Wenn nun eine gleichförmige, regelmäßige Buchstabenfolge „...KIBAKIBAKIBAKIBAKIBA...“ betrachtet (noch besser: gehört) wird, wird entweder „KiBa“ oder „BaKi“ produziert (bzw. gehört). Wird versucht, beides zu sprechen, so wechseln sich beide Sprechweisen nach typischerweise drei Sekunden ab (dito für Hörweisen).
- Zuletzt ein graphisches Beispiel: Der Necker-Würfel, wie in Abbildung 7 zu

sehen. Wird das innere Quadrat betrachtet, so sind die untere linke und die obere rechte Ecke ausgezeichnet, da von diesen Punkten drei Linien entspringen. Das Gehirn interpretiert dieses zweidimensionale Linienmuster aber als eine dreidimensionale Wahrnehmungsgestalt, bei der entweder die eine oder die andere Ecke hinten bzw. vorne liegt. Der Betrachter (o.B.d.A. auch „die Betrachterin“) kann nun an sich selbst feststellen, daß die Interpretationsvariante nach ca. drei Sekunden umspringt.

Empirisch läßt sich dieses Phänomen durch Reproduktionsexperimente belegen. In Abbildung 8 ist das Ergebnis eines solchen Experimentes dargestellt. Hier wurden Reizstrecken im 0,5 s-Abstand von 0,5 s bis 7 s präsentiert und daraufhin reproduziert. Neben der theoretischen $S = R$ -Geraden („Stimulus = Response“), die die identische Abbildung darstellt, ist auch das geometrische Mittel aller Reizstrecken \bar{d} aufgetragen. Es fällt auf, daß bis 3 s die Antworten die Reizstrecke leicht überschätzen, darüber aber die Antworten relativ immer kürzer ausfallen. Die Reproduktion von Zeitintervallen bis etwa 3 s kennzeichnet also einen begrenzten Integrationsmechanismus des Gehirns.

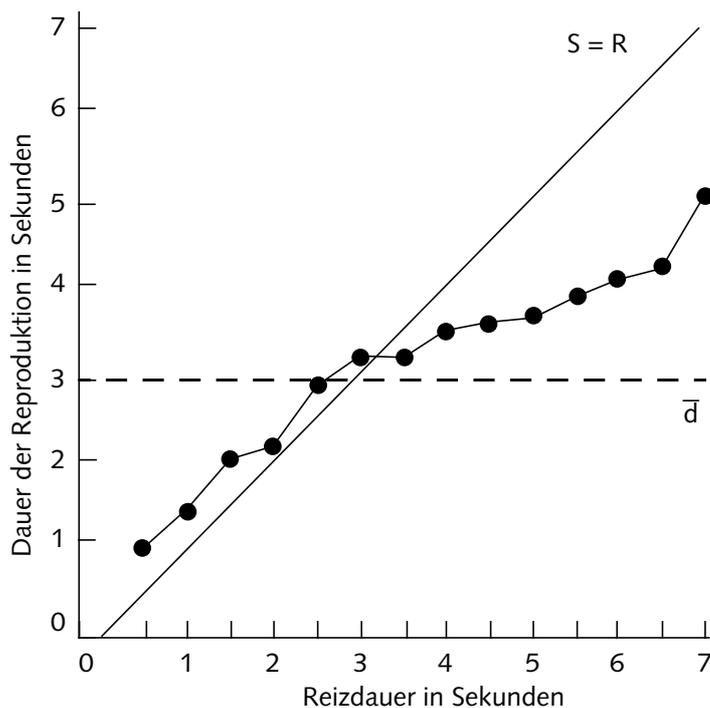


Abbildung 8: Reproduktion von Zeitstrecken nach [Poe97]

3.3.1 Zeitliche Kontinuität

Trotz der oben vorgestellten zeitlichen Segmentierung haben wir Menschen ein kontinuierliches Zeiterleben. Dies erhalten wir durch einen weiteren Mechanismus unseres Gehirns [Poe97]. Schon was jeweils in das Bewußtsein gelangt, ist nicht unabhängig von den vorangegangenen Bewußtseinsinhalten. Über eine solche inhaltliche Vernetzung aufeinanderfolgender Bewußtseinsentitäten ergibt sich der subjektive Eindruck einer zeitlichen Kontinuität. Hier liegt eine aktive Leistung des Gehirns vor. Die kann belegt werden, wenn Fälle betrachtet werden, bei denen dieser Mechanismus gestört ist. Ein schizophrener Patient ist im Extremfall nicht mehr in der Lage, aufeinanderfolgende Bewußtseinsinhalte so aufeinander zu beziehen, daß die Bedeutung eine sinnvolle Gedankenkette ergibt. Für einen solchen Patienten gehen die Kontinuität des Erlebens und der subjektive Eindruck eines zeitlichen Stromes verloren.

3.4 Zeitbänder

Auch andere Quellen berichten von ähnlichen Modellbildungen für die menschliche Zeitverarbeitung. So schreibt Ballard in [Bal97] in Anlehnung an [New90] von kognitiven Hierarchien, die zusammen mit ihren temporalen Randbedingungen in der folgenden Tabelle 6 als sog. Zeitbänder dargestellt sind.

Zeitband	Primitive	Beispiel
10 s	Komplexe Aufgabe	Ein Zug im Zeit-Schach
2 s	Einfache Aufgabe	Einen Satz ^a äußern
300 ms	Physikalischer Akt	Bewegen der Augen
50 ms	Neuronaler Akt	Einen Reiz bemerken
10 ms	Neuraler Schaltvorgang	
1 ms	Neuronale Entladung	

Tabelle 6: Zeitbänder der menschlichen Kognition nach [New90]

a. In dem Kontext ist ein kurzer englischer Satz gemeint.

Auffällig ist hier, daß von einer anderen Art neuronaler Kernereignisse ausgegangen wird. Nicht von oszillatorischen Prozessen, wie bei Pöppel, sondern von einfachen

Entladungen, lokalen Schaltvorgängen (die der Einfachheit halber als 10-stufig angenommen werden¹⁾), einer Signallaufzeit sowie simplen motorischen Vorgängen. Ballard betont für die Diskussion dieser Hierarchien zwei Punkte:

„The first is that the abstract analysis of hierarchical systems finds a reality in the construction of the brain. [...] The second point is that the constraints that the individual levels impose on models of computation are quite severe.“
[Bal97], Seite 17 Mitte

Ballard folgert daraus fünf grundlegende, mathematische und rechnerische Konzepte: (1) „fitness“, (2) „programs“, (3) „data“, (4) „dynamics“ und (5) „optimization“. Hier soll nur Ballards „dynamics“ kurz näher betrachtet werden: Der Zustand eines physikalischen Systems hat eine ihm zugeordnete Dynamik. Das ist die zeitliche Änderung des Zustandsvektors, der so eine Trajektorie beschreibt, die am besten differential durch die Änderungsrate in der Zeit beschrieben wird. Die am einfachsten zu behandelnden Systeme sind dabei die linearen. Solche Systeme sind vollständig durch ihre Eigenwerte und Anfangsbedingungen beschrieben. Nichtlineare Systeme können durch lineare Systeme approximiert werden, die lokal zu Punkten stabilen Gleichgewichtes (sog. Attraktoren) sind.

3.5 Zusammenfassung

Wenn man von leicht verschiedenen Betrachtungsweisen absieht, beschreiben Pöppel und Ballard doch das gleiche Grundphänomen. Schließlich sind oszillatorische Prozesse (Pöppel) auch (nichtlineare) dynamische Systeme (Ballard). Beide sprechen von einer Hierarchie der temporalen Prozesse mit verschiedenen Ausprägungen von zeitlichen Intervallen, die summarisch zu Entitäten in höheren Schichten zusammengefaßt werden können.

-
1. Diese Annahme ist mit erhöhter Vorsicht zu genießen: Wie neuere Ergebnisse der Neuroanatomie zeigen, ist die Anzahl der Neuronen und ihr Vernetztheitsgrad sehr viel höher als bisher angenommen. So führt Pöppel in seinem Hauptvortrag auf der Jahrestagung für Künstliche Intelligenz in Bremen, 1998 aus, daß im Gehirn wohl in der Größenordnung von 10^{23} Neuronen vorhanden sind, die eine durchschnittliche (Verbindungs-) Weglänge von nur vier (4) haben. Ein zehnstufiger Schaltvorgang könnte also mehr als „einmal durch das ganze Gehirn und zurück“ bedeuten. Ballard hingegen geht von 10^{11} Neuronen im Gehirn aus ([Bal97], Seite 31, zweiter Absatz)

Kapitel 4

Gestennotationssysteme

Um Gesten in einer informatischen internen symbolischen Repräsentation adäquat zu behandeln, ist es sinnvoll, neben den theoretischen Anforderungen auch existierende Notationen für Gesten zu betrachten. Diese sind zwar für einen menschlichen Betrachter entwickelt und dienen meist der linguistischen Analyse einer Gesamtäußerung, doch sind die für diesen Fall positiven Merkmale einer Gestennotation nicht notwendigerweise schädlich für eine interne symbolische Repräsentation. Diese Betrachtungen dienen dem Zweck, neben dem Vermitteln einer Übersicht, Material zu präsentieren auf das die Konzeption und Realisation des Gestenerkennungsrahmens sich bezieht. An die einzelnen Betrachtungen schließt sich eine Diskussion und Zusammenfassung an, in der das jeweilige System in Hinblick auf die bereits besprochenen theoretischen Eigenschaften aber auch in Hinblick auf hervorstechende Merkmale hin besprochen wird.

Zuerst einmal sind jedoch die Begriffe „Notation“ und „Notationssystem“ zu klären:

Definition: Notation: Im allgemeinen eine Form der Aufzeichnung oder ein System von Zeichen, wie z.B. die Notenschrift in der Musik (-wissenschaft).

Definition: Notationssystem: Ein Fachbegriff (siehe z.B. auch „Schmidt’sche Hierarchie“ auf Seite 26), der den systematischen Charakter einer Notation betont. Dieser Begriff wird auch benötigt, um zwischen dem Formalismus und der Handlung zu unterscheiden. Wie bisher wird auch im folgenden der Begriff „Notationssystem“ synonym zu „Notation“ verwendet; es sei denn, ein spezielles, extra definiertes Notationssystem ist gemeint. Dann wird der Begriff „Notationssystem“ durch einen den Kontext festlegenden Bezeichner qualifiziert, wie z.B. „Harnard’sches Notationssystem“.

Das etablierte Verfahren zur Notation extralinguistischer Kanäle ist die *Transskription*. Dabei werden in einer Partitur ähnlichen Einteilung in verschiedenen Zeilen für ausgewählte Merkmalsgruppen die einzelnen zu transkribierenden Merkmale verschriftlicht. Dabei sind in den letzten Jahren halbautomatische Hilfssysteme für Rechenanlagen auf dem Markt, die es erlauben, digitalisierte Video- oder Audiodaten bequemer zu handhaben. Eine der wesentlichen Erleichterungen ist dabei sicherlich

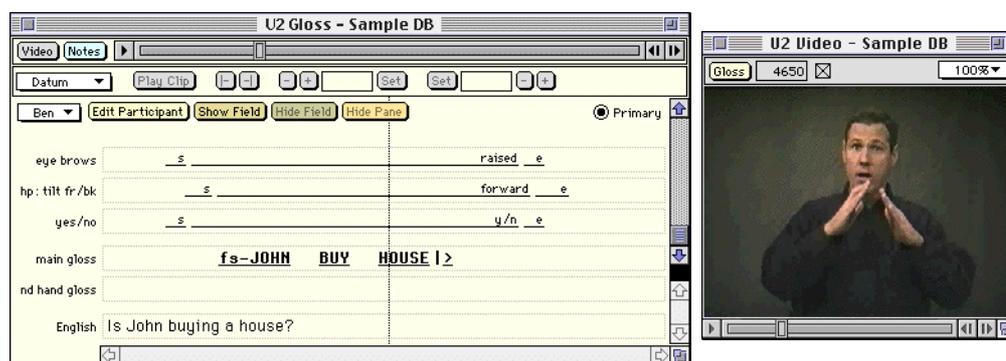


Abbildung 9: SignStream: Eingabe- & Videofenster

die Abkopplung vom trägen, physikalischen Band- oder Plattenmedium. Auch die einzel- oder halbbildgenaue¹⁾ Positionierung der Transkriptionsmarken kann sehr hilfreich sein, insbesondere wenn zeitliche Auswertungen erwünscht sind.

Beispiele für solche Systeme sind z.B. SyncWriter²⁾ und SignStream³⁾. Ersteres ist ein allgemeines Transkriptionswerkzeug, siehe auch Abbildung 10 auf Seite 39, während letzteres in einem speziellen Forschungskontext [NLB+97] für die Transkription in amerikanischer Gebärdensprache „American Sign Language“ (ASL) entwickelt wurde, siehe auch Abbildung 9.

Diese Transkriptionssysteme lösen aber nicht das Problem einer adäquaten Notation. Dies Problem enthält im Kern die Wahl der richtigen Menge von Merkmalen für ein sinnvoll geformtes System von Zeichen, ihren Anordnungen und darauf definierten

1. Einige Videoverfahren, insbesondere das „Phase Alternating Line“ (PAL), das Ende der 1950er Jahre von W. Bruch entwickelt wurde und das am Ende jeder Zeile die Phase des Farbdifferenzsignals um 180° ändert (umpolt), benutzt 50 Halbbilder pro Sekunde (einmal alle ungerade Zeilen, dann alle geraden Zeilen). Dabei handelt es sich um zeitlich unabhängige Bilder und nicht etwa um die nachträgliche Codierung eines Vollbildes.
2. SyncWriter™: Hersteller: med-i-bit GmbH in Kooperation mit dem Zentrum für Deutsche Gebärdensprache der Universität Hamburg. Autor: Erich Walter. Abbildungen aus Hersteller-Web-Seiten entnommen („Unified Resource Locator“ (URL): <http://www.sign-lang.uni-hamburg.de/Artikel/1/Muenchen.html>).
3. SignStream™: Hersteller: American Sign Language Linguistic Research Project: Boston University, Gallaudet University, Dartmouth College, und Rutgers University, USA. Autoren: Boston University: Carol Neidle, Dawn MacLaughlin, Jack Hoza, Robert G. Lee; Dartmouth College: Otmar Foelsche, David Greenfield; Gallaudet University: Benjamin Bahan; Rutgers University: Judy Kegl. Abbildungen von Web-Seiten des Herstellers (URL: <http://www.bu.edu/asllrp/SignStream/>).

The screenshot shows two windows from the SyncWriter software. The left window, titled 'Sample U (Band)', displays a timeline with timecodes (09:12, 09:39:18, 09:40:18) and a grid of video frames. Below the timeline is a table with columns for different features: Mimik, Glosse, Gloss / Engl., Mundbild, HamNoSys, Übersetzung, and English transl. The right window, titled 'Sample U (Partitur)', shows a detailed score for 'Sample V' with two sections. Section 1 includes timecodes (09:38:15, 09:38:19, 09:39:06) and a table with columns for Mimik, Glosse, Gloss / Engl., Mundbild, HamNoSys, Übersetzung, and English transl. Section 2 includes timecodes (09:39:12, 09:39:18) and a similar table. Both windows include a 'Play Scene' button.

links oben: „Endloses“ Eingabeband mit verschiedenen Merkmalsgruppen

rechts unten: Generierte Partitur für den Ausdruck

Abbildung 10: SyncWriter

Funktionen. Im weiteren stellt sich auch die Frage, inwieweit ein eigenes System von *Glyphen* (graphischen Zeichen) für diese Symbole sinnvoll sein kann.

In den vorhergehenden Abschnitten wurde dargelegt, welche theoretischen Anforderungen an solche Notationen gestellt werden: Das Wesentliche muß unterscheidbar sein, die Zeichen müssen eindeutig sein, die Glyphen sollten verständlich sein. Auch sollte die innere Struktur kompositionell sein (Siehe „Schmidt'sche Hierarchie“ auf Seite 26).

4.1 Labanotation (Kinetographie Laban)

Als erstes soll hier die *Labanotation*, eine Transkription von Bewegungen, insbesondere des Tanzes, vorgestellt werden. Anhand eines kurzen Beispielen soll dann ein Eindruck von der Mächtigkeit einerseits und der Aufwendigkeit der Notation andererseits gegeben werden. Anschließend erfolgt eine Diskussion anhand der oben dargelegten Punkte.

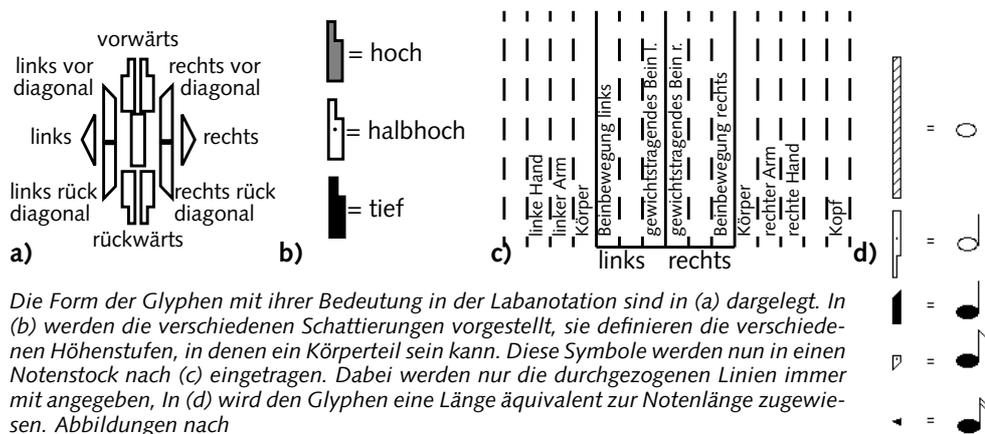


Abbildung 11: Labanotation: Zeichenerklärungen

Das nach Rudolf von Laban benannte kinetographische System *Labanotation* ist eine Bewegungs- und Tanzschrift [Lab95], die von von Laban 1928 zum ersten Mal publiziert wurde und nun in verschiedenen Variationen und Überarbeitungen vorliegt. Ihr Aufbau entspricht einem senkrechten Spurensystem, daß vorwärts, also von unten nach oben gelesen wird. Dabei werden musikalische Merkmale, wie Taktmaß und Takteinteilung sowie Zeitmaße im Taktsystem notiert, wie auch die Bewegungsart und -richtung einzelner Körperteile. Dabei wird aber immer Bezug auf das zugrundeliegende Taktmaß genommen. So findet man die Äquivalente zu den verschiedenen Notendauern (ganze Note, halbe Note, viertel Note usw.) auch in dieser Tanzschrift wieder (siehe Abbildung 11). In Abbildung 12 ist ein Beispiel dargestellt, das im weiteren etwas genauer erklärt wird:

GRUNDPOSITION. Die Zahlen zur Linken geben das Taktmaß an: $\frac{3}{4}$ Takt. Begonnen wird mit geraden Knien und zusammenstehenden Füßen. Die Arme sind zur Seite in einem 45° Winkel abwärtsgerichtet abgespreizt.

ERSTER TAKT. Es ist weiterhin von unten zu lesen! Auf dem ersten Schlag einen Schritt vorwärts auf den linken Fuß; das Knie bleibt gerade. Gleichzeitig werden die Arme nach vorne auf Schulterhöhe gehoben, bis sie parallel zum Boden sind. Mit dem zweiten Schlag wird mit dem rechten Fuß ein Schritt nach rechts gemacht, und das Gewicht vom linken Fuß genommen. Die Arme öffnen sich zur Seite während sie parallel zum Boden bleiben. Beim dritten Schlag wird der linke Fuß zum rechten gesetzt und das Gewicht auf beide verteilt. Die Arme an den Seiten senken sich auf 45° .

ZWEITER TAKT. Beim ersten Schlag, einen Schritt vorwärts mit dem rechten Fuß. Gleichzeitig heben sich die Arme vorwärts bis in Schulterhöhe in der zweiten Hälfte des Schlags bis auf einen 45° Winkel nach oben. Mit dem zweiten Schlag einen Schritt mit dem linken Fuß nach links und die Arme öffnen sich zur Seite, parallel zum Boden. Auf dem dritten Schlag einen Schritt nach rechts mit dem rechten Fuß; das Gewicht verteilt sich auf beide Füße. Die Arme bleiben an den Seiten aber senken sich auf 45° ab.

Erster und zweiter Takt enthalten den sog. Kastenschritt¹⁾.

DRITTER TAKT. Mit dem ersten Schlag werden zwei Schritte vorwärts mit leicht gebeugten Knien (links-rechts) vollführt. Jeder Schritt dauert $\frac{1}{2}$ Zähler. Die Arme nehmen sich den ganzen Schlag Zeit, um nach vorne parallel zum Boden gehoben zu werden. Auf den Schlägen zwei und drei wird ein langsamer Schritt vorwärts auf den linken Fuß durchgeführt. Gleichzeitig öffnen sich die Arme zu den vorderen Diagonalen, bleiben aber nach wie vor parallel zum Boden (siehe Abbildung 11-a).

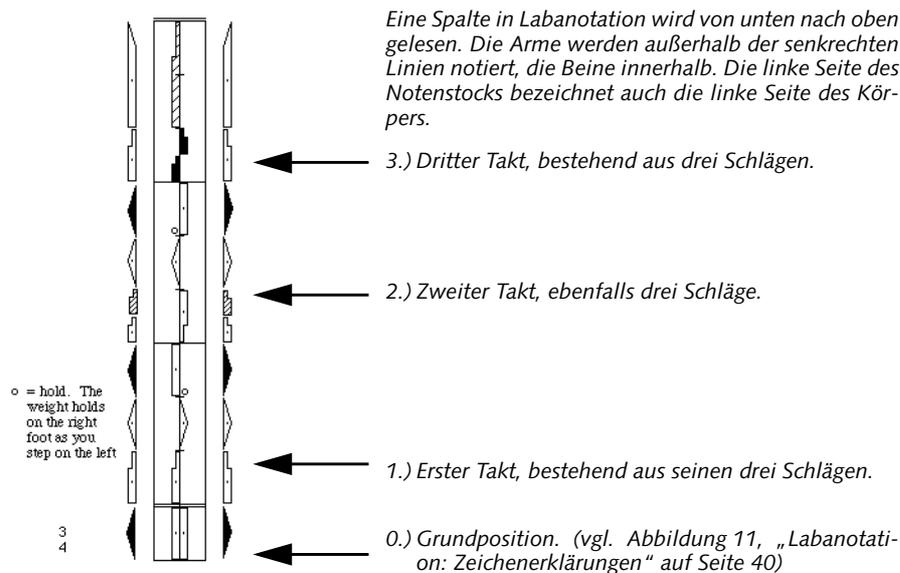


Abbildung 12: Labanotation: Einführungsbeispiel

1. In „Tanzlehreranweisungen“: Links vor – Schritt nach rechts mit rechtem Bein – schließen
– Rechts zurück – Schritt nach links mit linkem Bein – schließen.

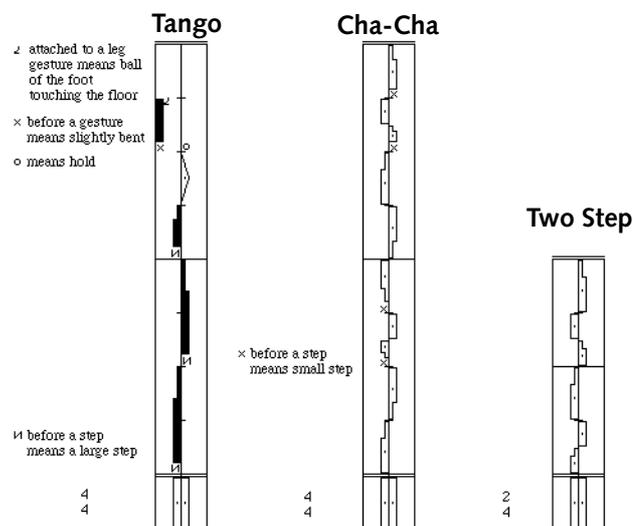


Abbildung 13: Labanotation: bekannte Tanzschritte

Abschließend sei hier noch die Labanotation einiger bekannter Tanzschritte in Abbildung 13¹⁾ aufgeführt.

4.1.1 Diskussion

Wie Schmidt ([Sch97], Abschnitt 8.2) darlegt, ist die Labanotation ein Schmidt'sches Notationssystem und erfüllt somit schon einmal wichtige Voraussetzungen für den technischen Einsatz, nämlich die Abzählbarkeit der Symbole und ihre definierte Verankerung in Objekten/Effekten. Die wenigsten Zeichen dieses Systems jedoch haben eine RIM (vgl. „„Readily Inferable Meaning““ und „„Richly Grounded Symbols“““ auf Seite 29), was dazu führt, daß das Erlernen dieses Systems wahrscheinlich schwierig ist. Um dies zu belegen wurde, eine kleine Studie angefertigt, in welchem Maße die Labanotation von Tänzern, also in ihrer Zieldomäne, verwendet wird.

EINSATZHÄUFIGKEIT IN DER ZIELANWENDUNG. Befragungen von zwölf (12) professionellen Tänzern internationaler Herkunft des Bielefelder städtischen Ballettensembles²⁾ ergaben die im folgenden dargestellten Antworten auf die jeweiligen Fragen:

1. Siehe <http://www.dancenotation.org/notationbasics/LNBasics.html>
2. Vielen Dank an Micki und Bernd Volkmer, die die Befragung ihrer Kollegen durchführten.

- „Haben Sie schon einmal von der Tanzschrift ‚Labanotation‘ gehört?“ Darauf antworteten sechs (6) mit „Ja“.
- „Haben sie die Labanotation schon einmal angewendet?“ Darauf antworteten alle mit „Nein“.
- „Könnten Sie die Labanotation in der Zukunft anwenden?“ Auch darauf antworteten alle mit „Nein“.

Dem Fragesteller fiel auf, daß die Befragten sich auf die Probe gestellt fühlten und ihren professionellen Ehrgeiz herausgefordert sahen, daher sind die Antworten zur ersten Frage als eher zu hoch- als zu niedrig gegriffen anzusehen. Diese kleine Studie belegt, wenngleich sie auf Grund der fehlenden Repräsentativität der Stichprobe auch keinen Beweis darstellt, daß dies Notationssystem heutzutage in seiner Domäne nur geringe Verbreitung hat.

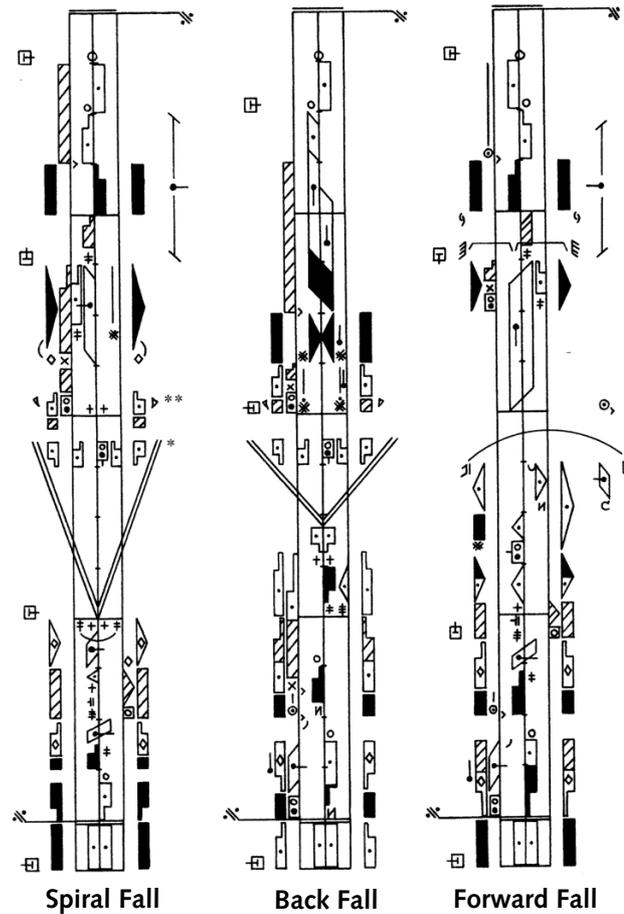
Die Labanotation ist als Tanzschrift auf ihr Einsatzgebiet zugeschnitten und intendiert keine allgemeine Notation von Gesten. Soll sie trotzdem als allgemeine Notation der Bewegungen der oberen Gliedmaßen zweckentfremdet verwendet werden, so ist es absehbar, daß besondere Aufmerksamkeit der relativen Zeitskala (musikalischer Takt) zufließen muß. Es ist noch zu zeigen, ob Gesten intrinsisch getaktet geäußert werden und daher in ein solches Zeitschema einordbar sind. Im weiteren ist die Inflation an Glyphen beklagenswert; in einem Werk zur Labanotation [Hut77] besteht das sog. Symbolglossar aus elf eng bedruckten Seiten mit weit über 300 Glyphen (eigene Zählung). Welche Folgen dies hat, mag das Beispiel in Abbildung 14, „Komplexe Bewegung aus [Hut77], Seite 397“ auf Seite 44 illustrieren. Dabei wird auch verständlich, warum eine praktische Anwendung dieses Systems selten ist.

4.2 Berner System zur Untersuchung nonverbaler Interaktion

Neben der Labanotation, die sich eines umfangreichen Glyphensortiments bedient, existieren weitere, stärker an eine maschinelle Verarbeitung angepaßte Systeme. Darunter fällt das in diesem Abschnitt prototypisch betrachtete „Berner System zur Untersuchung nonverbaler Interaktion“.

An der Universität Bern¹⁾ wurde in den 1970er Jahren mit der Entwicklung eines Systems zur Untersuchung nonverbaler Interaktion begonnen und dies Projekt mehrjährig fortgesetzt [FHP+81][HFF81]. Eine Motivation der Schöpfer dieses Systems war,

1. In Zusammenarbeit mit einer Reihe anderer Institute und Einrichtungen, siehe [FHP+81].



Spiral Fall Back Fall Forward Fall
Abbildung 14: Komplexe Bewegung aus [Hut77],
Seite 397

daß sie bei der Methodik der Analyse nonverbaler Interaktion funktionale Klassifikation vorfanden, die sie als eine ...

„Attribuierung funktionaler Etiketle nach dem vertrauten Vorbild der Ausdrucksdeutung“ [FHP+81], Seite 206

... im Gegensatz zu eigentlicher Analyse ansehen und ablehnen, da so die Interpretation der Bedeutung schon vom Kodierer vorgenommen wird und somit eine intersubjektive Analyse des Notierten nicht mehr möglich sei. Vielmehr argumentieren sie für eine reine Transkription, die eine empirische Referenz verfügbar macht und den Un-

tersuchenden in die Lage versetzt, Verhaltensphänomene, deren Funktion er untersucht, auch zu beschreiben — also für eine Transkription mit minimaler Interpretation.

Die Autoren des Berner Systems gehen von der Betrachtung von fünfzig (50) Systemen zur Sprachtranskription aus und finden, daß alle auf einem bivariaten Codierungsprinzip beruhen, nämlich einer phonetischen und einer zeitlichen Dimension. Die erste wird durch die einzelnen Zeichen des Codes erfaßt, die andere durch ihre Reihung dargestellt. Auf dieser Grundlage verfolgen sie einen zeitreihenbasierten Ansatz für ihr Notationssystem.

In Tabelle 7 sind die erfaßbaren Bewegungsvariationen dargestellt die in einem achtzigspaltigen sog. Ablochschemata numerisch kodiert eingetragen werden. Dies Schema erlaubt eine einfache Übertragung in die damals benutzte EDV, denn es ist direkt in Lochkartenstanzungen abbildbar.

Körperteil	# ^a	Dimensionen	Skalenniveau / Pos.	Erfaßbare Bewegungsvariation
Kopf	3	Sagittal	Ordinal / 5	Heben/Senken des Kopfes
		Rotational	Ordinal / 5	Links-/Rechtsdrehung des Kopfes
		Lateral	Ordinal / 5	Links-/Rechtskipfung des Kopfes
Rumpf	3	Sagittal	Ordinal / 5	Vorbeugen/Zurücklehnen des Rumpfes
		Rotational	Ordinal / 5	Links-/Rechtsdrehung des Rumpfes
		Lateral	Ordinal / 5	Links-/Rechtskipfung des Rumpfes
Schultern ^b	2	Vertikal	Ordinal / 3	Heben/Senken der Schulter
		Tiefe	Ordinal / 3	Vor-/Zurückschieben der Schulter
Oberarme ^b	3	Vertikal	Ordinal / 9	Heben/Senken des Oberarms
		Tiefe	Ordinal / 9	Vorwärts-/Rückwärtsbewegung
		Berührung	Nominal / 7	Oberarmkontakt mit Objekt

Tabelle 7: Zusammenfassung des Berner Notationssystems

Körperteil	# ^a	Dimensionen	Skalenniveau / Pos.	Erfassbare Bewegungsvariation
Hände ^b	9	Vertikal	Ordinal / 14	Auf-/Abwärtsverlagerung der Hand
		Horizontal	Ordinal / 9	Links-/Rechtsverlagerung der Hand
		Tiefe	Ordinal / 8	Vorwärts-/Rückwärtsverlagerung
		x/y Orient.	Ordinal / 9	Richtung der Hand in der Vertikalebene
		z Orient.	Ordinal / 5	Vorwärts-/Rückwärtsbeugung und
		Drehung	Ordinal / 9	Auf-/Abwärtsdrehung der Handfläche
		Öffnung	Ordinal / 4	Öffnen/Schließen der Hand
		Faltung	Nominal / 2	Finger (nicht) ineinander gefaltet
Berührung	Nominal/52	Handkontakt mit Objekt		
Füße ^b	7	Vertikal	Ordinal / 9	Auf-/Abwärtsverlagerung des Fußes
		Horizontal	Ordinal / 7	Links-/Rechtsverlagerung des Fußes
		Tiefe	Ordinal / 7	Vorwärts-/Rückwärtsverlagerung
		Sagittal	Ordinal / 5	Auf-/Abwärtskipfung des Fußes
		Rotational	Ordinal / 5	Einwärts-/Auswärtsdrehung des Fußes
		Lateral	Ordinal / 5	Einwärts-/Auswärtskipfung des Fußes
		Berührung	Nominal/10	Fußkontakt mit Objekt
Oberschenkel ^b	3	Vertikal	Ordinal / 5	Heben/Senken des Oberschenkels
		Horizontal	Ordinal / 5	Links-/Rechtsverlagerung des O.S.
		Berührung	Ordinal / 3	Kontakt der O.S. im Bereich des Knie
Sitzposition	2	Horizontal	Ordinal / 3	Links-/Rechtsverlagerung der S.P.
		Tiefe	Ordinal / 3	Vorwärts-/Rückwärtsverlagerung

Tabelle 7: Zusammenfassung des Berner Notationssystems (Forts.)

a. Anzahl der Dimensionen (wie in der nächsten Spalte aufgeführt).

b. Getrennte codierung für den linken und rechten Körperteil

In [FHP+81], Seiten 224-234 werden für all diese Bewegungsvariationen Definitionen, bzw. Erklärungen gegeben. Diese erfolgen meist über Beispiele in Form von Photographien oder durch textuelle Beschreibung, nicht jedoch durch die Angabe von Winkelbereichen oder ähnlich meßbarer Grundlagen.

Beispiele und Wahl der Merkmale lassen darauf schließen, daß das Berner System hauptsächlich für die Analyse von Aufnahmen sitzender Probanden entwickelt wurde. Diese Selbsteinschränkung ist aber willkürlich, da einige Merkmale nur besser ausdifferenziert werden müßten, um universeller einsetzbar zu sein.

4.2.1 Diskussion

Auch das Berner Notationssystem ist ein Schmidt'sches Notationssystem ([Sch97], Abschnitt 8.2), sofern eine Zuordnung von Notat und Bedeutung erfolgt, denn die wichtigste Errungenschaft im Berner System ist das Abkoppeln der Notation von der Funktion der zu notierenden Bewegung.

In den vorliegenden Beispielen [FHP+81], [HFF81] wird mit einer diskreten, äquidistanten sowie monotonen Zeitskala, meist in 5 s Diskretisierungsschritten gearbeitet. Da das Berner System ähnlich dem in der Computergraphik bekannte „key-framing“ Bewegung durch die zeitliche Folge von Posturen beschreibt, ist die Wahl der Länge der Zeitschritte von größter Wichtigkeit, da Bewegung so nur implizit codiert ist. Der aber sicherlich auffälligste Nachteil ist die numerische Notation in der sog. Datenmatrix, die vielleicht effizient von einem Rechner verarbeitet werden kann, aber für einen Menschen entweder nur mit großer Mühe oder langem Training entschlüsselt werden kann. Dies liegt zum einen an der Darstellung in numerischen Codes und zum anderen in der teilweise willkürlichen Wahl dieser Codenummern. Durch die Art der Beschreibung ist es aber sicher auch möglich, bei geeigneter Wahl der Quantisierungsschritte, eine sehr gut reproduzierbare Darstellung zu erhalten.

4.3 Hamburger Notationssystem

Nach der Labanotation und dem Berner System wird nun das „Hamburger Notationssystem für Gebärdensprachen“ (HamNoSys) eingeführt, da es wiederum anderen Designkriterien entspringt. Dabei wird Wert auf den kompositionellen Charakter HamNoSys' gelegt, der bei den Diskussionen für den Entwurf des Rahmensystems zur Gestenerkennung in den weiteren Kapiteln noch eine große Rolle spielen wird.

Die Autoren von HamNoSys beschreiben ihr System wie folgt:

„Das Hamburger Notationssystem für Gebärdensprachen, kurz HamNoSys, ist ein ‚phonetisches‘ Transkriptionssystem für Gebärden, das in seiner ersten Version vor zehn Jahren ganz in der Tradition Stokoe-basierter Systeme stand und das relativ weite Verbreitung gefunden hat. Die aktuelle Version 3 berücksichtigt erstmalig in größerem Umfang gewonnene praktische Erfahrungen in der Anwendung des Systems sowie Erkenntnisse über andere Gebärdensprachen als ASL, die diverse implizite Annahmen in HamNoSys falsifizierten. Sie geht damit sowohl im Aufbau als auch in der Beschreibungsmächtigkeit über den Stokoe-Rahmen hinaus. HamNoSys nimmt keinen

Bezug auf national verschiedene Fingeralphabete und ist damit prinzipiell international einsetzbar.“ (von dem World-Wide-Web-Server des Instituts für deutsche Gebärdensprache und Kommunikation Gehörloser der Universität Hamburg¹⁾)

Stokoe beschreibt in [Sto72] ein in [SCC76] verwendetes Notationssystem, auf dem die Ursprünge von HamNoSys beruhen. Es wurde dazu entwickelt, ASL in einem Wörterbuch zu notieren. Bei auf Stokoes Ansatz basierenden Systemen klassifizieren anhand mehrere unabhängiger Bereiche, typischerweise Handform, Handposition, Bewegung, Blickrichtung und Mimik. In HamNoSys wird eine Beschreibung in den Kategorien von manuellen Parametern (Handform, Handorientierung, Handposition und Bewegung) und nicht-manuellen Parametern (Rumpf, Kopf, Blickrichtung, Mimik und Mundform) vorgeschlagen. Am Institut für deutsche Gebärdensprache und Kommunikation Gehörloser der Universität Hamburg wurde ein auf ähnlichen Grundsätzen basierendes System seit Anfang der 1990er Jahre mit der Zielsetzung entwickelt, nicht nur die deutsche Gebärdensprache (DGS) zu notieren, sondern vielmehr von nationalen Fingeralphabeten und Zeichensprachen unabhängig zu sein. Dabei ist HamNoSys nicht das einzige System, das an diesem Institut entwickelt wurde, von Paspasyrou [Pap90] und von Fischer und Hanke werden auf dem World-Wide-Web-Server des Institutes²⁾ weitere Systeme zur Gebärdennotation vorgestellt. Statt einer Addition von Ausführungsaspekten strebt Paspasyrou ein Schriftsystem an, das über das Kriterium der Bedeutungsunterscheidung ausgewählte Aspekte der beiden Hauptparameter „Ausführungsorgan“ und „Bewegung“ miteinander kombiniert. Fischer und Hanke führen aus, daß die Bedeutungsunterscheidung auf der cheirologischen³⁾ (vgl.: „phonologischen“), nicht auf der graphematischen Seite ermittelt wird; in Anlehnung an den Fachterminus „Phonographie“ zur Charakterisierung von lautsprachbezogenen Schriftsystemen kann man also sagen, das Schriftsystem von Paspasyrou basiere auf dem Prinzip der Cheirographie³⁾. Im weiteren enthält das Schriftsystem von Paspasyrou das Leerzeichen als Segmentierungsglyphen, was unter

-
1. Unter „Unified Resource Locator“ (URL): <http://www.sign-lang.uni-hamburg.de/Projekte/HamNoSys/HamNoSys.html>. Überschrift: Notation von Gebärdensprache: HamNoSys
 2. Unter URL: <http://www.sign-lang.uni-hamburg.de/Arbeitspapiere/1/Inhalt.html>. Überschrift: Modell einer Notationsform als Diskussionsgrundlage zur Entwicklung einer Gebärdensprach-Gebrauchsschrift
 3. Cheirologie (gr. *cheir* Hand), Hand- oder Fingersprache der Taubstummen, hingegen beziehen sich die Begriffe Chirognomik, Chirologie, Chiromantik auf das Weissagen aus Handlinien; nach [DHS94].

linguistischer Sicht problematisch für Zeichensprachen ist, da diese in einem nicht a-priori segmentierten Strom geäußert werden. Ziel von [Pap90] ist jedoch nicht, eine „phonologische“ Beschreibung einer Gebärdensprache zu liefern, sondern eine Gebärdensprach-Gebrauchsschrift (am Beispiel der DGS) vorzuschlagen. Auch Fischer und Hanke wollen eine Gebärdensprach-Gebrauchsschrift schaffen, insbesondere für die DGS und mischen dafür HamNoSys-Glyphen mit Papaspyrous und eigenen Strukturen.

Bevor jedoch weitere Eigenschaften von HamNoSys diskutiert werden, wird im folgenden das Notationssystem zuerst einmal eingeführt. Eine ausführliche und vollständige Beschreibung des Systems ist in [Pri89] und auf dem World-Wide-Web-Server des Instituts für deutsche Gebärdensprache und Kommunikation Gehörloser der Universität Hamburg¹⁾ zu finden.

In HamNoSys werden zusammengehörige Bewegungen in kommagetrennten Zeichengruppen (Worten) notiert. Die erste Komponente eines Wortes ist ein optionaler Symmetrie-Operator, die zweite die Anfangskonfiguration, die in die vier Klassen non-manuelle Komponenten, Handform, Handstellung und Lokation untergliedert werden kann, die dritte bezeichnet Aktionen. Die folgenden Abschnitte dienen der Beschreibungen einiger Symbole.

4.3.1 Anfangskonfiguration

Die Anfangskonfiguration notiert diejenige Postur, die zu Beginn der Ausführung einer Geste eingenommen wird. Eine Anfangskonfiguration definiert sich durch *Handform*, *Handstellung* und *Lokation*.

Zur Notation der Handform stehen sechs *Grundformen* als Grundsymbole zur Verfügung, die mit Modifikatoren kombiniert werden. Abbildung 15 zeigt alle Grundfor-

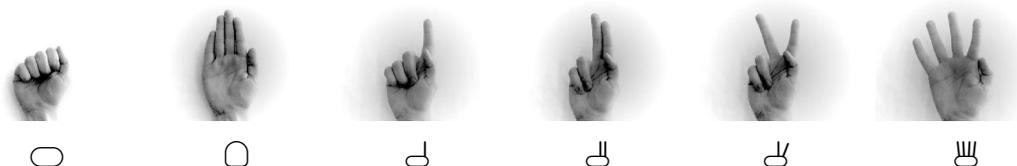


Abbildung 15: HamNoSys: Grundformen

men zusammen mit ihrer Notation in HamNoSys.

1. Unter URL: <http://www.sign-lang.uni-hamburg.de/Projekte/HamNoSys/HamNoSysErklaerungen/deutsch/Inhalt.html>. Überschrift: HamNoSys 3.0

Die Grundformen lassen sich mit diakritischen Zeichen (Akzenten) für Daumenstellung und Beugung (siehe Abbildung 16) kombinieren.

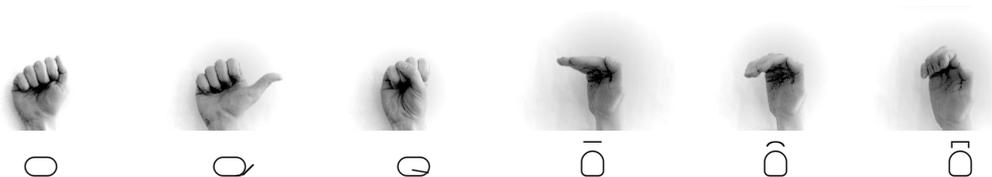


Abbildung 16: HamNoSys: Modifikationen der Grundformen

Die Modifikatoren der Verbindungsformen sind Zeichen für die Öffnungsweite und Beugung der beteiligten Finger (siehe Abbildung 17). Handstellungen lassen sich in



Abbildung 17: HamNoSys: Verbindungsformen

zwei Parameter aufteilen: Die Fingeransatzrichtung (Orientierung der Mittelhandknochen), welche durch die Orientierung der Handlängsachse gegeben ist, und die Handflächenorientierung, die bei gegebener Fingeransatzrichtung durch die Rotation der Hand um die Unterarmachse festgelegt ist. In Abbildung 18 auf Seite 50 sind Beispiele

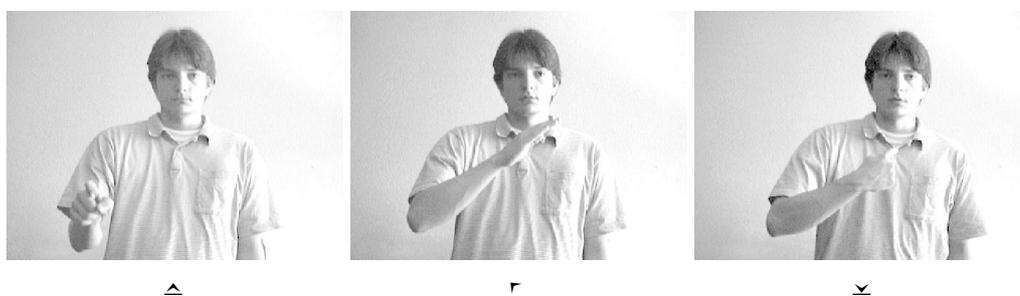


Abbildung 18: HamNoSys: Fingeransatzrichtungen

für verschiedene Fingeransatzrichtungen zu sehen. Dabei deutet in der Notation die Spitze des Dreiecks die Richtung an, und der Strich legt den Blickwinkel fest. Ohne Strich: Blick auf die koronale Ebene, Strich unter dem Dreieck: Blick auf die transver-

sale Ebene (Bodenebene, versetzt in die Körpermitte), Strich links vom Dreieck: Blick auf die sagittale Ebene (Symmetrieebene eines Menschen).

Die Handflächenorientierung wird durch Ellipsen-Symbole gekennzeichnet, wobei die schwarze Hälfte die Handinnenfläche andeutet; Abbildung 19 zeigt einige Beispiele.

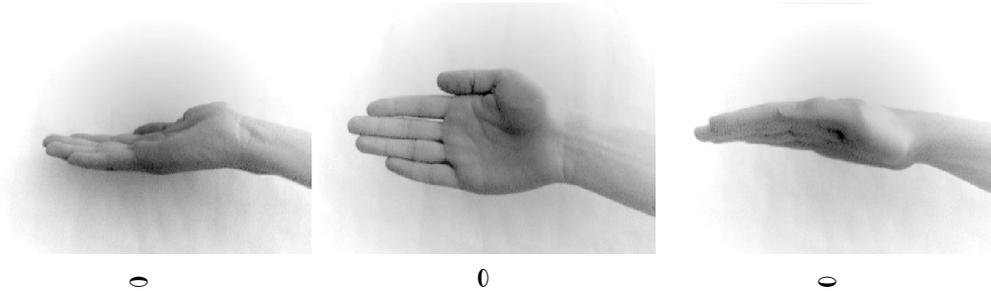


Abbildung 19: HamNoSys: Handflächenorientierung

Der Lokationsparameter gibt in HamNoSys den Aufenthaltsraum der Hand wieder. Er gliedert sich in die Dimensionen Höhe, laterale Position und Entfernung vom Körper. In Abbildung 20 auf Seite 52 ist die Einteilung des vertikalen Raumes mit den entsprechenden Zeichen zu sehen.

Zur Notation der lateralen Position stehen Modifikatoren der Höhenangaben zur Verfügung. Der Modifikator steht links von dem Basiszeichen, wenn eine Position links von der Körpermitte dargestellt werden soll, und entsprechend rechts vom Basiszeichen für die rechte Körperseite. Er hat die Form eines Quadrats und ist ausgefüllt bei einer kleinen Abweichung nach links bzw. rechts von der Mitte, und nicht ausgefüllt bei einer stärkeren Abweichung nach links bzw. rechts. Das Wort \square etwa notiert eine Position an der linken Seite des Brustbereichs.

Die dritte Dimension zur Festlegung einer Handposition ist der Abstand vom Körper. Diesbezüglich wird unter anderem zwischen \supset (hinter dem Körper), \times (Kontakt mit dem Körper), \supset^c (nahe am Körper) und $\supset\supset$ (weitestmöglicher Abstand) unterschieden. Kommt keines dieser Zeichen in einem Wort vor, dann wird implizit ein normaler Abstand angenommen, der als \emptyset oder normalerweise durch Abwesenheit eines Glyphen gekennzeichnet wird. Als normaler Abstand wird der Distanzbereich bezeichnet, in dem sich die Hände bei der Ausführung der überwiegenden Zahl der Gesten befinden. Abbildung 21 auf Seite 53 zeigt die Aufteilung der transversalen Ebene, gesehen von oben, in die verschiedenen Raumbereiche. Der normale Distanzbereich ist darin der nicht markierte Bereich zwischen $\supset\supset$ und \supset^c .



Abbildung 20: HamNoSys: Aufenthaltsort der Hand in der Vertikalen

4.3.2 Aktionen

Mit Aktionen sind alle dynamischen Aspekte einer Geste gemeint. Hier wird notiert, wie sich — ausgehend von der Anfangskonfiguration — die Geste in der Zeit entwickelt. Aktionen gliedern sich in absolute und relative Bewegungen. Absolute Bewegungen weisen einen Anfangs-, einen Zielort, und einen Bewegungstyp auf, wobei die Hand sich vom Start- zum Zielort bewegt (Beispiel: ◻→◻◻). Relative Bewegungen sind durch einen Startort und einen Bewegungstyp gekennzeichnet (Beispiel: ◻◻).

Bewegungen können translatorisch oder stationär sein. Bei einer translatorischen Bewegung bewegt sich die Hand als Ganzes, bei einer stationären Bewegung ändert sich die Position der Hand nicht, sondern nur ihre Form (Beispiel: ◻→◻◻). Translatorische Bewegungen werden weiter differenziert in gerade und gewölbte Bewegungen.

Jede Bewegung lässt sich durch zusätzliche Angaben genauer charakterisieren. Sie kann z.B. schnell oder langsam ausgeführt werden, abrupt stoppen oder besonders angespannt sein. Diese Eigenschaften werden in HamNoSys Bewegungsmodalitäten genannt.

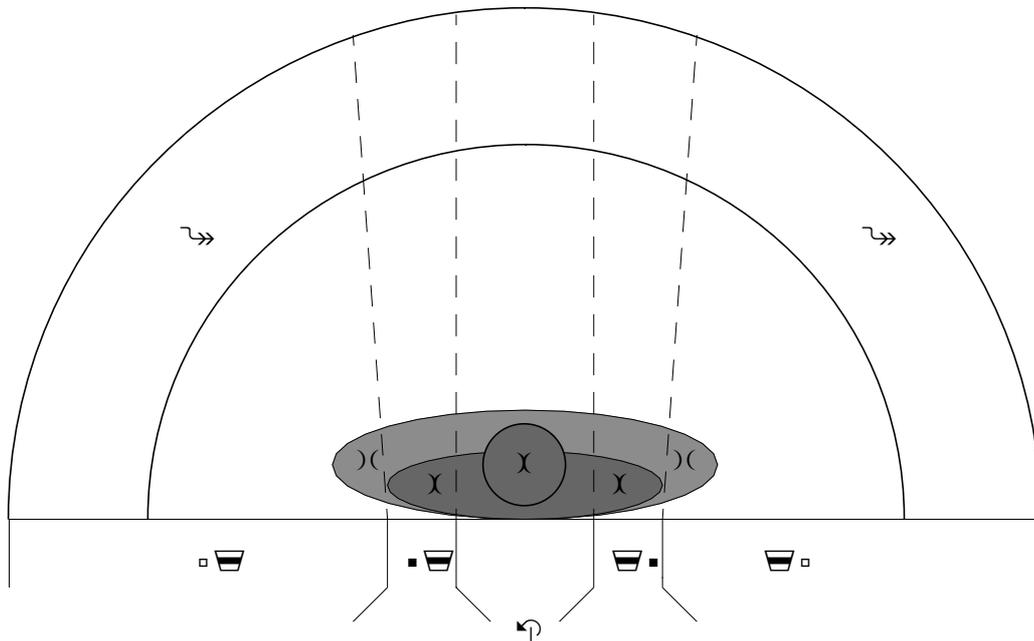


Abbildung 21: HamNoSys: Einteilung der transversalen Ebene

Aktionen lassen sich nach Belieben kombinieren. Auf diese Weise ist es möglich, auch komplexe Gesten in HamNoSys zu notieren, wie die Andeutung eines Quadrats durch „Nachzeichnen“ seiner Begrenzungslinien. Die Ausführung dieser Geste läßt sich in vier Bewegungsphasen einteilen. Alle vier Seiten des Quadrats werden dabei mit je einer geraden Bewegung der Hand dargestellt. Das Aufeinanderfolgen mehrerer Aktionen wird in HamNoSys mit abgerundeten Klammern notiert. Eine Bewegung im Quadrat wäre dann als $(\uparrow \rightarrow \downarrow \leftarrow)$ notiert. Die Hintereinanderausführung derselben Aktion ist mit den Wiederholungszeichen beschreibbar. Sie werden einer Aktion nachgestellt (etwa \uparrow^+), das Zeichen $^+$ steht für eine einmalige Wiederholung der Aktion.

BEIDHÄNDIGE GESTEN. HamNoSys unterscheidet zwischen der *dominanten* und der *nicht-dominanten* (auch *colateralen*) Hand. Die dominante Hand ist bei Linkshändern die linke, bei Rechtshändern die rechte Hand. Die Notation einer Geste, z.B. $\triangle \triangle 0$ bezieht sich auf die dominante Hand. Beidhändige symmetrische Gesten lassen sich durch Voranstellen eines Symmetriemarkers notieren. Die Fingeransatzrichtung muß bei rechts-links Vertauschung ebenfalls vertauscht werden. Die Handflächenorientierung ist bei der nicht-dominanten Hand grundsätzlich in der sagittalen Ebene gespiegelt. In Abbildung 22 ist beispielhaft die Notation von beidhändigen Gesten zu sehen.



: \bar{O} \circ \square \rightarrow $\#$

: Hand \wedge \circ \square \uparrow

: Hand \wedge \circ [?] X \square \rightarrow $\#$

Links: „Heranwinken“-Geste; Mitte: „Stopp“- oder „Wegdrücken“-Geste; Rechts: Infantile Hohngeste

Abbildung 22: HamNoSys: Beidhändige symmetrische Notation

4.3.3 Diskussion

Der Reichtum an Details, insbesondere die genaue Beschreibung der Handposturen, läßt die Ausrichtung von HamNoSys auf Zeichensprachen erkennen. Doch durch die Verallgemeinerung von einer bestimmten auf alle Zeichensprachen ergibt sich ein über den Rahmen von Zeichensprachen hinweg nutzbares Repositorium an Notationsmöglichkeiten für Gestenformen.

Es bleibt hervorzuheben, daß in HamNoSys zwar die Kodierung der Gesten unabhängig von der jeweiligen Funktion einer Geste vorgenommen wird, der Symbolvorrat aber durch eine langjährige Analyse von funktionstragenden, hier semiotischen, Gesten entstanden ist, und so aus „outstanding features“ (Kendon, vgl.: „Innere Struktur von Gesten“ auf Seite 21 ff.) besteht. Auch legt HamNoSys eine Syntax für die einzelnen Grundzeichen fest, mit denen dann komplexere Notate geformt werden können. Die Zeit jedoch wird nur implizit behandelt, zwar existieren Strukturen zur Notation von „Nacheinander“ und „Gleichzeitig“, so ist aber der Begriff „Gleichzeitig“ als menschliche Beschreibungskategorie und nicht im physikalischen Sinn zu verstehen.

4.4 Zusammenfassung

Die behandelten Gestennotationen sind unter speziellen Aufgabenstellungen entwickelt worden und gehen von verschiedenen Voraussetzungen aus. So nimmt z.B. die Labanotation als Notation für einen Tanz, der wiederum Musik voraussetzt, die Existenz eines Taktes zwingend an. Eine Übersicht verschafft die Tabelle 8, jedoch ist an-

	Labanotation	Berner System	HamNoSys
<i>Behandlung von Zeit</i>	als musik. Takt	diskrete Abtastrate	implizit
<i>Segmentierung von Sinneinheiten</i>	indirekt, durch Takt	nein	Kommata
<i>Anzahl der Glyphen</i>	> 300 ^a	numerisch und beliebig	~200
<i>RIM der Glyphen</i>	gering	keine	hoch
<i>Benutzung in Zieldomäne</i>	gering ^b	gewisse Verbreitung ^c	wachsend ^d

Tabelle 8: Vergleich der Notationen

- Mit unterschiedlicher Bedeutung, je nach Position im Notenstock.
- Gering bei professionellen Tänzern. Jedoch gibt es international wissenschaftliche Institute, die mittels Labanotation Musiktheorien des Tanzes untersuchen.
- Verbreitung über die Teilnehmerinstitute des Grundprojekts und darüber hinaus. Viele Ergebnisse finden sich heute als Regeln zur Transkription wieder.
- Eine Grundlage für die Entwicklung einer Gebrauchsschrift für die deutsche Gebärdensprache des Instituts für Deutsche Gebärdensprache und Kommunikation Gehörloser. Anwendung in technischen Systemen vgl. [HH96], [HBK99].

zumerken, daß die verschiedenen Systeme zu verschiedenen Zeiten für verschiedene Zwecke entwickelt wurden und nur bedingt vergleichbar sind. Festzuhalten bleibt jedoch, daß es sich durchgesetzt hat, die Form und die Bedeutung/Funktion von Gesten unabhängig zu notieren. Ein großes Problem bleibt die Darstellung von Zeit bzw. zeitlichen Zusammenhängen. Potentiell bietet das Berner System die physikalisch präzisesten Angaben, doch Labanotation und HamNoSys modellieren Zeit in einer für ihre jeweiligen Domäne angemessenen Art und Weise. Während im Berner System die geringste Interpretation vorgenommen wird, notiert HamNoSys schon klar segmentierte

Einheiten, allerdings ohne ihnen per se eine Bedeutung zuzuweisen. In der Labanotation ist dies nicht direkt vorgesehen. Technisch etwas Unhandlich sind die Systeme mit den eigenen Glyphen (Labanotation, HamNoSys), doch wenn die Anzahl der Glyphen sich im Rahmen hält und vor allen Dingen ihre Anordnung zu Buchstaben isomorph vorgenommen wird (HamNoSys), lassen sich leicht bijektive Abbildungen auf einfachere technische Darstellungen finden. Haben sie Glyphen eines Notationssystems eine hohe inhärente Herleitbarkeit (RIM), kann das Lesen von Notaten von menschlichen Bedienern einfach erlernt werden (HamNoSys). Wird ein System in seiner Ziel-domäne auch angewendet ist dies ein weiterer Hinweis darauf, daß nicht nur theoretische Überlegungen sondern auf Kriterien der Anwendbarkeit in das Design des Systems eingegangen sind.

Zu Anfang des Abschnittes „Theoretische Grundlagen“ auf Seite 11 wurden eine Reihe von Fragen formuliert, deren letzte (Nr. 7, auf Seite 11) in diesem Kapitel behandelt wurde. Wenn auch nicht alle Fragen letztlich beantwortet werden konnten, so ist in den vorangehenden Abschnitten zumindest eine Vertiefung der Problematik vorgestellt und der Bezug der einzelnen Fragestellungen untereinander und mit dem Gesamtthema dargelegt worden. Bevor jedoch ein neuer Ansatz zur Gestenbehandlung vorgestellt wird, wird eine Übersicht über einige gestenbehandelnde Systeme gegeben.

Kapitel 5

Gestenbehandelnde Systeme

In folgenden Kapitel werden einige Konzepte und Systeme beispielhaft vorgestellt, die zum einen Gesten im weiteren Sinne benutzen und deren Betrachtung Aufschluß über typische Sichtweisen und Ansätze gibt¹⁾. Zuerst wird ein Überblick über klassische Ansätze und erste Systeme gegeben, die Gesten als in der Eingabe behandelten, danach werden prototypisch ein amerikanisches und vier deutsche Systeme betrachtet. Die sich an die Einzelbetrachtungen jeweils anschließende Diskussion, betrachte die Systeme in Hinblick auf die bereits besprochenen theoretischen Eigenschaften aber auch in Hinblick auf hervorstechende Merkmale. Dabei wird auch betrachtet, ob und wie sich das System an geänderte Anforderungen anpassen kann oder angepaßt werden kann und auf welche Weise Wissen die Betrachtungen mit einbezogen wird.

5.1 Klassische Systeme

Das erste System, das auf multimediale Art und Weise Gestik und gesprochene Sprache integrierte, wurde von Richard A. Bolt [Bol80] unter dem Namen „Put-That-There“ vorgestellt. Es benutzt Spracherkennung und einen drei-dimensionalen Sensor, um Information aus der Sprache des Benutzers und der Position eines Zeigers auf einer wandgroßen Anzeigetafel auf Nachfrage hin aufzunehmen.

5.1.1 Werkzeugebene

Eine große Anzahl von klassischen gestenbehandelnden Systemen setzt die Handhabung auf der *Werkzeugebene* (engl., tool-level) an [WG89]. In diesen Systemen imitieren Gesten entweder Aktionen in der realen Welt wie greifen und werfen oder sind rein symbolisch, weisen also einer Bewegung, Postur oder Kombination aus beidem eine eindeutige, abgeschlossene Bedeutung zu. Schnittstellen der Werkzeugebene können sehr effizient und manchmal auch intuitiv sein. Sie werden jedoch unhandlich, wenn die Funktionalität unter Hierarchieebenen von Menüs und Moden versteckt ist.

1. Für video-basierte Mustereerkennungssysteme für Gesten pflegt Markus Kohler, Universität Dortmund unter dem URL <http://ls7-www.cs.uni-dortmund.de/research/gesture/> eine aktuelle Übersichtstabelle.

5.1.2 Eingeschränktes Gestenvokabular (Gestikular)

Ein weiteres Charakteristikum solcher Systeme ist die Beschränkung auf eine kleine Anzahl wohldefinierter Gesten, die der Benutzer teilweise erlernen oder sogar dem System antrainieren muß, um sie anwenden zu können. Wahlster beschreibt in [Wah91] ein gutes Beispiel für ein solches System, das deiktische Gesten benutzt. In seiner XTRA genannten Schnittstelle wählt der Benutzer aus einem Satz von Icons den Typ der nachfolgenden Geste aus, deren Bedeutung ist dann die durch das Icon definierte. Zwar ist diese Schnittstelle kontextabhängig, doch zwingt sie den Benutzer, spezielle Regeln für die Bedienung des Computers zu erlernen.

Zusätzlich zur Rolle der Manipulation auf der Werkzeugebene spielen unsere Hände eine große Rolle in der Kommunikation, da sie die natürliche Verbindung in die raumzeitliche Welt sind, in der wir leben [RS91].

5.2 ICONIC

Auch das „Iconic“-System benutzt Gesten in seiner Mensch-Maschine-Schnittstelle. Es wurde in der Mitte der 90er Jahre am Media Laboratory des Massachusetts Institute of Technology entwickelt [KST92][Wex94]. In einer Beispielapplikation dient es dazu, Einsatzplanung zur (Wald-)Brandbekämpfung durch Computereinsatz zu unterstützen. Dabei werden die Einsatzkräfte und Brandherde in Form von Icons auf einer an eine (Lein-)Wand projizierten computergenerierten Karte des Einsatzgebietes dargestellt. Als Systembenutzer ist ein Einsatzleiter konzipiert, der sich mittels verschiedener Referenzformen in Sprache und Gestik auf die räumlich präsentierten Entitäten beziehen kann, wenn er seine Anweisungen formuliert.

„Iconic“ sammelt Daten von zwei VPL DataGloves auf und wenn eine „komplette Geste“ erkannt ist, wird diese in einer Ausgabestruktur an weitere Verarbeitungsstufen weitergereicht.

Eine „komplette Geste“ ist dabei nach der klassischen Theorie (Efron) beschrieben, in der eine Geste aus Vorbereitungs-, Ausführungs- und Rückstellungsphase besteht. Der Benutzer beginnt seine Geste mit den Händen an der Hosennaht, führt seine Geste aus und bewegt seine Hände wieder an seine Seiten zurück. Das Heben und Absenken der Hände werden von dem Gestenerkennungssystem als Start- und Endmarker für eine Geste erkannt. Die Schlagphase wird im weiteren durch die von Kendon beschriebenen Pausen vor und nach dem Schlag erkannt, da sie eine charakteristische Signatur

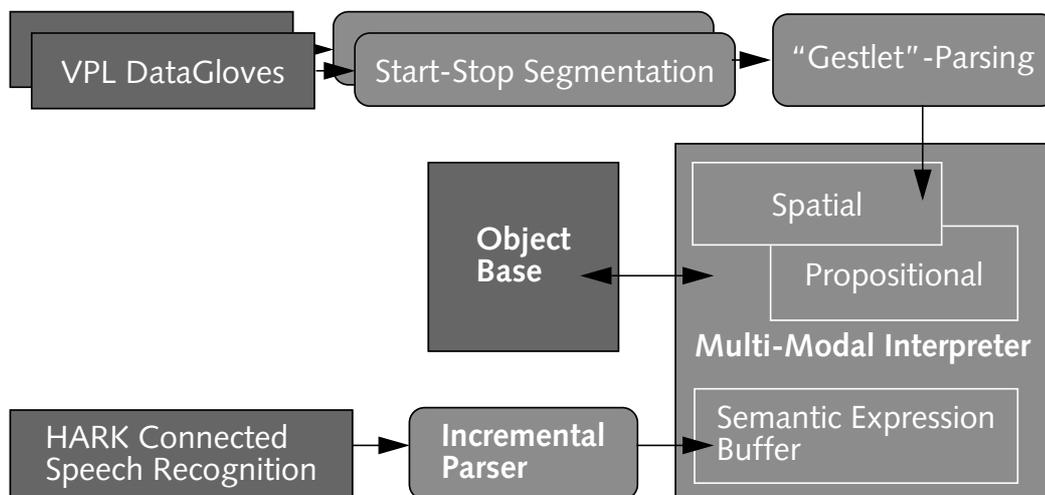


Abbildung 23: Das „Iconic“ System nach [Wex94]

in den Beschleunigungsprofilen hat. Eine Übersicht über die Systemarchitektur ist in Abbildung 23 gegeben.

Der „Iconic“ Gestenerkennung nimmt die Datenstrukturen der Handschuhe und von daran befestigten Polhemus 6DOF Positionssensoren und mischt sie zu signifikanten Intervallen, um dann Änderungen und stabile Zustände in den Datenwerten zu erkennen.

Diese Intervalle werden dann einer Menge von Grundformen zugewiesen, von denen angenommen wird, daß es sich dabei um primitive Komponenten von ikonischen und pantomimischen Gesten handelt. Die so kombinierten Daten werden durch Position, Bewegung und Orientierung der jeweiligen Hand angereichert, um ein sogenanntes „Gestlet“ [KST92] — eine Subgeste — zu erzeugen.

5.2.1 Diskussion

Obwohl Wexelblat eingesteht, daß Sparrell mit „Iconic“ nie vorhatte, eine generelle Lösung des Problems der gestischen Eingabe zu erstellen, sondern einen erweiterbaren Prototypen vorstellt, formuliert er doch eine Kritik an „Iconic“ [Wex94]:

Die Verwendung von Kendons Einteilung ist fragwürdig. Sogar Kendon selber führt aus, daß nur die Schlagphase selber notwendig ist. Das System aber ist so gebaut, daß der Benutzer gezwungen ist, die Pausen u.U. künstlich einzufügen, hauptsächlich weil

die Datenauswertung nicht schnell genug arbeitet. Das bedeutet für den Benutzer, daß er künstlich die Haltephasen verlängern muß, die in uneingeschränkter Gestenproduktion sehr kurz sind oder ganz wegfallen.

Die direkte Verwendung der Handschuhdaten als Merkmale in den „Gestlets“ macht das System unmittelbar geräteabhängig. Obwohl dies unter theoretischer Betrachtung kein Problem darstellt, propagiert die Geräteabhängigkeit so doch durch den ganzen Code, was das System wenig übertragbar macht.

Die Klassifikation basiert auf ASL, der amerikanischen Gebärdensprache, was als höchst fragwürdig zu kennzeichnen ist, da keine Erkenntnisse vorliegen, daß eine solche post-facto Kategorisierung übertragbar ist. Unter diesem Licht erscheinen die „Gestlets“ direkt als Schablonen (engl., templates), mit den bekannten Nachteilen eines solchen Ansatzes¹⁾.

Der bei „Iconic“ verfolgte Ansatz ist nicht skalierbar in Hinblick auf den Grad der Abstraktion der „Gestlets“. Generell sollte der Gestenerkennung in der Lage sein, inhaltsreichere Symbole zu generieren, wenn mehr über eine Geste bekannt ist.

Ein subtileres Problem liegt in dem Klassifizierungsansatz. Die Merkmale der Gesten sind eben die, die der Interpret benötigt, um eine eins-zu-eins Abbildung auf die Objekte in der Applikation zu leisten. Dies führt zu dem Problem, daß „Iconic“ vom Benutzer erwartet, daß er eine ikonische oder pantomimische Geste vollführt. Das System versucht dann, (Form-) Merkmale zu benutzen, um den Typ zu markieren. Das ist eine Vermengung von Form und Intention, die es dem Interpret hinterher unmöglich macht, multi-modale Kontextinformation der Gesteneingabe richtig zuzuordnen, da er schon vorbelastet ist.

5.3 Gebärdenerkennung mit Sensorhandschuhen

Seit 1994 wird an der Technischen Universität Berlin ein *Interdisziplinäres Forschungsprojekt* der Arbeitsgruppen Mikroperipherik, Semiotik und Technische Informatik mit dem Titel „Gebärdenerkennung mit Sensorhandschuhen“²⁾ (IFP-GS) unterhalten. Hier bildet das computergestützte Erkennen menschlicher Gebärden die Brücke zwischen dem systematischen Beschreiben von Gebärdenkodes und dem Messen der

1. Skaliert nicht gut über eine wachsende Zahl von Schablonen, keine Generalisierungsleistung für das zu erkennende Objekt, vgl. [Hum95] und [Nie83] Seite 105.

2. Siehe auch URI: http://pdv.cs.tu-berlin.de/forschung/IFP_dt.html

physikalischen Rohdaten von Gebärdenäußerungen. Die an den empirisch untersuchten Gebärden erprobten Methoden der Gebärdenbeschreibung und -notation werden darüberhinaus an den künstlichen Gebärdenkodes in der Mensch-Maschine-Interaktion getestet.

Eine zweite Aufgabe des IFP-GS besteht darin, geeignete Sensoren für die Messung der menschlichen Handbewegungen zu entwickeln, wobei Bewegungen in der Hand und Bewegungen der Hand im Raum unterschieden werden. Fingerbewegungen können mit Winkelsensoren gemessen werden. Aus den vielen unterschiedlichen Möglichkeiten, räumliche Handbewegungen zu erfassen wurde die Messung der dabei auftretenden dreidimensionalen Beschleunigungen ausgewählt. Diese Methode besitzt den Vorteil, Bewegungen zeitlich erheblich besser auflösen zu können (Abtastraten im *kHz*-Bereich) als andere, z.B. videobasierte Verfahren (Abtastrate 25-50 Hz).

Als verbindendes Element wird die Bearbeitung der grundlegenden Schritte auf dem Weg zur automatischen Erkennung von Gebärden eines gegebenen Codes gesehen. Dazu gehört im ersten Schritt die Eingabe menschlicher Gebärdenbewegung in den Rechner, also die Bewegungserfassung mittels Sensorhandschuh. Der nächste Schritt besteht aus der Verarbeitung und Erkennung von Gebärden. Nach der erfolgreichen Erkennung einzelner exemplarischer Gebärdenfelder wird eine auf variablen Gebärdenkodes reagierende Erkennungssoftware benutzt.

Als Notationssystem zur Beschreibung der Handbewegungen wird zunächst eine Untermenge von HamNoSys (vgl. „Hamburger Notationssystem“ auf Seite 47) vorgestellt, die über das in [HH96] vorgestellte System KINOTE – A Kinematic Notation for Gestural Motion Trajectories definiert sind. KINOTE besteht aus einer Menge von Funktionsscharen, deren Scharparameter zusammen mit der Grundfunktion die Beschreibungselemente von KINOTE ausmachen. Die Beschleunigungsdaten des Handschuhs werden auf in KINOTE beschriebene Trajektorien mit einem bestimmten Satz von Parametern und Grundfunktion abgebildet und aus dieser das HamNoSys-Symbol gewonnen.

In einer neueren Publikation [HHH98] wird dieser Schritt aber zu Gunsten einer Clusteranalyse aufgegeben. Die so erhaltenen Cluster werden auf verschiedene Weise klassifiziert werden, darunter kommen auch Hidden-Markov-Modelle (HMMs) zum Einsatz. Letztere hauptsächlich, um die zeitliche Integration der Information handhaben zu können. Es werden prinzipiell Bewegungsphasen erkannt, und Posturen als Pausen interpretiert und als Segmentierungshilfen benutzt.

Im IFP-GS werden einige wichtige Schritte der Gestenerkennung wie die systematische Beschreibung von Gebärdenkodes und dem Messen der physikalischen Rohdaten von Gebärdenäußerungen exemplarisch durch die beteiligten Arbeitsgruppen sehr gut beleuchtet. Im Kontext dieser Arbeit soll vor allem der Informatikanteil betrachtet werden. Hier ergeben sich folgende Anmerkungen.

5.3.1 Diskussion

Die ausschließliche Verwendung der Bewegungsphasen widerspricht eigentlich den „Gestengrammatiken“, wie sie in „Innere Struktur von Gesten“ auf Seite 21 vorgestellt wurden, da sie davon ausgehen, daß der Apex eine Postur darstellt.

KINOTE erlaubt zwar eine sehr saubere theoretische Darstellung von Trajektorien im Raum, ist aber bei näherer Betrachtung vielleicht für Gebärdensprachen adäquat, für coverbale Gesten jedoch sind Formen, wie zum Beispiel Raumschrauben mit variablen Radius, die gut in KINOTE dargestellt werden können, viel zu spezifisch und ausgelehnt.

Die daraufhin benutzte Merkmalsgewinnung durch Dimensionsreduktion mittels eines Cluster-Algorithmus in einen Strom von Codebuchindizes formt zwar signifikante Symbole, die aber „nur“ erlernt wurden, deren Wahl also keine Aussage über Gesten sondern nur über die Trainingsmenge zuläßt, was in diesem Kontext durchaus erwünscht ist. D.h. die emblematische Gesten werden direkt aus ihrer Form, ohne interpretierbare Zwischenstufe, erkannt.

Solange man HMMs verwendet, wird sich ein Trainieren des Modells nicht vermeiden lassen. Eine Fußnote in [HHH98] gibt eine Trainingszeit von einem Monat (!) auf einer DEC Alpha 3000/700AXP von 1995 für 500 emblematische Gesten an. Dies verhindert aber, daß das System aufgabenorientiert konfiguriert oder erweitert werden kann. Eine weitere Betrachtung des HMM-basierten Ansatzes ist bei der Diskussion von „SignRec“ auf Seite 65 ff. zu finden.

5.4 Viena

Das Projekt „*Virtual Environments and Agents*“¹⁾ (Viena) entstand im Rahmen des Forschungsprogrammes „Künstliche Intelligenz und Computergraphik“ der Universität Bielefeld unter Förderung des Forschungsverbundes „KI-NRW“ des Landes Nord-

1. Siehe auch URI: <http://www.TechFak.Uni-Bielefeld.DE/techfak/ags/wbski/viena/>

rhein-Westfalen bis 1997 durch die Arbeitsgruppe Wissensbasierte Systeme der Technischen Fakultät der Universität Bielefeld [WC95].

Viena stellt die Implementation einer Mensch-Maschineschnittstelle in Form eines sog. Interface-Agenten [Lau90b][MK93] dar, mit deren Vermittlung es einem Benutzer möglich ist, eine computergraphisch dargestellte Situation aus einem Innenarchitekturszenario¹⁾ mittels einfacher, natürlicher Sprache umzugestalten. Diese Schnittstelle ist eine Agentur aus spezialisierten Agenten, wie in Abbildung 24 dargestellt.

Über eine multimodale Eingabeschnittstelle übersetzt das Viena-System Benutzereingaben in eine interne Darstellung, die einem Mehragentensystem übergeben wird. Ein Buchführer-Agent verwaltet die erweiterte Grafikdatenbank und liefert anderen Agenten auf Anforderung Daten über die derzeit sichtbare Szene. Ein System von Raumagenten übersetzt qualitative Angaben wie „links von“ in entsprechende Geometriedaten. Dabei kann das System sich dynamisch an individuelle Benutzerpräferenzen adaptieren. Ähnlich übernehmen andere Agenten Spezialaufgaben bei der

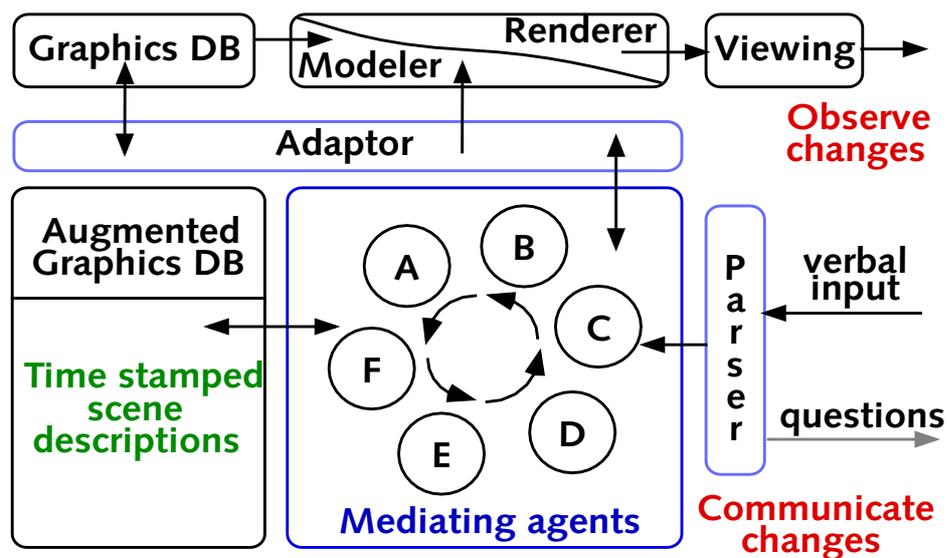


Abbildung 24: Die Agentur des Viena Systems nach [CJW95]

1. Aus naheliegenden Gründen wurden die Räume und Möbel aus der eigenen Universität modelliert, es wurden aber auch Geometriedaten, die von der ostwestfälischen Möbelindustrie freundlicherweise zur Verfügung gestellt wurden, benutzt.

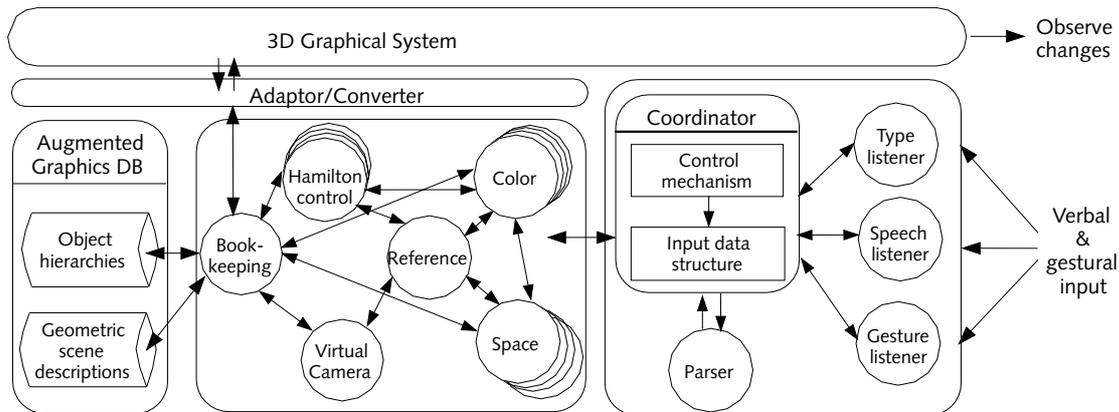


Abbildung 25: Die erweiterte Agentur des Viena Systems (nach [WLJ+97])

Umsetzung von Benutzereingaben. Die Agenten kooperieren bei der Berechnung der geänderten Szene. Der Benutzer kann die angebotene Szene in weiteren Eingaben korrigieren, das heißt, die Bedeutungsberechnung mit dem System aushandeln. Ein Adapter leistet die Datenkonvertierung zwischen den Agenten und dem Grafiksystem.

Dieses System erfuhr eine Erweiterung in Richtung einfacher Bewegungserkennung durch [Fro95] und einer benutzeradaptiven und multimodalen Eingabearchitektur, wie in [Len98] beschrieben und in Abbildung 25 skizziert. So ist es in Viena möglich, elliptische Bezüge aufzulösen und sich an Benutzerpräferenzen, z.B. bezüglich der Auswahl von deiktischen oder intrinsischen Referenzsystemen bei der Objektmanipulation, anzupassen.

Gesten können von zwei unabhängigen Sensoren erfaßt werden. Zum einen können sog. Richtungsgesten, also Hand- oder Armbewegungen nach links oder rechts mittels eines kamerabasierten Verfahrens erkannt werden. Dabei wird das Prinzip des Bewegungsmelders auf ein farbsegmentiertes Bild angewandt und eine Bewegungsrichtung ermittelt [Fro95]. Das andere System basiert auf einem Nintendo Powerglove, dessen Position auf den Mauszeiger abgebildet wird, die Postur „Ausgestreckter Zeigefinger“ (HamNoSys: \triangleleft oder \triangleright) löst dann die Zeigegeste aus, die dann eine Liste von ausgewählten Objekten zurückliefert [Len98]. Zeitlich werden diese Gesten in der sog. erweiterten Eingabestruktur einem zeitlich nachfolgenden deiktischen Sprachelement zugeordnet, dem sog. *Gestenplatz*. Dabei wird die Regel beachtet, daß Gesten der Sprache vorauslaufen. Die Integration basiert auf einem Zeitfenstermodell, das durch angeregte Schwingungszustände determiniert wird.

5.4.1 Diskussion

Zu Viena bleibt zu sagen, daß die Benutzung von Hand-(Arm-)bewegungen und auf den Mauszeiger abgebildeten Zeigegesten eine recht rudimentäre Form der Gestenbehandlung darstellt. Dabei muß aber beachtet werden, daß die Nutzung verschiedener Eingabeformen nur einen Teil der Herangehensweisen an das eigentliche Ziel Vienas — einem intelligenten Mittler zwischen Mensch und Maschine — ausmacht.

5.5 SignRec

Im Projekt „Automatische videobasierte Erkennung kontinuierlicher Gebärdensprache mit Hidden Markov Modellen – SignRec“ [HBK99] wurde im Rahmen eines bereits durchgeführten europäischen Forschungsprojekts¹⁾ SignPS am Lehrstuhl für Technische Informatik der Fakultät für Elektrotechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen Verfahren für die Erkennung isolierter Gebärden auf der Basis von Videodaten mittels Bakis-Hidden-Markov-Modellen (Bakis HMMs, siehe [Sch95], Kapitel 5) entwickelt. Dabei wird ein Ansatz verfolgt, der sich direkt aus dem Ansatz der HMM-basierten Spracherkennung ableitet. Aus einem Videobild eines mit Farbmarken behafteten Produzenten wird ein Merkmalsvektor extrahiert, der die manuellen Parameter widerspiegelt, ohne sie explizit zu modellieren. Mittels einer Menge von Bakis HMMs — eines für jede Gebärde — und eines stochastischen Sprachmodells (bigram/unigram) wird dann die Gebärde erkannt. In einem Evaluationszenario wurde eine Vokabular von 52 Gebärden, aus sieben Wortarten bestehend, benutzt. Es wurden 4 Stunden Videomaterial einer sich in DGS ausdrückenden Person benutzt, davon $\frac{1}{2}$ Stunde als Testdaten, der Rest als Trainingsdaten. Dabei erreichte das System die in Tabelle 9 auf Seite 65 dargestellten Ergebnisse.

Sprachmodell	Genauigkeit
<i>keines</i>	94,0%
<i>Unigram</i>	94,2%
<i>Bigram</i>	95,0%

Tabelle 9: SignRec Ergebnisse, vereinfacht nach [HBK99]

1. Siehe URL: <http://www.techinfo.rwth-aachen.de/hienz/research.html>

5.5.1 Diskussion

Einer der wesentlichen Vorteile eines HMM-basierten Ansatzes ist, daß die HMMs das Segmentierungsproblem als Nebenprodukt lösen. Dieses Merkmal, Zeit als Übergänge zu betrachten, führt aber zu Schwierigkeiten bei dem Design der HMMs. Die Zustandsanzahl in Bakis-HMMs kann variiert werden, was zu verschiedenen langen „Ketten“ führt. Wie lang muß eine Bakis-HMM-Kette für eine gegebene Gebärde sein? In SignRec wurde eine Heuristik verwendet, die die temporale Länge einer Gebärde mit der Länge (Zustandsanzahl) des HMMs korreliert¹⁾.

Ein solches HMM-basiertes Verfahren hat die schon für künstliche neuronale Netze im Abschnitt „Repräsentation“ auf Seite 24 diskutierten Vor- und Nachteile. In diesem Projekt dokumentiert sich ein weiteres Mal der prinzipielle Nachteil trainingsbasierter Verfahren: Die auf den ersten Blick hoch wirkenden Erkennungsraten²⁾ beziehen sich auf nur einen Produzenten, der dieses System 3,5 Stunden lang trainieren mußte. Wenn die Trainingsmenge auf mehrere Produzenten und mehrere Gebärdens ausgedehnt wird, muß die Merkmalsselektion anhand der dann zu ermittelnden Testergebnisse neu bewertet werden. Eine Hochrechnung der oben präsentierten Resultate ist also nicht ohne weiteres möglich. Dies liegt wiederum daran, daß HMMs genausowenig wie künstliche neuronale Netze aktiv oder durch ihre Form die innere Struktur des von ihnen erkannten Musters erklären können.

5.6 SGIM

Das Projekt „sprach- und gestenbasierte Interfaces für Multimediaanwendungen“ (alternativ „Speech and Gesture Interfaces for Multimedia“), kurz SGIM, wurde 1996 als neues Rahmenprojekt für die Arbeiten im Bereich „Gesten“ an der Arbeitsgruppe Wissensbasierte Systeme der Universität Bielefeld gegründet. In SGIM werden sprachbegleitete und sprachbegleitete Körpergestik, insbesondere die der oberen Gliedmaßen, in multimedialen Umgebungen untersucht und der Versuch unternommen, sie technisch nutzbar zu machen.

5.6.1 SGIM in der Virtuellen Wissensfabrik

In der 1996 aufgebauten Arbeitsgemeinschaft „Multimedia NRW: Die virtuelle Wissensfabrik“ liegt einer der Schwerpunkte auf der Mensch-Maschine-Interaktion.

-
1. Herman Hienz während seines Vortrags auf dem Gesture Workshop '99 in Gif-sur-Yvette.
 2. Technisch gesehen sind sie tatsächlich nicht zu monieren, das Problem wird aber durch Wexelblats „Kritik“ auf Seite 6 verdeutlicht.

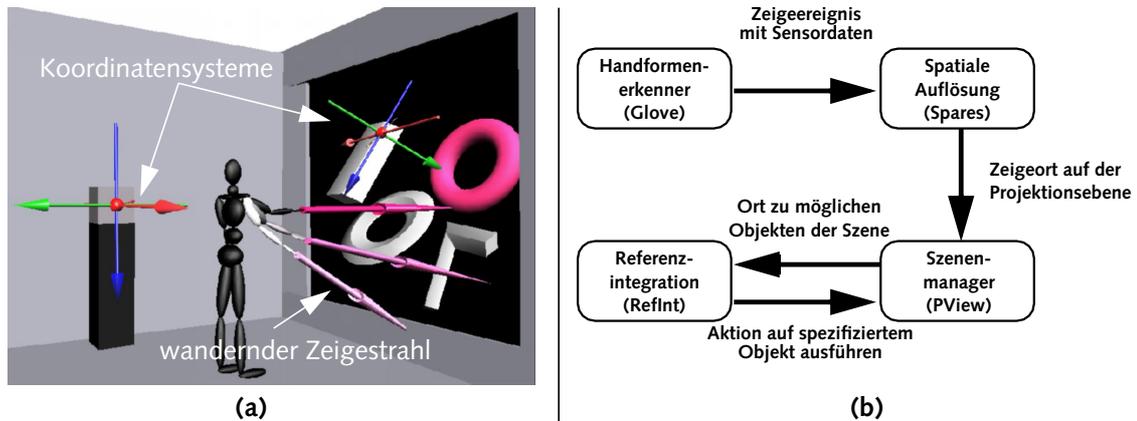


Abbildung 26: SGIM: schematische Sicht und Architektur, nach [LFJ+98]

Als Beitrag des SGIM-Projektes werden Techniken entwickelt, die dem Benutzer eines Multimedia-Systems den Einsatz grober, auf Armzeigen und Kopfstellung basierender gestischer Kommunikation ermöglichen. Damit sollen Begrenzungen von üblichen Bildschirm-Displays überwunden werden und erweiterte gestische Interaktionstechniken für den Einsatz mit Groß-Displays (Wandprojektionen, Workbenches, Caves) vorbereitet werden, die ein möglichst freistehendes, komfortables Agieren erlauben.

Als Erprobungsszenario dient, in Abstimmung mit den anderen Projekten der Arbeitsgemeinschaft, eine mobile Plattform („City-Mobil“), die in einer Großbildprojektion aus CAD-Modellen virtuell konstruiert wird. Die Bielefelder Gruppe „Multimodale Mensch-Maschine-Interaktion“ (M4I), das Institut für Medienkommunikation der GMD St. Augustin und das Europäische Zentrum für Mechatronik in Aachen¹⁾ arbeiten an der Entwicklung eines gemeinsamen Demonstrationssystems mit dem Ziel, die Forschungsergebnisse umzusetzen und sie in einer realen Anwendung zu evaluieren.

In SGIM wird als Schlüsselfunktionen das Auswählen und darauf folgende Manipulieren von virtuellen Objekten mittels Gesten betrachtet. Dabei müssen neben qualitativen Merkmalen auch quantitative Merkmale von Gesten ausgenutzt werden. Zu diesem Zweck wird ein Agentensystem vorgeschlagen, dessen Skizzen in Abbildung 26 zu sehen sind. In Abbildung 26 (a) bedeutet ein Benutzer einen Reifen,

1. Weitere Kooperationen bestehen mit dem Graduiertenkolleg „Aufgabenorientierte Kommunikation“ und dem Sonderforschungsbereich 360 „künstliche situierte Kommunikatoren“ an der Universität Bielefeld.

der Zeigestrahle überstreicht dabei die Szene. Auch sind die Koordinatensysteme eines Objekts und der Orts- und Lagesensoren (hier ein „Flock of Birds“ Sensor für sechs Freiheitsgrade) eingezeichnet. Es muß also nicht nur ermittelt werden ob der Benutzer zeigt, sondern auch wohin er zeigt. Eine Bewegung wird als Gestenhypothese durch das Agentensystem gemäß Abbildung 26 (b) gereicht. Die Validität einer solchen Hypothese wird mit weiteren gleichzeitig z.B. aus einem Spracherkenne erhaltenen Hypothesen bewertet und im Falle einer signifikanten Beurteilung, wird eine Aktion wie z.B. eine Translation ausgeführt. Eine Gestenhypothese wird aus Merkmalen der Handkonfiguration (Winkel und deren ersten beiden Ableitungen in der Zeit) und Handposition gewonnen. Dabei können interaktiv beliebig viele Merkmalsätze zur Formbeschreibung von Gesten erzeugt werden und damit auch beliebig viele Gesten definiert werden.

5.6.2 Diskussion

Die Art der verwendeten Gesten ist durch die Schnittstellenmetapher der Anwendung gegeben und orientiert sich an der Domäne — der virtuellen Konstruktion. Die Definition der Gesten erfolgt direkt über ordinale Sensorparameter und nicht über eine Schicht, die Begriffe (nominal) aus der Anwendungs- oder Gestendomäne benutzt wie „zur Projektion hin“ oder „gebeugt“.

5.7 Zusammenfassung

Ähnlich wie bei den Notationssystemen ist auch für die gestenbehandelnden Systeme eine direkte Gegenüberstellung mit Vorsicht zu betrachten, da alle Systeme andere Zielsetzungen und Voraussetzungen haben. In Tabelle 10 ist trotzdem eine Zusammenfassung wichtiger Eigenschaften dargestellt.

	ICONIC	IFP-GS	VIENA	SignRec	SGIM
<i>Einsatzzweck</i>	Brandbekämpfung	Emblem-erkennung ? ^a	Schnittstellen Agent	Gebärdenspracherkennung	virtuelle Konstruktion
<i>Sensoren</i>	Polhemus & Data-Glove	TUB-Glove	Nintendo PowerGlove	Video-kamera	Flock of Birds & CyberGlove

Tabelle 10: Gestenbehandelnde Systeme

	ICONIC	IFP-GS	VIENA	SignRec	SGIM
<i>Art der Gesten</i>	ikonisch / pantomimisch	emblematisch	deiktisch / emblematisch	Gebärden	Typen I, II & IV ^b
<i>Anzahl der Gesten</i>	wenige (? ^c)	500 Elemente	3	53	definierbar
<i>Modalitäten</i>	Sprache / Geste	Geste	Sprache / Geste	Geste	Sprache / Geste
<i>Training</i>	?	ja	nein	ja	nein
<i>Benutzerabhängig</i>	?	ja	nein	(ja) ^d	(nein) ^e
<i>Echtzeitfähig</i>	?	?	einige s Verzögerung	?	ja

Tabelle 10: Gestenbehandelnde Systeme (Forts.)

- a. eigentlich Gebärdenerkennung. Doch in [HHH98] wird sich nur auf Embleme bezogen.
 b. vgl. Tabelle 3, „WBS System nach [LW98]“ auf Seite 17
 c. Das „?“ zeigt an, daß keine gesicherten Angaben vorliegen.
 d. Da mit nur *einem* Benutzer trainiert und getestet wurde.
 e. Solange die vorhandenen Sensoren (Handschuhe) passen.

Die meisten Systeme betrachten Gesten als eindeutige Zeichen, die direkt auf Funktionen abgebildet werden. Neuere Ansätze versuchen Gesten direkt auf gewisse Konstrukte des Sprachkanals (z.B. „Gestenplätze“ in Viena) abzubilden. Bei ICONIC, IFP-GS und Viena ist eine Unterscheidung von Form und Funktion nur im Ansatz möglich. Eine systematische Trennung der Formanalyse und der Bedeutungsanalyse auf verschiedenen Stufen ist nur zu finden, wenn die Formanalyse auf die Unterscheidung im Systemdesign vorgegebener Gesten beschränkt ist.

Kapitel 6

Konzeption eines Gestenerkenners

In den vorangehenden Kapiteln wurde Form, Bedeutung und interne Struktur von Gesten auf verschiedenen Ebenen vorgestellt, sowie Systeme betrachtet, die sich dieses komplexen Ausdruckssystems bedienen, um Mensch–Maschine–Schnittstellen auf die eine oder andere Art und Weise zu realisieren. Dabei wurde vor allen Dingen die Gesten und ihre Eigenschaften in den Vordergrund gestellt. In diesem Kapitel werden nun (software-) technische Aspekte mit einbezogen und somit eine Konzeption erarbeitet. Darüberhinaus wird diese Konzeption durch eine Architektur¹⁾ für ein Gestenerkennungssystem umgesetzt.

6.1 Systemgrundlagen

Eine wichtige Anforderung an ein Softwaredesign ist die opaquere Kapselung von Hardwareabhängigkeiten; dazu kommen Anforderungen an die Robustheit und Wartbarkeit von Objekt- und Quellcode. Ein effizientes Design versucht Objektcodesegmente so gut es geht wiederzubenutzen, damit eventuelle Fehler einfacher gefunden werden und nachhaltig beseitigt werden können. Code, der auf einer opaqueren Kapselung der Hardware aufsetzt, wird portabel genannt, wenn es Kapselungen mit der gleichen Schnittstelle für verschiedene Hardwareplattformen gibt. Damit wird klar, daß klare Spezifikationen wichtig sind, und die Verwendung standardisierter Codesammlungen (Bibliotheken) elementar ist. Auch die verwendeten Programmierparadigmen und Programmierstile spielen dabei eine Rolle, z.B. können nur Programme die logisch parallelisiert wurden, dann auch parallel auf Multiprozessor-Maschinen laufen. Die hier angeführten Kriterien ziehen sich durch verschiedene Ebenen, von der Wahl der Programmiersprache, der Benutzung von weiteren Bibliotheken bis zu dem Erstellen eigener Bibliotheken und Programme.

6.1.1 Systemkonzeption

Diese Anforderungen werden damit angegangen, Probleme modular zu spezifizieren und möglichst unabhängig von einander zu implementieren. Im weiteren sollten die Module in Gruppen ähnlichen Abstraktionsgrades (sog. *vertikale Organisation*) und

1. Eine Architektur wird hier als die Realisation einer Konzeption verstanden.

Aufgabenbereiches (sog. *horizontale Organisation*) gegliedert werden. Die vertikale Organisation wird typischerweise durch die Unterteilung in verschiedene Schichten vorgenommen, die eine bestimmte Sicht auf das Problem, opaque bezüglich weiter „unten“¹⁾ liegenden Schichten, darstellt; also ein Implementierungsunabhängiges API²⁾ zur Verfügung stellt. Die horizontale Organisation wird durch die funktionale Gliederung der Module vorgenommen. Interaktionen dieser Module werden typischerweise durch Module in tieferen Schichten realisiert. So strukturierte Systeme werden „*modular*“ — im Gegensatz zu „*monolithisch*“ — genannt.

Neben diesen allgemeinen Anforderungen lassen sich weitere aus der spezifischen Aufgabe „Gestenerkennung“ ableiten: Das System sollte schritthaltend im Sinne der MMK sein, also die Eingabe in Zeiten analysieren, die eine Antwort in einem für den interagierenden Menschen akzeptablen Zeitrahmen liefert. Es sollte einfach an eine konkrete Aufgabenstellung anpaßbar sein. Es sollte benutzerunabhängig sein. Und nicht zuletzt sollte es — da die Gestenforschung noch große Fortschritte macht — an neue Erkenntnisse anpaßbar und erweiterbar sein.

VERTIKALE ORDNUNG. Als vertikale Ordnung wird das in Abbildung 27 dargestellte Schichtenmodell vorgeschlagen. In dem Modell werden jeweils aktive Komponenten als Schicht und Datenstrukturen als Austauschmedium angesehen. Eine solche Struktur orientiert sich an dem zu erwartenden Datenfuß und ermöglicht eine schnelle Datenreduktion auf bedeutungsstarke Entitäten (Symbole).

In der Sensorschicht ist die Informationsverarbeitung direkt mit physikalischen Effekten gekoppelt, ihre wichtigste Leistung besteht in der Konvertierung der beobachteten Variablen (Observablen) in eine maschinenlesbare Darstellung. Dabei sind allgemeine Filter und Konvertierungsroutinen durchaus in dieser Schicht zu verorten, solange sie keine weiteren Annahmen einfließen lassen, als die, die zur Darstellung der Observablen

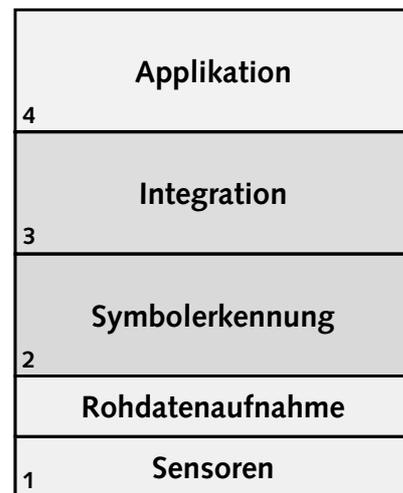


Abbildung 27: Schichtenmodell

1. In der Tradition der Schichtmodelle wird die spezialisierteste bzw. hardwarenäheste Schicht auch erste oder unterste Schicht genannt.
2. API: engl. Abk. Application Programmer's Interface: Programmierschnittstelle für den Anwendungsprogrammierer

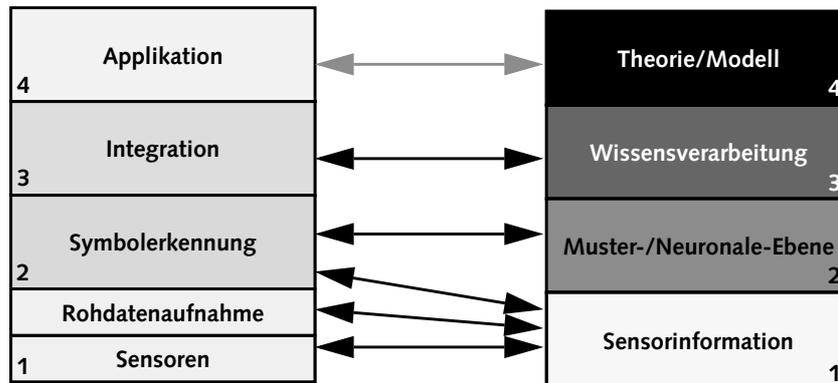


Abbildung 28: Schichtenmodellvergleich, eigenes und [Rad96]

bekannt sind. In der Rohdatenaufnahme wird die nun vorliegende Information in Hinblick auf weitere Verarbeitungsstufen digital aufbearbeitet. In der Symbolerkennung werden aus den Daten relevante Merkmale gefiltert und daraus Symbole generiert, die im weiteren Prozeß verarbeitet werden. Darauf hin kann dann in der Integrations-ebene in mehreren Schritten die symbolische Information kondensiert werden, so daß sie in einem für eine gewählte Applikation zutreffenden Format vorliegt. Als Applikation wird jenes System aufgefaßt, welches die kommunizierte Information in gleichwelcher Form umsetzt. Da höhere Funktionen der Kommunikation, wie z.B. die Diskurskontrolle, sehr stark mit der Umsetzung der Information verwoben sind, werden sie hier ebenfalls zur Applikation gerechnet.

Dieser Ansatz läßt sich mit ähnlichen vergleichen. Radermacher stellt in [Rad96] ein Schichtenmodell für den kognitiven Apparat eines zukünftigen autonomen Systems in einem Perception-Aktion Modell dar. In Abbildung 28 sind die beiden Modelle nebeneinander dargestellt¹⁾. Bei Radermacher findet auf der untersten Ebene schon die Merkmalsextraktion statt. Die so erhaltene subsymbolische²⁾ Information wird dann auf der Merkmalsebene, gegebenenfalls unscharf, klassifiziert und das Resultat dann der Wissensverarbeitung übergeben. Dort wird die nun symbolische Information im Kontext bekannter Konzepte identifiziert und integriert. Auf der Theorie-Ebene werden dann Weltmodelle benutzt, um z.B. durch logisches Schließen Pläne zu erarbeiten

1. Sie auch [Jun98], Abschnitt 5.1 für eine Diskussion Radermachers Modells im Kontext einer anderen Architektur.
2. Subsymbolischer Information kann weder direkt eine Bedeutung noch ein Bezeichner zugewiesen werden.

und zu verfolgen. Radermachers Modell ist kognitiv fundiert und beschreibt ein komplettes *Regelungssystem*. Das hier vorgeschlagene Modell hingegen beschreibt einen Ausschnitt eines Gesamtsystems, dessen Anwendungsteil sich in der Applikationsschicht verbirgt. Daher sind Applikationsebene einerseits und Theorie-Ebene andererseits nur bedingt mit einander vergleichbar. Der wesentliche Unterschied liegt in der Bewertung der Sensor-Ebene. Radermacher verortet in dieser Ebene schon die Merkmalsgewinnung und führt Phoneme als Beispiel aus der Sparacherkennung als solche Merkmale an. In dem hier vorgestellten Modell ist hingegen die Merkmalsgewinnung in die Symbolerkennung verlagert, damit die Schnittstelle der ersten und der zweiten Ebene eine technisch einfachere Struktur hat. Gleich sind die Modelle jedoch in einem wesentlichen Element: Mit ansteigen der Ebene wächst die Komplexität der verarbeiteten Information und höhere Ebenen haben nur eine mittelbare Sicht auf die Sensorik. Diese Form der Modellierung entspricht dem aus Bewußtseinsmodellen hergeleiteten hierarchischen Zeitmodellen, wie sie in Abschnitt „Zeitbegriffe“ auf Seite 31 dargestellt sind.

In Tabelle 11 werden die Zeitmodelle den Schichten des Modells grob zugeordnet, in

Zeit	Schichten	Ballard	Pöppel
>10 s	Applikation	Komplexe Aufgabe	semantische Bindung
2 s – 3 s	Integration	Einfache Aufgabe	subjektive Gegenwart
300 ms	Symbol- erkennung	Physikalischer Akt	—
50 ms		Neuronaler Akt	Ordnungsschwelle
10 ms		Neuraler Schaltvorgang	visuelles GZF ^a
1 ms – 7 ms	Sensoren	Neuronale Entladung	—

Tabelle 11: Einordnung des Schichtenmodells in die Zeitmodelle

a. Gleichzeitigkeitsfenster, siehe „Gleichzeitigkeit – Ungleichzeitigkeit“ auf Seite 31

[Jun98], Seite 84 ist zudem eine grobe Einordnung der Zeitskalen in Radermachers Modell zu finden.

HORIZONTALE ORDNUNG. Neben der Komplexität der verarbeiteten Information, kann auch nach Aufgabenbereichen geordnet werden. So können Module aufgabenorien-

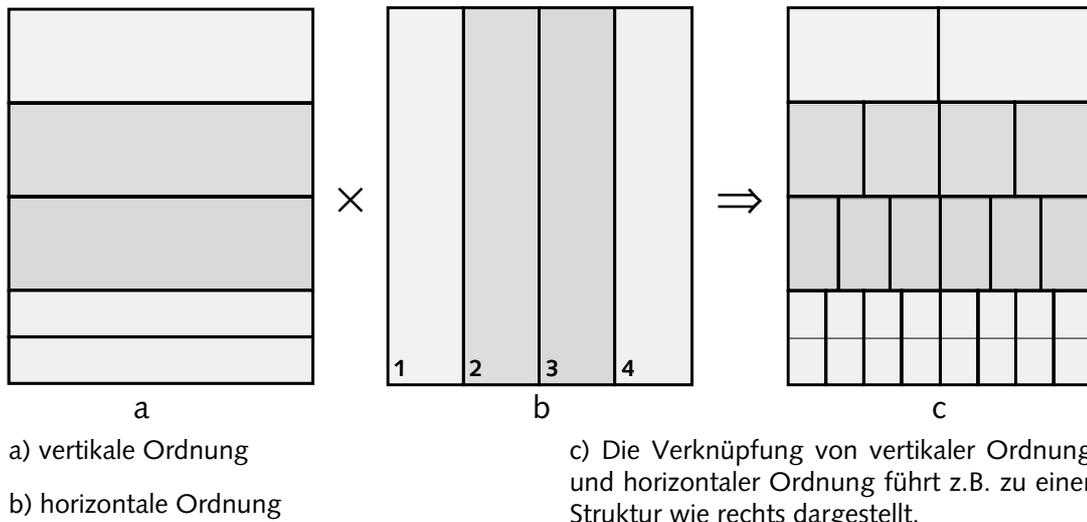


Abbildung 29: vertikale und horizontale Ordnung

tiert klassifiziert werden wie z.B. in Abbildung 25, „Die erweiterte Agentur des Viena Systems (nach [WLJ+97])“ auf Seite 64 in „Color“, „Space“ und „Reference“. Abbildung 29 stellt die gleichzeitige Anwendung von vertikaler und horizontaler Ordnung dar. Die dabei resultierende Struktur kann als Graph aufgefaßt werden, wenn man die Flächen als Knoten und ihre direkte Nachbarschaftsbeziehung über vertikale Schichtgrenzen hinweg als Kanten interpretiert.

Solche hierarchischen Graphen finden sich auch bei Minskys „Society of Mind“ [Min85] und werden als Agentensysteme beschrieben. Die Grundfrage, mit der Minsky sich auseinandersetzt, ist: Wie entsteht Intelligenz aus Nicht-Intelligenz? Minsky schlägt vor, eine Gesellschaft von kleinen Einheiten aufzubauen, die er Agenten nennt. Da an diesen Einheiten keine Intelligenz festzumachen ist, muß diese durch die Interaktion untereinander entstehen. Minsky erkennt damit die Bedeutung der Interaktion und Kommunikation für die Intelligenz an, deren Beobachtung schon von Turing als so wichtig erachtet wurde, daß er schon 1950 auf der Grundlage der Beobachtung von Kommunikationsprozessen seinen Turingtest formulierte. Minsky geht also davon aus, daß sich jeder Prozeß in feinere Prozesse zerlegen läßt. Seine Sichtweise sprengt einen Vier-Schichten-Rahmen und erlaubt Verfeinerungen bis zu elementaren Verarbeitungsvorgängen und Verallgemeinerungen bis zu der gesamten betrachteten Entität. Im weiteren soll das Schichtenmodell als eine Gliederung einer

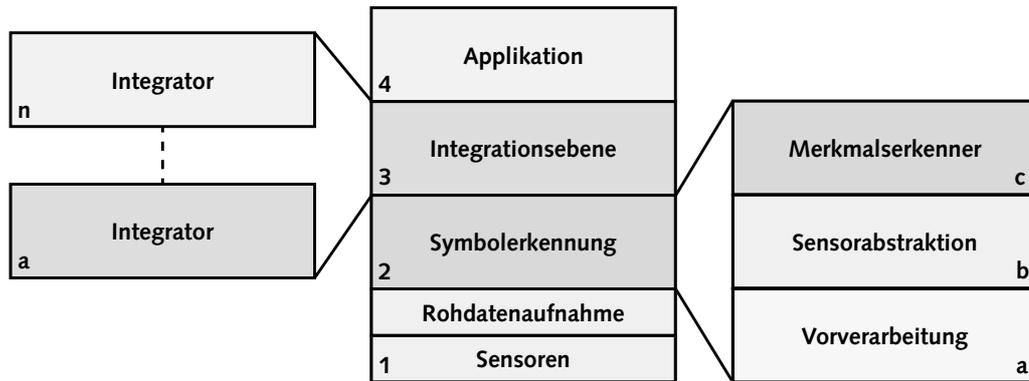


Abbildung 30: Das erweiterte Schichtenmodell

größeren Modul- bzw. Agentenhierarchie begriffen werden; die Schichten können also gegebenenfalls untergliedert sein.

Wird die Tabelle 11 unter diesen Aspekten betrachtet, fällt auf, daß die Symbolerkennungsschicht auffällig viele Zeitebenen umfaßt. Das legt nahe, diese Schicht weiter zu unterteilen. Dabei wird der Gedanke verfolgt, daß die Observablen nicht notwendigerweise die aussagekräftigsten Daten liefern, sondern daß es nur mit Hilfe von a priori Wissen herleitbare Variablen gibt, die vorzugsweise die Merkmale für die nächste Stufe liefern. Neben Minskys Hierarchie wurden bereits im Abschnitt „Repräsentation“ auf Seite 24 weitere hierarchische Theorien betrachtet. Zusammen betrachtet legen diese nahe, auch den Integrationsprozeß hierarchisch zu organisieren. Das nach diesen Vorgaben erweiterte Schichtenmodell ist in Abbildung 30 zu sehen. Im folgenden Abschnitt werden nun softwaretechnische Kriterien auf diese Konzeption angewendet.

6.1.2 Softwaretechnische Überlegungen

Neben den konzeptionellen lassen sich die folgenden, in Angang „Prinzipien der Softwareentwicklung“ auf Seite 155 ausgeführten, technische Anforderungen formulieren: Software sollte modular, portabel und seiteneffektfrei aufgebaut sein und trotzdem jede Funktion nur einmal implementiert sein. Die Schnittstellen sollten wohldefiniert sein und das Programmiermodell auf Parallelisierbarkeit achten.

Als Lösung dieses Problems lassen sich drei Konzepte verfolgt: Zu einem ein streng auf „shared objects“ — gemeinsam genutzten Bibliotheken — aufbauendes Programmiermodell und zum anderen daß ein auf interne Parallelisierbarkeit achtenden-

des Programmierparadigma, wie es in Angang „Thread-sichere Programmierung“ auf Seite 167 dargestellt ist und im weiteren das Strukturierungsparadigma der agentenorientierten Programmierung. Gemeinsam genutzten Bibliotheken sprechen einerseits die Forderung nach wohl definierten Schnittstellen an und andererseits ermöglichen sie es, eine Funktion nur einmal zu implementieren und trotzdem in verschiedenen Modulen zu benutzen.

AGENTEN. Der hier benutzte Agentenbegriff bezieht sich einerseits auf Agenten im Sinne Minskys [Min85] (vgl. Abschnitt „Systemkonzeption“ auf Seite 71) andererseits auf Agenten, die mehr im Sinne von Wooldridge und Jennings [WJ95] ausgeprägt sind. Eine mittlerweile weit verbreitete Definition ist die sogenannte schwache Definition von Wooldridge und Jennings [WJ95]. Dort wird ein Hardware oder Software basiertes System als Agent bezeichnet, wenn es über folgende Eigenschaften verfügt:

1. *autonom* (evtl. semi-autonom): Der Agent ist eigenständig und kontrolliert seine Aktionen und inneren Zustände.
2. *sozial*: Der Agent kann mit anderen Agenten interagieren/kommunizieren.
3. *reaktiv*: Der Agent reagiert auf Umweltreize.
4. *proaktiv*: Der Agent verfügt über zielorientiertes Verhalten und zeigt Initiative.

Von stärkeren Definitionen wird meistens zusätzlich die Umsetzung von „menschlichen“ Konzepten gefordert. Damit sind im allgemeinen die sog. mentalen Zustände („mental states“), also beliefs (Überzeugungen, Annahmen), commitments (Verpflichtungen), decisions, intentions, goals etc. gemeint. Bekannte Beispiele hierfür sind die sog. intentionalen Systeme („intentional systems“) oder BDI-Agenten. In diesem Text werden Agenten nach der schwachen Definition begriffen. Eine Übersicht über die Eigenschaften und Modellierungsverfahren findet sich im Anhang „Agentenorientierte Programmierung“ auf Seite 156.

Auf der Seite der Modellarchitektur wird hier, wie von Minsky vorgeschlagen, von einer Hierarchie von relativ einfachen Einheiten ausgegangen, die zusammen (und nur zusammen) eine komplexe Aufgabe bewältigen. Dabei sollen die Agenten einer Hierarchieebene ähnlich komplex sein und können wiederum als Menge von Agenten modelliert sein. Dabei ist eine sehr große Spannweite möglich, von kleinsten Operatoren bis zu planenden und handelnden Systemen. Durch das Kommunikationsparadigma der Agenten, ergibt sich die Schnittstellenbeschreibung automatisch aus dem Agentendesign, genauer gesagt aus der Nachrichtendefinition.

Nicht nur das Agentensystem, sondern auch viele andere Funktionen, die von mehreren Programmen benutzt werden, sind als gemeinsam benutzte Bibliotheken realisiert, was zudem noch eine effiziente Speichernutzung zur Folge hat. Diese gemeinsam benutzte Bibliotheken können für verschiedene Betriebssystemvarianten verschieden implementiert sein. Der hier verfolgte hierarchische Ansatz realisiert allgemeine gemeinsam benutzte Bibliotheken, die auf systemabhängige gemeinsam benutzte Bibliotheken aufbauen. So läßt sich der Portierungsaufwand minimalisieren.

In den folgenden Abschnitten werden die Schichten, soweit sinnvoll, einzelnen behandelt und Realisationsalternativen perspektivisch besprochen. Danach wird eine Architektur für ein Gestenerkennungssystem vorgeschlagen werden.

6.2 Schicht 1: Datenaufnahme

Um die Bewegungsdaten eines Menschen aufzunehmen, können verschiedene Arten von Sensoren benutzt werden. Dabei werden verschiedene Ansätze verfolgt: Passive, in denen ein Bild mittels Licht, Infrarot oder Ultraschall aufgenommen wird. Aktive, bei denen aktiv Wellen emittiert und empfangen werden. Aufgrund von Doppler oder ähnlichen Effekten kann dann ein Rückschluß auf die Position und Raumlage des Sensors vorgenommen werden. Dabei kommen elektromagnetische und Ultraschallwellen zum Einsatz. Nicht zuletzt existieren Exoskelett-Ansätze, in denen, wie in der Goniometrie, außen am Körper fixierte Biege- oder Winkelsensoren benutzt werden und somit die Gelenke des Skelettes an der Körperoberfläche nachzubilden versuchen.

Diese Ansätze haben wiederum verschiedene Eigenschaften. Diese Betrachtung verkompliziert sich dadurch, daß es sich im allgemeinen um komplexe Meßverfahren und -apparate handelt. Diese Verfahren sind, unter anderem, durch die folgenden Parameter gekennzeichnet:

INVASIVITÄT. In welchem Maße beeinflußt das Meßverfahren den Probanden? Mediziner benutzen unter anderem zur genauen Bestimmung von Bewegungsparametern in die jeweiligen Knochen geschraubte Gestänge, die mit den entsprechenden Skalen versehen sind. Dies ist die wohl invasivste benutzte Methode. An der anderen Seite der Skala stehen z.B. Verfahren der kamerabasierten Bildverarbeitung, wenn sie ohne Markierungen auf dem Probanden auskommen.

GENAUIGKEIT. Mit welchem Meßfehler muß gerechnet werden? Neben der absoluten und relativen Größe des Fehlers lassen sich auch Fehlerursachen auf verschiedenen

Ebenen ausmachen, wie z.B. Befestigung am Probanden, Umgebungseinflüsse und dem Meßverfahren inhärente Probleme.

GESCHWINDIGKEIT. Wann, nach dem Ereignis in der realen Welt, liegen die Daten zur Verarbeitung vor? Einige Verfahren eignen sich nur zur offline-Auswertung — also zu einer Auswertung im Nachhinein und sind u.U. sogar nicht praktikabel automatisierbar.

In der traditionellen Biometrie werden, insbesondere in der Ergonomie, gerne photometrische Verfahren benutzt [Win79]. Dabei werden z.B. Photographien von mit kleinen Lämpchen bestückten Versuchspersonen unter Langzeitbelichtung erstellt. So wird meistens vor einem Hintergrund mit bekannten Referenzpunkten aus zwei um 90° gedrehten Kamerapositionen aufgenommen. Aus den sich ergebenden Leuchtspuren der Lämpchen vor dem Referenzhintergrund läßt sich die Trajektorie des Lämpchens in der Kamaraebene rekonstruieren. Mit diesen beiden Projektionen läßt sich nun eine dreidimensionale Trajektorie schätzen. Statt der Lämpchen können auch Reflektoren bei geeigneter Beleuchtung zum Einsatz kommen. Ein ähnliches Verfahren arbeitet mit Film. Hier kann über die Bildnummer auch Information über Ableitungen des Ortes gewonnen werden, da eine Zuordnung des Ereignisses an einen Zeitpunkt, relativ zum Filmbeginn, möglich ist, da die Aufnahmefrequenz der Filmkamera konstant und bekannt ist. Bei der Langzeitbelichtung wird auch mit Stoboskopbeleuchtung oder blinkenden Lämpchen bekannter Frequenz gearbeitet, was einen ähnlichen Effekt wie Film hat, nur daß hier die Abfolge der Posturen nur durch weiteres Wissen über den Versuchsablauf rekonstruiert werden kann. Solches Wissen wäre z.B., daß ein jedes Objekt sich kontinuierlich bewegt und es sinnvoll ist, gewisse Obergrenzen für die zu beobachtenden Geschwindigkeiten anzunehmen. Invasivität und Genauigkeit sind bei diesen Verfahren sicherlich hinreichend, doch in Hinsicht auf die Geschwindigkeit ist es offensichtlich, daß es sich um offline-Verfahren handelt.

Videokamerabasierte Bildverarbeitungsverfahren sind aktueller Forschungsgegenstand und erfordern meist noch entweder spezielle Markierungen oder besondere Beleuchtung [HG98]. Ein fortschrittliches Verfahren präsentiert Nölker in [NR98b] als „Visual Body Sensor“ (VisBoS). Dabei werden zwei stereoskopische 192×144 Punkte Bilder mit je 5 Hz verwendet, um Schulter-, Ellenbogen- und Handwurzelposition eines an einer bestimmten, 2,5 m von den Kameras entfernten Stelle stehenden Benutzers mittels neuroinformatischer Verfahren zu ermitteln. Ein prinzipielles Problem dieser Verfahren ist die geringe Auflösung der Kameras, so daß es entweder erforderlich ist, mit einer bewegten Kamera den Probanden zu verfolgen

[VRR98][BBB+98][TM98] oder die Hände werden nur in einer bestimmten Armposition erkannt [NR98], die die Kamera erfassen kann. Allerdings können einfache Verfahren, die sich an oben beschriebenen photometrischen Verfahren anlehnen und z.B. die Leuchtspuren verfolgen und so auf Positionen rückschließen, für spezielle Anwendungen akzeptable Ergebnisse liefern. Da aber Videokameras eine feste Abtastfrequenz und — wie erwähnt — eine geringere Auflösung haben, kommen auch diese Verfahren zu hohen Ungenauigkeiten, obwohl die algorithmische Aufarbeitung einfacher ist.

Eine weitere Klasse von Sensoren arbeitet mit pulsierenden magnetischen Feldern. Ihr größter Nachteil liegt in der Tatsache, daß die Position des Gerätes, oder zumindest die seiner Antenne, bestimmt wird. Diese Antenne (Sensor) ist bei einem Anbieter $2,54\text{ cm}$ mal $2,54\text{ cm}$ mal $2,03\text{ cm}$ groß und mittels einem mehradrigen Kabel mit der Auswertungseinheit verbunden, die wiederum dann mittels einer seriellen Verbindung an den die Meßdaten weiterverarbeitenden Rechner angeschlossen ist. Dies zwingt einen dazu, die Versuchsperson mit Vorrichtungen zum Befestigen der Sensoren auszustatten und eine Kabelführung¹⁾ zu konstruieren, die einem Verheddern des Kabelbaumes entgegen wirkt. Ihr größter Vorteil ist die große Genauigkeit in den 6 gemessenen Freiheitsgraden sowie die hohe Abtastfrequenz von ca. 144 Hz (20 Hz alternativer Anbieter). In den drei Freiheitsgraden der Translation kann mit einer mittleren Standardabweichung von $0,254\text{ cm}$ über dem Meßbereich und in den drei Freiheitsgraden der Rotation mit einer mittleren Standardabweichung von $0,5^\circ$ ($0,1^\circ$ alternativer Anbieter) über dem Meßbereich nach Anwendung der Entzerrungsfunktion gerechnet werden²⁾. In Tabelle 12 sind die einzelnen Verfahren noch einmal zusammenfassend dargestellt. Allgemein läßt sich folgendes zu den Sensorarten (aktiv, passiv, exoskelett) sagen:

PASSIV. Diese Verfahren haben eine geringe zeitliche wie örtliche Auflösung und erfordern zusätzlichen Rechneaufwand (Bildverarbeitung) im Hauptrechner. Sie sind per se 2D-Verfahren, die die 3. Dimension nur durch stereoskopische Berechnungen erfassen können. Sie gehören zu den am wenigsten invasiven Verfahren. Ihr größtes

-
1. Seit kurzer Zeit existiert auch ein Produkt, das die Übermittlung der Daten vom Körper zu einer Zentraleinheit kabellos vornimmt. Die einzelnen Sensoren sind aber nachwievor durch Kabel mit einem kleinen Rucksack verbunden, der dann die Übertragung vornimmt.
 2. Dies sind Spezifikationen nach [Asc96] wobei die englische Abkürzung RMS für „root-mean-square“ benutzt wurde, die entweder als Standardabweichung oder aber als $\sqrt{\langle x^2 \rangle}$ aufzufassen ist.

Verfahren	Freiheitsgrade	Invasivität	Präzision	Geschw.
<i>Photometrie</i>	Pos. 2D + Trajektorie	Marker	rel. hoch	offline
<i>Video</i>	Pos. 2D	max. Marker	gering	max. 60 Hz
<i>Stereo-Video</i>	3D - Verdeckung	max. Marker	gering	max. 60 Hz / 5 Hz typisch
<i>Elektro-magnetisch</i>	Pos. 3D + Rot. 3D = 6 DOF	Sensoren mit Kabeln	3 mm bzw. 0,5°	20 Hz – 144 Hz

Tabelle 12: Vergleich der Sensortypen (Körper)

Problem ist, daß Körperteile durch andere Körperteile verdeckt sein können (*Verdeckungsproblem*).

AKTIV. Aktive Verfahren haben eine gute zeitliche wie örtliche Auflösung (zumindest bei elektromagnetischen Wellen) in 3 bis 6 Freiheitsgraden („degrees of freedom“ DOF). Der Berechnungsaufwand ist typischerweise in eigener, vom Hauptrechner unabhängiger, Hardware gelöst, da Signallaufzeiten usw. gemessen werden müssen. Sie haben eine mittlere Invasivität, da die kleinen Empfänger/Antennen und ihre Kabel am Körper befestigt werden müssen. Ihr größtes Problem ist eben diese Befestigung bei Wahrung eines angemessenen Invasivitätsgrades (*Befestigungsproblem*, engl. „mounting problem“).

EXOSKELETT. Diese Verfahren haben zwar eine nur durch die mechanische Genauigkeit und Abtastrate des Analog-Digital-Konverters (DAC) beschränkte (also sehr hohe) zeitliche Auflösung, liefern aber keinerlei Information im Ortsraum. Dafür können sie eine hohe Auflösung im Konfigurations-(Winkel-)raum haben. Auch hier besteht das Befestigungsproblem.

Datenhandschuhe sind die am häufigsten anzutreffenden Exoskelettsensoren. Sie haben typischerweise 18 – 22 Parameter und eine Winkelauflösung von teilweise unter einem Grad. Für eine erschöpfende Untersuchung des heutzutage am häufigsten anzutreffenden Modells — dem CyberGlove CG1801 — siehe [KHW95].

6.3 Schicht 2a: Vorverarbeitung

Die Vorverarbeitung dient der Konsolidierung der Daten aus technischer Sicht. Dabei sollen insbesondere statistische Fehler und systematisch korrigierbare Fehler der Meßapparatur ausgeglichen werden, bzw. die Größe dieser Fehler erkannt werden, sodass die weitere Verarbeitung mit den so erhaltenen Vertrauensintervallen operieren kann.

In dieser Schicht wird auch sichergestellt, daß alle Meßwerte in der richtigen Reihenfolge gemäß ihres Zeitstempels ausgewertet werden. Auch sind Plausibilitätsbetrachtungen sinnvoll, um vereinzelte „Ausreißer“ in den Meßwerten isolieren zu können.

6.4 Schicht 2b: Körpermodell

In den vorherigen beiden Abschnitten wurde dargestellt, wie schwierig es ist, Positions- oder Winkeldaten vom Körper eines Probanden zu erhalten. Keines dieser Verfahren liefert eine holistische Sicht auf den Probanden, sondern immer eine Meßdatenauswahl mit der sich erhofft wird, die Bewegung beschreiben zu können. Um nun a priori Wissen über den menschliche Körper in die Problembearbeitung mit einzubringen, können Erkenntnisse aus Anthropometrie, Anatomie und Physiologie verwendet werden. In diesen Wissenschaften ist das Skelett und das Muskelsystem untersucht worden und es stehen Erkenntnisse über Geometrie und die Arbeitsbereiche der einzelnen Körperteile zur Verfügung. In Abbildung 31 ist eine Zeichnung des Knochen- und Knorpelaufbaus des Schultergürtels zu sehen.

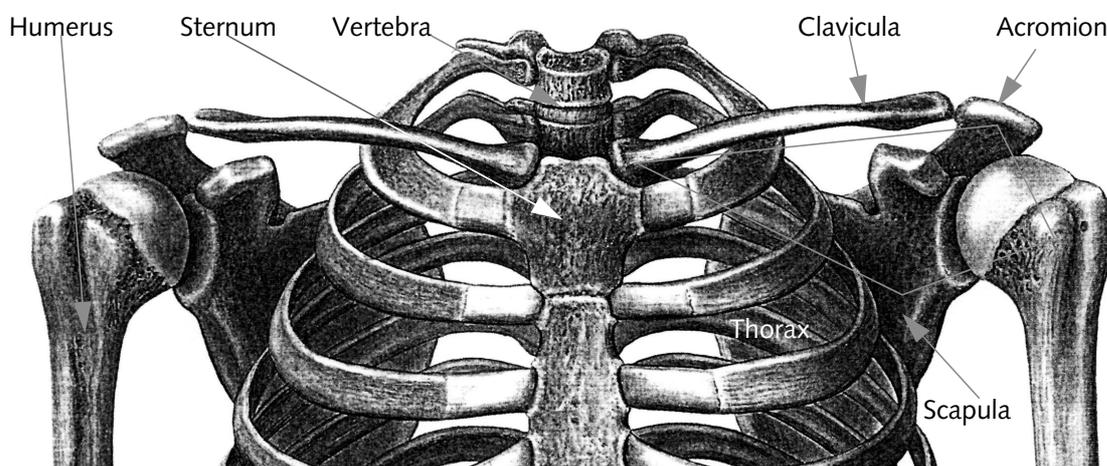


Abbildung 31: Das menschliche Skelett: Schultergürtel aus [RY94]

Auf den ersten Blick wird klar, das es sich dabei nicht um ein einfaches System weniger Gelenke und Segmenten handelt, wie es üblicherweise in technischen Systemen betrachtet wird. Zu beachten ist, daß ein Schultergelenk sozusagen „frei“ an drei Punkten aufgehängt ist: Einerseits ist es mittelbar über die Clavicula am Sternum (ventrale Befestigung) und direkt über die Scapula am Thorax befestigt (dorsale Befestigung). Abbildung 32 zeigt die große Beweglichkeit dieses Apparates, der nicht nur zur Abduktion/Adduktion des Armes sondern auch zur Elevation und Depression der Schulter fähig ist.

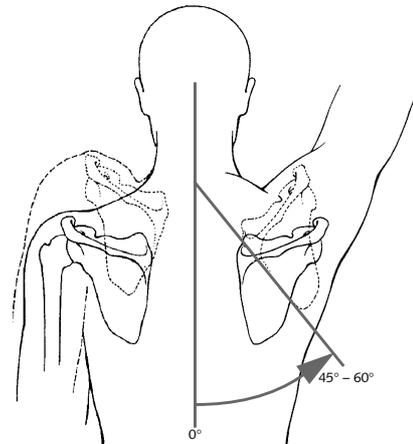


Abbildung 32: Bewegung des Schulterblattes nach [RY94]

Das Schultergelenk kann unabhängig von der Armstellung angehoben werden oder, wie auch in Abbildung 33 zu sehen, in Folge der fortgesetzten Abduktionsbewegung des Armes angehoben werden. In Abbildung 33b ist der Punkt markiert, an dem das Acromion den Humerus blockiert und somit den gesamten Schulterapparat bestehend aus Scapula und Clavicula zwingt die Abduktionsbewegung durchzuführen. Durch diese Einschränkung ist eine Kopplung der Abduktion mit der Elevation gegeben, die in einem Modell z.B. durch geeignete Beschränkungen für den Abduktionswinkelbereich darstellbar sind. Im weiteren ist auch eine Modellierung durch eine abhängige Gewichtsfunktion vorstellbar, deren Berechnung unter Umständen jedoch aufwendig sein kann.

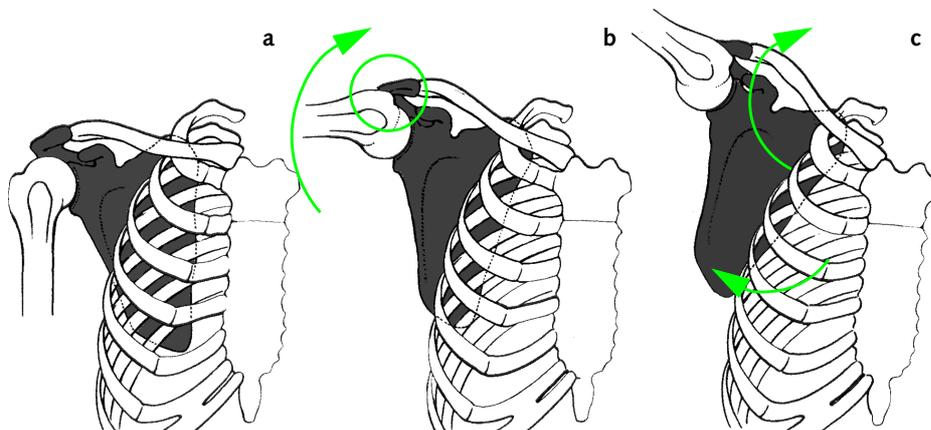


Abbildung 33: Thoraxial- und Armskelett bei Elevation nach [RY94]

In Abbildung 34 ist Hand und Arm in Supinationstellung¹⁾ skizziert. Neben den verschiedenen Bezeichnern für die einzelnen Körperteile sind auch noch einige wichtige Achsen verzeichnet:

- In der Flexionsachse wird die Ellenbeuge (Cubitus) bewegt.
- Der Unterarm dreht sich in der Pronations-/Supinationsachse.
- Die Hand wird in den Palmarflexions-/Dorsalextensionsachsen ein- und ausgeklappt.
- Das Abwinkeln der Hand zu den Seiten hin erfolgt in der Ulnar-/Radialabduktionsachse.

Wie in Abbildung 34 zu sehen ist, stehen diese Achsen weder Senkrecht noch in Vielfachen von 90° aufeinander, noch laufen die inneren Achsen mittig durch die Segmente. Auch schneiden sich die n . und die $(n+1)$. Achse nicht notwendigerweise. Das Ellenbogengelenk ist vergleichsweise einfach, hat es doch eine nur einen Freiheitsgrad mit einer festen Drehachse.

Hellgrau ist in Abbildung 34 die sog. Luntanumsäule in der Mittelhand dargestellt, in deren mittlerem Segment (Os capitatum) die Ulnar-/Radialabduktionsachse der Hand liegt. Der dunkelgrau dargestellte Mittelhandknochen (Os scaphoideum) dreht sich bei einer Abduktion in einer ulnar-radialen Achse. Da die Dorsalextension hauptsächlich in der distalen, die Palmarflexion aber hauptsächlich in der

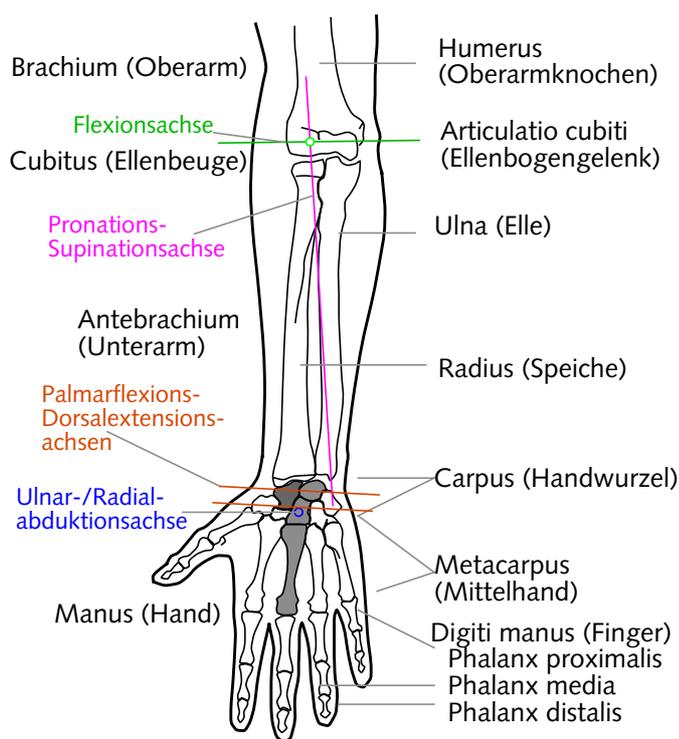


Abbildung 34: Supinationsstellung der Hand

1. Supinationstellung: Bei hängendem Arm zeigt die Handfläche nach vorne und der Daumen nach außen. Ulna und Radius kreuzen sich *nicht*.

proximalen Handwurzelreihe auftritt, wird klar, daß bei gleichzeitiger Abduktion und Extension oder Flexion der Hand sich die Achsen verlagern und zu einem komplexen Arbeitsraum der Hand führen.

In der Tabelle 13 sind exemplarisch für das Schultergelenk einige, den Arbeitsbereich kennzeichnende, Winkel aufgeführt. Diese Werte wurden ermittelt, um „Normal“-werte für die klinische Untersuchung zu haben und damit pathologische Situation diagnostizieren zu können.

	allein	mit Schlüsselbeingelenk
<i>Anteversion</i>	90°	170°
<i>Retroversion</i>	30° — 40°	40°
<i>Abduktion</i>	90°	180°
<i>Adduktion</i>	30°	40°
<i>Innenrotation</i>	70°	100°
<i>Außenrotation</i>	60°	90°

Tabelle 13: Bewegungsumfänge des Schulterbereichs aus [Ben94]

Neben diesen Eigenschaften des Skelettes sind aber auch einige Eigenschaften des Muskel- und Sehnenapparates erwähnenswert. Zum einen die Tatsache, daß die Finger von Muskeln des oberen Unterarmes gesteuert werden und zum anderen, daß Muskel bei Anspannung eine bestimmte, feste Form annehmen. Auch wenn letzteres trivial erscheint, so hat gerade dies eine Auswirkung auf die Sensorpositionierung. Wenn ein Sensor (oder auch Marker für ein Videoverfahren) über einem Muskel platziert wird, so wird er bei Anspannung des Muskels bewegt, auch wenn sich keine Änderung der Geometrie des Gesamtsystems dadurch ergibt. Auch in der Frage der Invasivität ist dies zu beachten, da ein umgeschnallter Sensor, z.B. den Armdurchmesser beschränken kann, was bei einer Muskelkontraktion dann zu Irritationen führt und den Sensor aus seiner Position rutschen lassen kann (Befestigungsproblem).

6.4.1 Modellsicht

Die oben dargelegten anatomischen Erkenntnisse müssen nun in eine Modellsicht formalisiert werden. Eine direkte Umsetzung der anatomischen Gegebenheiten ist zwar möglich, führt aber schon in der Schulter schnell zu mehr als 30 Freiheitsgraden [Kap84], was für die hier betrachteten Zwecke viel zu detailliert ist.

Körpermodelle wurden z.B. von Lenarcic und Umek [LU94] (siehe Abbildung 35) und auch Ko ([Ko94] pp. 84-86, vgl. Abbildung 36 auf Seite 87) vorgeschlagen, wobei ersteres sehr abstrakt aber auch einfach ist und das zweite sehr detailliert und noch dafür relativ kompliziert ist.

Aus diesen Vorbildern wurde ein eigenes Modell entwickelt, das hier das *geometrische Körpermodell* genannt wird (siehe Abbildung 37 auf Seite 88). Auf der linken Seite sind die Gelenke mit den Namen der entsprechenden anthropometrischen, meßbaren Gelenkparameter verzeichnet, auf der rechten Seite sind die einzelnen Segmente mit den Namen der bekanntesten der beteiligten Knochen sowie die weiterhin benutzte Segment- und Winkelbezeichnung (L_1-L_3 und $\alpha/\beta_1-\alpha/\beta_3$) aufgeführt. Im weiteren sind die designierten Befestigungspunkte für die realen 6 DoF Positionssensoren durch ihre Koordinatensysteme eingetragen.

Damit ein solches Körpermodell technisch nutzbar wird, müssen einige Bedingungen erfüllt werden:

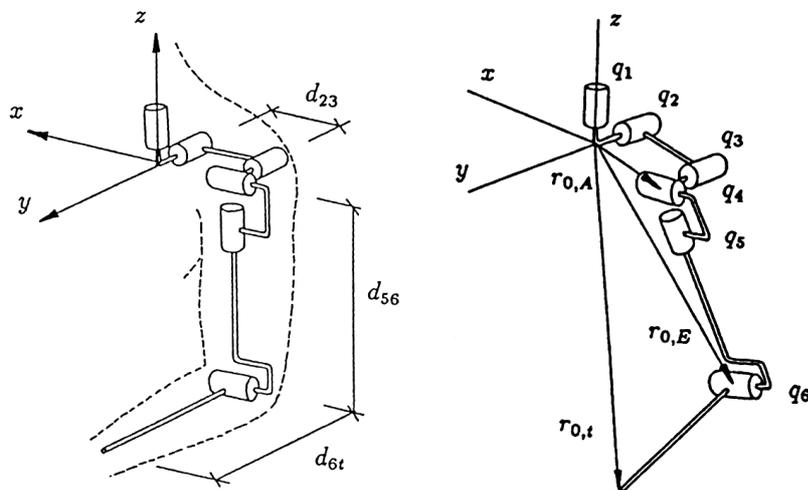


Abbildung 35: Armmodell aus [LU94]

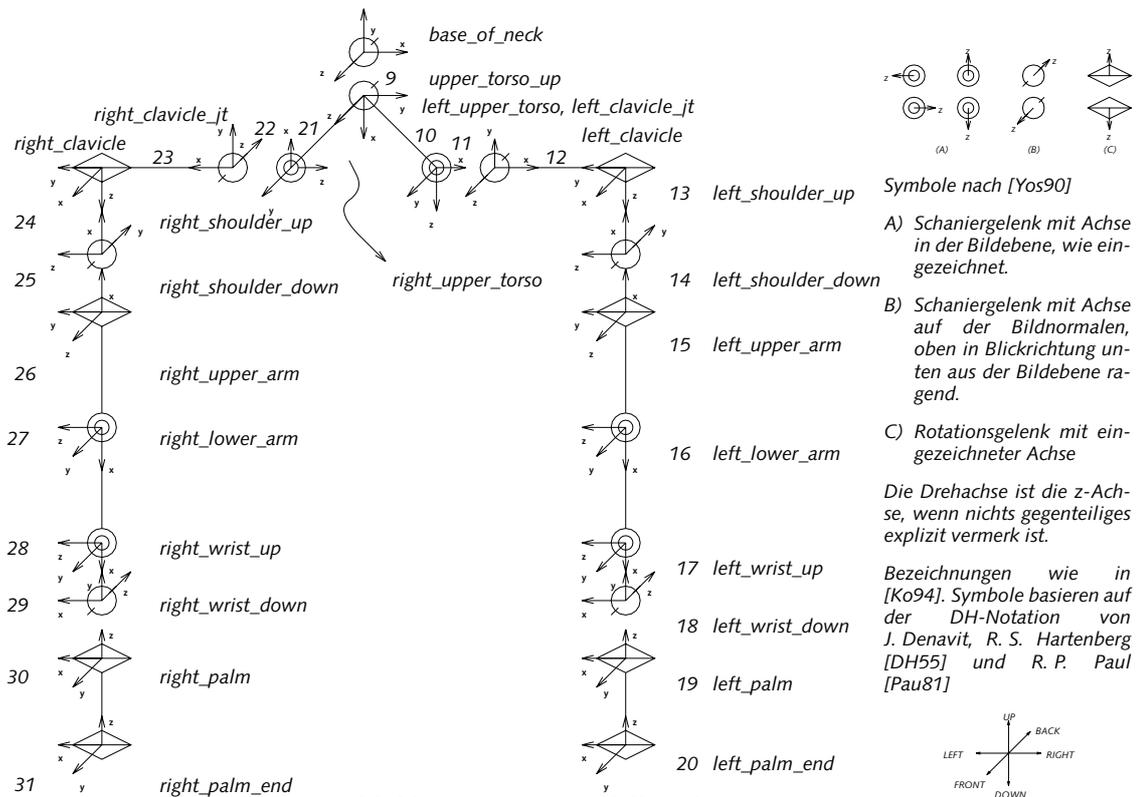
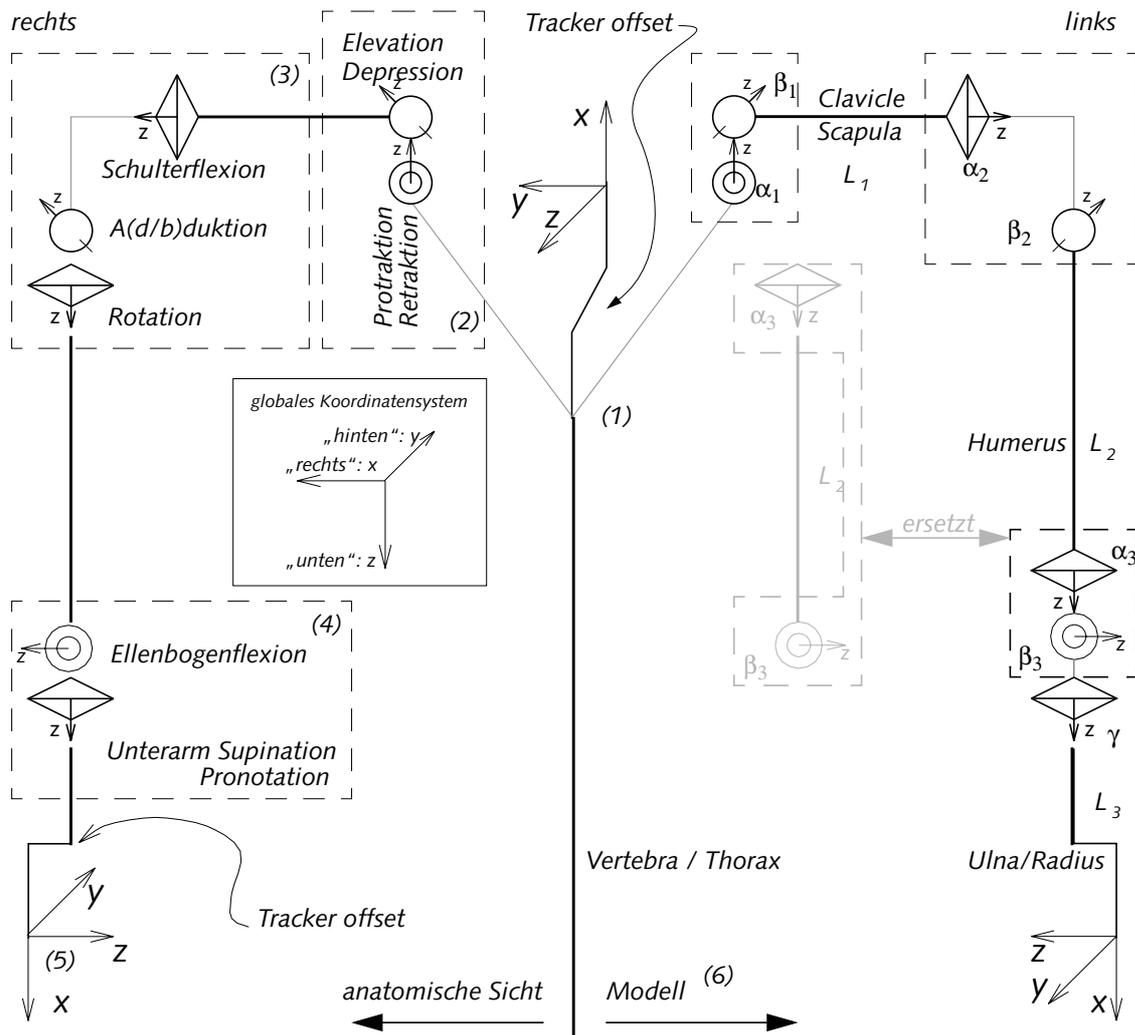


Abbildung 36: Armmodell nach [Ko94]

Ein solches Modell sollte eine Implementierung ermöglichen, die „anytime“-fähig ist. „Anytime“-fähig bezeichnet dabei die Eigenschaft, daß der implementierte Algorithmus zu jeder Zeit eine mehr oder weniger gute oder spezifische Lösung präsentieren kann. Im weiteren wird gefordert, daß mit fortschreitender Zeit die Lösung besser werden soll. Auf den vorliegenden Fall bezogen heißt das, daß das Körpermodell zu jedem Zeitpunkt eine sinnvolle und den physischen Beschränkungen eines Menschen entsprechende Lösung liefern soll und je nach verfügbarer Rechenzeit, diese der tatsächlichen Postur des Probanden in der realen Welt möglichst gut entsprechen soll. Dieser die Bewegung beschreibende Teil soll daher im folgenden *dynamisches Körpermodell* genannt werden.

Ein solches Modell sollte robust gegen kurzzeitige Fehler in den Meßdaten (Glitches) sein. Zur Robustheit gehört auch die Eigenschaft, daß auch Punkte außerhalb des Konfigurationsraumes sinnvoll behandelt werden. Z.B. wird der Ellenbogen 2 m hinter dem Körper angegeben, so ist dies physiologisch quasi unmöglich, trotzdem ist es



- 1) Schnittpunkt Wirbelsäule, Schlüsselbeine. Referenzpunkt für die Größenbestimmung, daher ist dort auch ein 6 DoF Positionssensor vorgesehen
- 2) Schulter-Pseudogelenk mit drei modell. Freiheitsgraden
- 3) Schulter-Wirbelsäulen-Pseudogelenk mit zwei modellierten Freiheitsgraden
- 4) Ellenbogen-Pseudogelenk, mit zwei modellierten Freiheitsgraden. Die Supination und Pronotation des Unterarmes wird der Einfachheit halber in als ein Freiheitsgrad in diesem Punkt modelliert.

- 5) Position eines der 6 DoF Positionssensoren am Handgelenk

- 6) Diese Seite wird im Abschnitt „Anwendung des MMC-Netzes auf das Körpermodell“ auf Seite 116 erklärt.

Nur bezeichnete Segmente haben eine von Null verschiedene Länge. Die grauen Linien bezeichnen Pseudosegmente der Länge Null. Die Symbole sind, wie schon bei Abbildung 36, nach [Yos90] gewählt, und auch hier wird versucht möglichst nahe an der DH-Notation [DH55][Pau81] zu bleiben.

Abbildung 37: Geometrisches Körpermodell nach [FW98]

wünschenswert, wenn der Ellenbogen bis an die Grenzen des Konfigurationsraumes geführt wird und die Diskrepanz zur Zielpostur für nachfolgende Fehlerbehandlungsschritte vermerkt wird, ohne zu undefinierten Systemzuständen zu kommen.

Ebenso muß ein solches Modell das vorwärts- sowie das rückwärtskinematische Problem lösen können, damit die Positionen, an denen die tatsächlichen Sensoren angebracht werden können, freier bestimmt werden können. Auch sollten unterbestimmte Fälle sinnvoll behandelt werden, also einerseits zu physiologisch möglichen und andererseits aus dem Kontext erschlossenen, sinnvollen Positionen führen.

6.4.2 Bekannte Methoden

Klassische Methoden sind unter diesen Bedingungen nur kompliziert zu implementieren. Diese Methoden, ob direkt oder iterativ, haben spezifische Vor- und Nachteile. Die direkte Lösung (algebraisch oder geometrisch bei [KB82]), die sich aus der Lösung eines System aus n Gleichungen mit n Unbekannten herleitet, führt nicht immer zu einer geschlossenen Form. Um dieses Verfahren sinnvoll anzuwenden, muß also ein geometrisches Modell vorliegen, daß eine geschlossene Form als Lösung hat. In den meisten Fällen wird so ein Lösungsraum beschrieben, in dem eine günstige Lösung ausgewählt werden muß. Günstig in dem Sinne, daß die Konfigurationsdifferenz in allen Winkeln gering ist.

Andere, iterative Verfahren benutzen eine Jacobi-Matrix (siehe [Pau81] oder [Yos90]) und deren (Pseudo-) Inverse. Solche Verfahren sind langsamer und konvergieren zu nur einer Lösung und betrachten auch diverse Ableitungen der Bewegung. Allerdings ist die Jacobi-Matrix weder trivial aufgebaut noch deren Inverse einfach zu berechnen. Ist sie singular oder nicht quadratisch wird sich mit diversen Hilfs- und Näherungskonstrukten, wie z.B. Pseudoinversen und vieler numerischer Verfahren beholfen. Trotzdem können die Lösungen degeneriert sein, d.h. es treten Gelenkgeschwindigkeiten auf, die nicht zur Affektorgeschwindigkeit beitragen.

Meist läßt sich ein gegebenes invers-kinematisches Problem so nur durch größere Einschränkungen an den Anwendungsbereich oder durch das Einbringen von applikationsbezogenem Modellwissen lösen. Hierzu kommt noch, daß die anfallenden Berechnungen teilweise sehr aufwendig sind, was einen Einsatz unter Echtzeitbedingungen unter Umständen vereitelt, da in fast jedem Teilschritt aufwendige Ausnahmebehandlungen erforderlich sind. Die Anytime-Fähigkeit läßt sich bei iterativen Verfahren vielleicht noch herleiten, die direkten Verfahren erzeugen jedoch aus einer Konfiguration eine Nachfolgekongfiguration ohne gültige Zwischenschritte.

6.4.3 Existierende Systeme

Neben den zuvor beschriebenen Methoden, existieren noch diverse Systeme, die Körpermodelle enthalten. Viele dieser Systeme sind für den Einsatz in der Ergonomie bestimmt, einige von ihnen arbeiten offline, also ohne direkte Rückkopplung zum Benutzer und fallen somit aus dem Skopus. Ein weiteres Feld findet sich bei den computergraphischen Systemen, doch liegt dort die Betonung auf einem guten Äußeren und nicht auf der Abbildung der Realität. Beispielsweise werden die Augen vergrößert und das Verhältnis von Rumpf zu Extremitäten dahingehend verändert, daß die Extremitäten verlängert werden — ein Zugeständnis an unser derzeitiges Schönheitsideal.

Exemplarisch für die verbliebenen Systeme wird hier nun das durch [BPW93] und weitere Veröffentlichungen sehr bekannte und kommerziell erhältliche *Jack*-System betrachtet. *Jack* ist eine Entwicklung des Center for Human Modeling and Simulation der University of Pennsylvania (Moore School of Engineering) und stellt eine vollständige virtuelle Umgebung dar, die mit anthropometrisch exakt parametrisierbaren Figuren anreicherbar ist, und mittels C- und Lisp-APIs oder direkter Positionssensorenanbindung beeinflußt werden kann. Die Umgebung ist physikalisch modelliert und auch jede virtuelle Figur liefert Simulationsdaten ihrer 39 Segmente und 38 Gelenke (mit zusammen 88 Freiheitsgraden) bezüglich ihrer Geometrie und zusätzlich bezüglich einwirkender Kräfte bzw. Drehmomente. Das *Jack*-System bietet die Möglichkeit, andere Systeme einzubetten, doch das genaue Gegenteil ist im hier betrachteten Rahmen gefragt. *Jack* ist zudem nicht im Quellcode verfügbar und kann somit nicht an variierende Hardwaregegebenheiten angepaßt werden.

6.4.4 Abstraktion

Ein Körpermodell liefert die Möglichkeit nicht nur Wissen in die Datenaufnahme einzubringen, sondern auch eine Abstraktion von den realen Sensoren vorzunehmen, und es für die weitere Entwicklung des Systems zu ermöglichen, an quasi beliebigen Stellen sog. „virtuelle“ Sensoren zu „befestigen“, also Positions- und Orientierungsdaten beliebiger Körpersegmente abfragen zu können. Ein solches Modell sollte anhand der Daten der Positionssensoren ein möglichst wirklichkeitsgetreues Abbild der Körperkonfiguration des Probanden liefern. Da mit z.B. drei Sensoren die Konfiguration von zwei Armen aber immer unterbestimmt ist, müssen weitere Wissensquellen benutzt werden, um mittels einiger Heuristiken möglichst plausible Zustandsübergänge zu finden.

6.4.5 Diskussion

Wenn ein Körpermodell benutzt wird, ergibt sich als auffälligster Nachteil für den Fall, daß reale und virtuelle Sensoren nicht übereinstimmen, daß der ermittelte Wert auf einer Heuristik beruht, also je nach Güte der Heuristik und den Umständen ein mehr oder minder fehlerbehafteter Wert ist. Andererseits wird Unabhängigkeit von der Position der Sensoren am Probanden, idealerweise sogar eine gewisse Unabhängigkeit von der Anzahl der Sensoren, gewonnen. Letzterenfalls sind Anzahl der Sensoren und Güte des Modells gekoppelt. Als obere Schranke ist trivialerweise die Anzahl der Freiheitsgrade des Modells zu nennen, denn wenn jeder Parameter gemessen wird, werden keine Heuristiken benötigt und der Fehler des Modells ergibt sich direkt aus dem Fehler für die einzelnen Sensoren. Für nicht entartete Fälle, ergibt sich eine untere Schranke von drei 6 DOF Sensoren für ein zweiarmiges Modell, wie in Abschnitt „Mean of Multiple Computations Network (MMC-Netz)“ auf Seite 110 noch begründet werden wird. Dabei sind dann Fehler bei der Ellenbogen- und Schulterposition zu erwarten.

6.5 Schicht 2c: Merkmalserkenner

Auf dem Körpermodell, das sich, wie in Abschnitt „Schicht 2b: Körpermodell“ auf Seite 82 dargestellt, aus *geometrischer* und *dynamischer* Komponente zusammensetzt, können nun weitere Verarbeitungsschritte aufsetzen. Als erster Schritt steht dabei die Codierung ausgewählter Merkmale in eine interne symbolische Repräsentation an. Gute interne Repräsentationen, wie schon im Abschnitt „Repräsentation“ auf Seite 24 eingeführt, stellen die in dem jeweiligen Kontext wichtigen Dinge, Eigenschaften oder Relationen explizit dar. Die in diesem Zusammenhang natürlichen Randbedingungen der Domäne werden herausgestellt, um Berechnungen zu vereinfachen und weniger relevante Details in den Hintergrund zu stellen, um übersichtlich zu bleiben. Im weiteren sind sie vollständig, kompakt, transparent, schnell bezüglich des Zugriffs und berechenbar. Besonders hervorzuheben sind dabei die transparente Darstellung, d.h. die repräsentierte Information bleibt für den Benutzer (Systemdesigner) gut verständlich, und die vollständige Darstellbarkeit der Effekte und Objekte.

6.5.1 Klassifikation

Allgemein betrachtet stellt sich in dieser Schicht eine Klassifikationsaufgabe von den Daten der (virtuellen) Sensoren zu Merkmalsklassen. Diese Merkmalsklassen sind ein Notationssystem das in der Betrachtungsgranularitätsstufe der Gesten noch keine Bedeutung hat und dessen Zeichen deshalb auch als Merkmale bezeichnet werden. Je-

doch haben diese Zeichen sehrwohl eine Bedeutung auf der Betrachtungsgranularitätsstufe der Gestenform oder Gestenstruktur, sind also auf dieser Ebene Symbole.

Die Aufgabe der Merkmalerkennung ist es ordinale Werte auf geeignete nominale Skalenteile abzubilden. Die durch diese nominale Skalen gegebenen Zeichen sollten den Anforderungen an eine Repräsentation genügen. Gibt es endlich viele dieser Skalen, läßt sich das Lexikon einfach aufzählen. Eine Zuordnung zu den betrachteten Effekten und Objekten ist hier über die Sensoren direkt gegeben, d.h. diese Zeichen sind kategoriale Repräsentationen der ikonischen Sensordaten, also durch die Sensordaten begründet. Wird dann noch eine Grammatik über diesen Zeichen definiert, ist auch die Anforderung nach einer Struktur erfüllt. Diese Anforderung, ebenso wie die Forderung nach einem prozeduralen Teil müssen hingegen in einer höheren Schicht verankert werden.

Um die Werte auf die nominalen Skalen abzubilden bieten sich verschiedenste Verfahren an. Es kann mit künstlichen neuronalen Netzen oder statistischer Klassifikation [Nie83] die Abbildung mit einer gewissen Fehlerrate erlernt werden oder sie kann mit sog. „Fuzzy Logic“ [KY95] modelliert werden. Auch eine direkte Klassifikation durch eine surjektive Abbildung ist möglich sofern das Wissen vorliegt, wie diese Abbildung zu formen ist. Welches Verfahren auch immer angewendet wird, die daraus entnommene Information wird die Realität immer nur unsicher und vage beschreiben.

UNSICHERES UND VAGES WISSEN.

„Insoweit sich die mathematischen Gesetze auf die Realität beziehen, sind sie nicht exakt. Und soweit sie exakt sind, beziehen sie sich nicht auf die Realität.“ Albert Einstein, 1921

Die klassische mathematische Klassifizierung von Aussagen als „wahr“ oder „falsch“, wie z.B. Bool sie in seiner Logik vorstellte, stößt bei vielen Anwendungen an ihre Grenzen als adäquate Repräsentation. Zu repräsentierendes Wissen kann z.B. *unsicher* sein, in dem Sinne, daß nicht vollständig gewährleistet ist, daß es den Weltzustand korrekt abbildet. Diese Unsicherheit rührt u.a. von nicht perfekten Sensoren, unzuverlässigen Informanten oder partiellen Regeln her. Zudem läßt sich noch eine andere Dimension des Problems ausmachen: In vielen Domänen sind die Konzepte, mit deren Hilfe Wissen formuliert werden kann, nicht scharf gegeneinander abgegrenzt. Sie werden als *vage* bezeichnet [Dav90b].

Das Wissen in der Domäne multimodale Interaktion ist sowohl unsicher wie auch vage. Jedes Sensordatum ist, je nach Qualität des Sensors, mit Fehlern behaftet (siehe Abschnitt „Schicht 1: Datenaufnahme“ auf Seite 78) und auch wenn die Form einer Geste richtig erkannt wurde, ist fraglich, ob die Regeln zur weiteren Analyse vollständig (und richtig) sind. Auch sind multimodale Äußerungen wie auch Gesten, wie in Abschnitt „Gesten aus Sicht des Rezipienten“ auf Seite 13 ausgeführt, keine scharfen Konzepte. So ist der Übergang von z.B. einer deiktischen Geste, einem Taktschlag und einer nicht semiotischen Geste (z.B. Handbewegung zum Beibehalten des Gleichgewichtes) fließend.

Obwohl Unsicherheit und Vagheit voneinander verschieden sind, tragen sie doch letztendlich zu einem gemeinsamen Fehlerphänomen bei. Dieser zu erwartende Fehler wird durch einen Vertrauenswert, der sog. *Konfidenz*, repräsentiert.

Definition: Hypothese: Im folgenden wird für Symbole, die mit einem Konfidenzwert versehen sind, der Begriff *Hypothese* verwendet.

6.5.2 Diskussion

Wesentlich für diese Schicht ist die Auswahl geeigneter ordinaler Variablen und die Abbildung auf geeignete nominale Skalen deren Teile als Symbole des ersten Grades anzusehen sind. Dabei muß ein gewisses Fehlerphänomen berücksichtigt werden, das auf vager und unsicherer Information beruht. Welche Kriterien nun über eine geeignete Realisation entscheiden hängt stark von der Realisation der Schicht darunter und den Anforderungen der nächsten Schicht ab. Daher wird diese Schicht noch ausführlicher in dem Kapitel „Vorschlag einer Realisation des Gestenerkenners“ auf Seite 101 dargestellt werden.

6.6 Schicht 3: Integration

In dem vorhergehenden Abschnitt „Schicht 2c: Merkmalerkennung“ auf Seite 91 wurde ein System vorgeschlagen, das Sensordaten in einfache Symbole wandelt. In der Integrationsschicht liegt die zu bearbeitende Information also in Form von Symbolen vor. Die symbolische Repräsentation ermöglicht eine einheitliche Darstellung von Information. Aufgrund dieser Vereinheitlichung ist es in dieser Schicht möglich, dieselben Methoden zur syntaktischen Symbolmanipulation für alle Symbole anzuwenden, völlig unabhängig davon, was sie bedeuten.

Wie schon im Abschnitt „Systemgrundlagen“ auf Seite 71 ausgeführt, ist eine innere Untergliederung der Integrationsebene von einfacheren zu komplexeren Systemen

Der umgangssprachlich formulierte Regelsatz kann in fünf Vorbedingungen $H_1 \dots H_5$ und eine Nachbedingung \bar{H} untergliedert werden, wie die Klammern anzeigen. Die prinzipielle Struktur einer Regel lautet:

$$\bar{H} : H_1, H_2, \dots, H_n \Rightarrow \text{assert}(f(H_1, H_2, \dots, H_n)), \quad (\text{Gl. 1})$$

wobei die H_i diejenigen Hypothesenklassen darstellen, zu denen Instanzen h_i im Gedächtnis vorhanden sein müssen, damit die Konsequenz der Regel (hinter dem \Rightarrow) ausgeführt wird. Der *assert*-Operator fügt seinen Operanden dem Gedächtnis hinzu. Die Funktion f beinhaltet die Randbedingung, der zwischen den h_i erfüllt sein soll. f liefert als Ergebnis eine neue Instanz h der Hypothesenklasse H . In den bisher betrachteten Beispielen modellierte die Funktion f eine einfache Randbedingung, z.B. Gleichzeitigkeit. Der umgangssprachliche Regelsatz weist hingegen eine kompliziertere Struktur auf, die sich formalisiert wie folgt liest:

$$\bar{H} : H_1, H_2, H_3, H_4, H_5 \Rightarrow \text{assert}(f(H_1, H_2, H_3, H_4, H_5)) \quad (\text{Gl. 2})$$

Offenbar muß in der, die Randbedingung beschreibenden Funktion f eine Prüfung auf Gleich- und Nachzeitigkeit durchgeführt werden. Anstatt die Funktion f als geschlossenen Term zu formulieren, läßt sie sich hier einfacher als Schachtelung von $g := \text{GLEICHZEITIG}$ und $n := \text{NACHZEITIG}$ auffassen:

$$f(H_1, H_2, H_3, H_4, H_5) = g(H_1, H_2, H_3, n(H_4, H_5)) \quad (\text{Gl. 3})$$

Schließlich können beide Funktionen zur Überprüfung der Randbedingung in zwei verschiedenen Regeln untergebracht werden:

$$\bar{H} : H_1, H_2, H_3, H_4, H_5 \Rightarrow \text{assert}(f(H_1, H_2, H_3, H')) \quad (\text{Gl. 4})$$

$$H' : H_4, H_5 \Rightarrow \text{assert}(n(H_4, H_5)) \quad (\text{Gl. 5})$$

Wie in dem Beispiel-Regelsatz schon zum Ausdruck kommt, ist ein Phänomen, daß bei normalen Parsern nicht berücksichtigt wird, im Kontext der Gestenerkennung von großer Wichtigkeit: Zeit. Wie in dem Kapitel „Theoretische Grundlagen“ auf Seite 11 dargelegt wurde, existiert a priori Wissen über die zeitlichen Randbedingungen von Gestik und Sprache. Auch zeitliche Bedingungen lassen sich, wie im Beispiel, in Regelsätzen, z.B. unter Zuhilfenahme von Zeitkalkülen, realisieren.

Neben der Formulierung der komplexen Randbedingung als Kombination von einfacheren Randbedingungen hat diese Aufspaltung einen anderen Vorteil: Es können Teilergebnisse in Form von neuen Hypothesen (hier H') berechnet werden, die später

in anderen Funktionen Verwendung finden. Ist die Regel durch eine komplexe Randbedingung beschrieben (Gleichung 2), dann muß für jede neue Basishypothese H_i der Gesamtausdruck neu berechnet werden. Die Aufspaltung einer komplexen Regel in Teilregeln kann durch das System durchgeführt werden, indem es die Definition einer neuer Hypothese als Kombination von anderen Hypothesen mit einer Menge von Basisfunktionen erlaubt, um die Definition dann selbständig in eine Menge von Regeln zu verwandeln, analog zum Schritt von (Gleichung 3) nach (Gleichung 4) und (Gleichung 5). Dieses wird im Weiteren als Regel-Konstruktor bezeichnet.

Die Tatsache das verschiedene Symbole aus verschiedenen Quellen mit jeweils eigener Verankerung in der Zeit integriert werden müssen, führt zu zwei nicht-trivialen Problemen:

Definition: Korrespondenzproblem: Wie sind die Querbezüge zwischen den einzelnen Quellen zu determinieren, vgl. [Sri95]? Dies Problem beschäftigt sich also sozusagen mit der Integration in der „Breite“ (siehe Abbildung 38 auf Seite 94).

Definition: Segmentierungsproblem: Wie sind die Grenzen zwischen Einheiten in der Zeit zu finden? Hier tritt die neue Dimension — die Zeit — in Erscheinung. Analog zu dem Korrespondenzproblem könnte dies als Integration in der „Tiefe“ bezeichnet werden.

Da Gesten Element der menschlichen Kommunikation sind, also von Menschen produziert und perzipiert werden, ist es gestattet, die Zeiteinteilung aus dem Kapitel „Zeitbegriffe“ auf Seite 31 auf die Gesten anzuwenden. Dabei wird klar, daß je einfacher die Elemente werden, desto kürzer die Zeitfenster sind, in denen die Elemente ausgewertet werden. Das kann technisch dadurch dargestellt werden, daß in den einzelnen Verarbeitungsstufen die Information nur über einen bestimmten Zeitpunkt vorgehalten wird — der Speicher also ein gleitendes Zeitfenster realisiert. Dieses Verhalten kann einfach dadurch erzeugt werden, das Information die älter als ein bestimmter Schwellwert ist, gelöscht wird. Auf diese Art und Weise wird der Suchraum für die einzelnen Integratoren und deren Regeln drastisch eingeschränkt ohne Information zu verlieren.

Als wichtiges Mittel zur Wissensorganisation wird hier eine Klassenhierarchie benutzt (siehe Abschnitt „Probleme der Taxonomie und Systematik“ auf Seite 19). In ihr existieren zwischen Begriffen Spezialisierungs- (bzw. Verallgemeinerungs-) Relationen, die eine azyklischen Graphen bilden. Ist ein Begriff eine Spezialisierung eines anderen, dann treffen für ihn alle Eigenschaften zu, die auch für den Oberbegriff gültig sind.

Dies erlaubt es, Aussagen über große Gruppen von Begriffen zu treffen, und somit allgemeines Wissen leichter auszudrücken.

Der Ansatz zur symbolischen Integration basiert auf der Annahme, daß das Integrationswissen in Form von Regeln über Hypothesen mit einer Randbedingung („constraint“) dargestellt werden kann. Programmsysteme, die solche Regelwerke effizient abarbeiten können, werden Produktionssysteme genannt. Sie bieten den Vorteil, das Regel-Wissen (Integrationswissen) und Fakten-Wissen (Hypothesen) getrennt voneinander deklariert wird und dann in einem Ausführungsschritt aus den bekannten Fakten mit dem angegebenen Regeln neue Fakten gefolgert werden [Gia94]. Wird dieses Verfahren mit Minskys Modell verglichen so ist eine Regel der kleinste modellierte Agent in diesem System.

Diese Regeln können zur Laufzeit des Systems geändert werden, und erlauben es so sich zur Laufzeit an die jeweiligen Anforderungen einzustellen. Dazu können Meta-Regeln definiert werden, die auf jeder Stufe aus den benötigten und erhaltenen Daten Schlußfolgerungen ziehen und so Einfluß auf den Prozeß nehmen können. Letztendlich stellt ein solches System die Infrastruktur für ein Expertensystem zur Verfügung. Es ist also prinzipiell möglich, auch sehr komplizierte Integratoren zu definieren. Auf diese Weise könnten sogar lernende Systeme implementiert werden.

DISKUSSION. Ein solches regelbasiertes System ist durchaus in der Lage die Produktionen einer Grammatik nachzuvollziehen. Es stellt einen Rahmen dar, der sehr große Freiheit in der Art der Implementation ermöglicht. Der größte Nachteil ist, daß ein solches System zwar das Potential hat sich dynamisch auf die für es wesentlichen Ereignisse zu beschränken, aber nicht die algorithmische Effizienz hat, wie ein Parser einer a priori definierten Grammatik. Im weiteren ist die Modellierung der Zeit zwar durch Regelsätze möglich, aber nicht effizient. Wie die Problematik der effizienten Behandlung der Zeit trotzdem, unter Zuhilfenahme der im Kapitel „Zeitbegriffe“ auf Seite 31 gesammelten Erkenntnisse angegangen werden kann, wird im nächsten Kapitel behandelt.

6.7 Architektur eines Gestenerkennungssystems

Wie zu Anfang dieses Kapitels angekündigt, wird nun eine Systemarchitektur auf der Basis der in diesem Kapitel erarbeiteten Konzeption vorgestellt. Sie ist in Abbildung 39 skizziert. Die am unteren Rand dargestellten Sensoren liefern Daten, die in verschiedenen Schritten zu immer komplexeren Symbolen verarbeitet werden. Daten, die die Konfiguration des Körpers betreffen, werden in der linken Seite behandelt, die Daten,

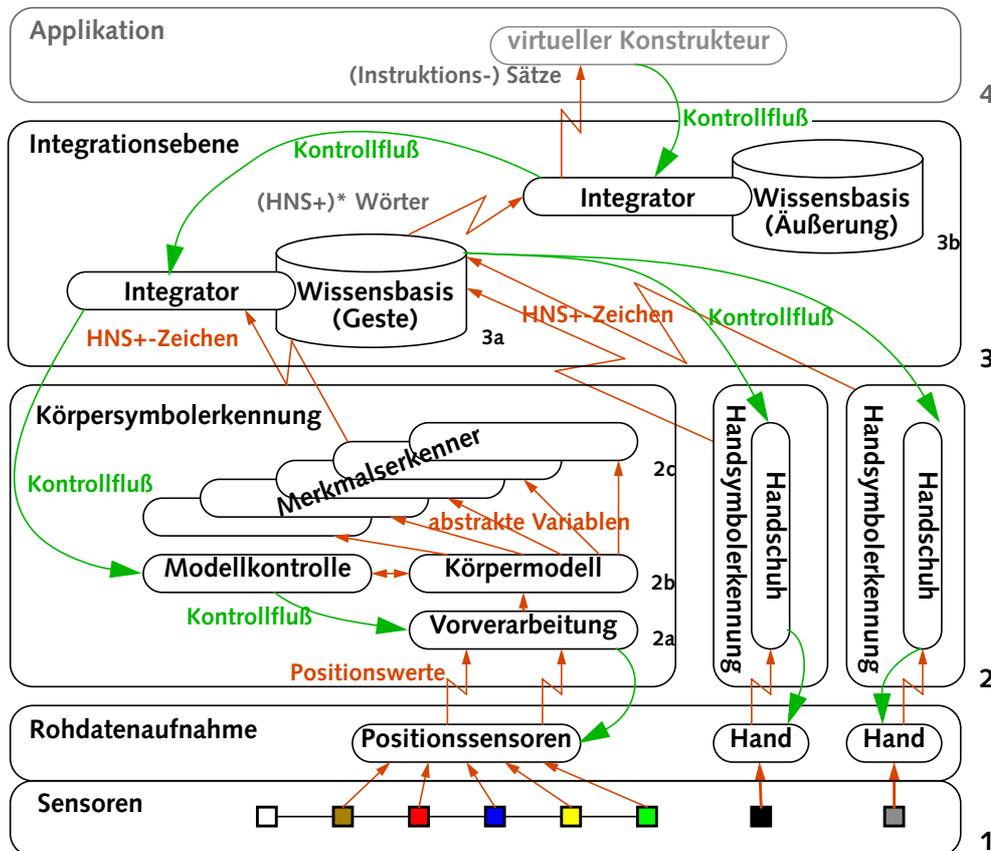


Abbildung 39: Vorschlag einer Systemarchitektur

die die Konfiguration der Hände betreffen, werden auf der rechten Seite behandelt. Wie zu sehen ist ergibt sich eine ausgezeichnete Verarbeitungsrichtung für immer komplexere Symbole, die wiederum aus Symbolen aus weniger komplexeren Ebenen zusammengesetzt sind, beziehungsweise sich direkt als Meßwerte von physikalischen Vorgängen ableiten.

Wie in Abbildung 40 dargelegt, ergibt sich so eine Hierarchie der einzelnen Symbolklassen. Die unterste Schicht wird von den Meßwerten geliefert, die von den Sensoren am Rumpf und den Händen empfangen werden. Die nächste Schicht stellt eine Generalisierung dar, in der nur für die folgende Verarbeitung bemerkenswerte Eigenschaften in einem Symbol aus der Symbolmenge, die hier einmal HNS+ genannt wird, codiert wird. In der nächsten Schicht werden diese Einzelzeichen mit der hier spezifi-

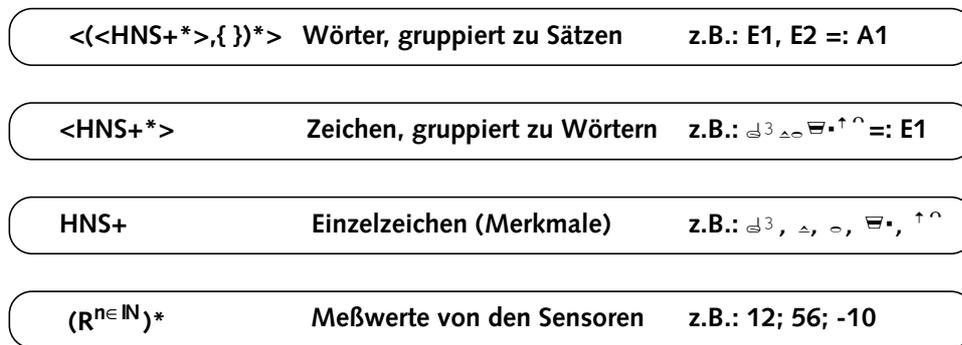


Abbildung 40: Hierarchie der Symbolklassen

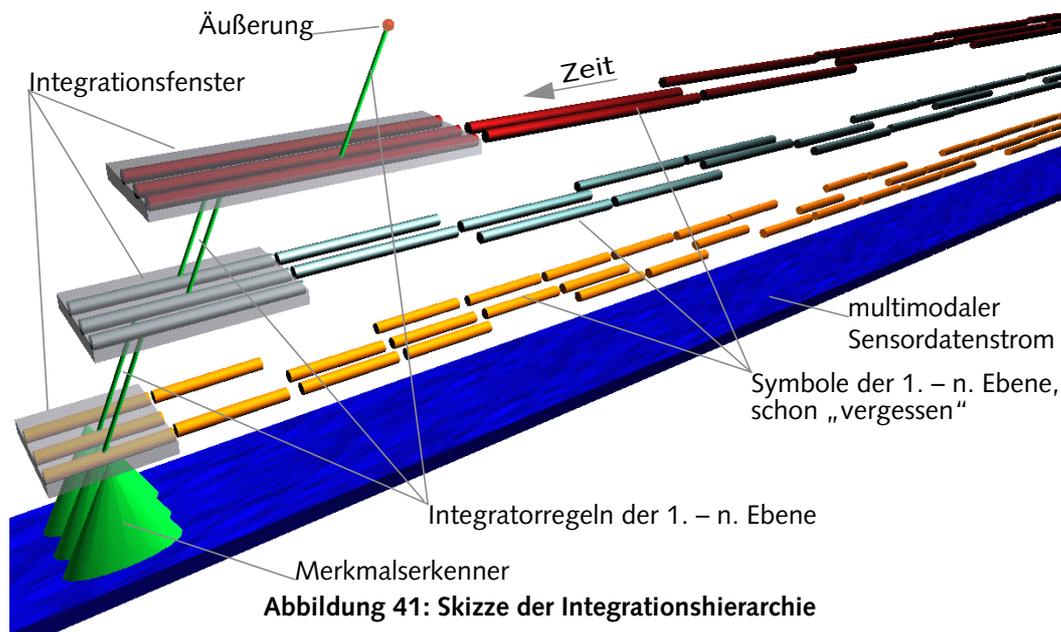
zierten Methode zur temporalen Integration zu neuen Symbolen integriert. Dieser Vorgang ist vergleichbar mit dem Zusammenfügen von Buchstaben zu einem Wort. Da hier nur ein endliches Vokabular verarbeitet wird, kann dieses Wort wieder als ein Symbol aus einem Symbolvorrat angesehen werden. Auf der anderen Seite fließen Kontrollinformationen von den oberen Schichten nach unten. Darin wird jeweils aufgabenorientiert spezifiziert oder ausgewählt, welche Information die höhere Schicht von der untereren Schicht benötigt, um ihre Funktion ausführen zu können. Auf diese Art wird auch sichergestellt, daß eine möglichst ereignisorientierte Abarbeitung zustandekommt; also keine Information produziert wird, die dann gar nicht ausgewertet wird.

6.8 Zusammenfassung

Es wird vorgeschlagen den Gestenerkennung als ein hierarchisch gemäß der Datenkomplexität organisiertes Multiagenten-System aufzufassen. Dabei werden die Signale, Merkmale und Symbole von unten nach oben weitergereicht und dabei eine quantitative Datenreduktion zu qualitativ höherwertigen Hypothesen durchgeführt. Die Anforderungsnachrichten werden hingegen von oben nach unten weitergeleitet und organisieren die Funktion des Agentensystems. Es werden Möglichkeiten der Bewegungsmessung zur Datenaufnahme verglichen und ein von den konkreten Sensoren abstrahierendes Körpermodell vorgeschlagen. Die Information dieser abstrahierten, virtuellen Sensoren wird dann von Merkmalerkennern auf ein Symbolsystem abgebildet. Dabei werden die Symbole mit einem Konfidenzwert annotiert und Hypothesen genannt. Der Integrationsrahmen bietet eine Möglichkeit zeitbasiert Hypothesen zu komplexeren Hypothesen zu integrieren. Im ganzen führt dies zu einer dreidimen-

sionalen Erweiterung der Abbildung 38 auf Seite 94, wobei die dritte, neu hinzu gekommene Dimension die Zeit ist. Dies ist in Abbildung 41 veranschaulicht. Dabei stehen die grauen Kästen für die Gedächtnisfenster der Integratoren, der blaue Balken für die Menge der kontinuierlichen Signale und die dünnen grünen Linien und Kegel für Merkmalerkennungs- oder Integrationsprozesse und die kleine Kugel als Ergebnis der Integration.

Wie die in diesem Kapitel vorgestellten Verfahren realisiert werden können, wird im nächsten Kapitel dargelegt. Dabei wird auch konkretisiert, wie jeweiligen Merkmale und Symbolsysteme definiert werden können. Daraufhin wird gezeigt werden, wie sich mit dem so erstellten System Gestenerkennungsszenarien bearbeiten lassen.



Kapitel 7

Vorschlag einer Realisation des Gestenerkenners

Im vorherigen Kapitel „Konzeption eines Gestenerkenners“ auf Seite 71 wurden die konzeptionellen und architektonischen Grundlagen erörtert. In diesem Kapitel wird nun vorgeschlagen, wie die einzelnen Bestandteile realisiert zu realisieren sind, beziehungsweise welche Lösungen zu den Problemen gefunden wurden, die sich aus der Konzeption ergeben.

7.1 Systemgrundlagen

Wie in der Architektur dargelegt, basiert die Realisation auf einem mehrschichtigen Modell und den Konzepten „Agenten“, „Parallelverarbeitung“ und „gemeinsam benutzte Bibliotheken“. Im weiteren soll nun vorgestellt werden, wie diese Konzepte ausgefüllt sind.

7.1.1 Agentensystem

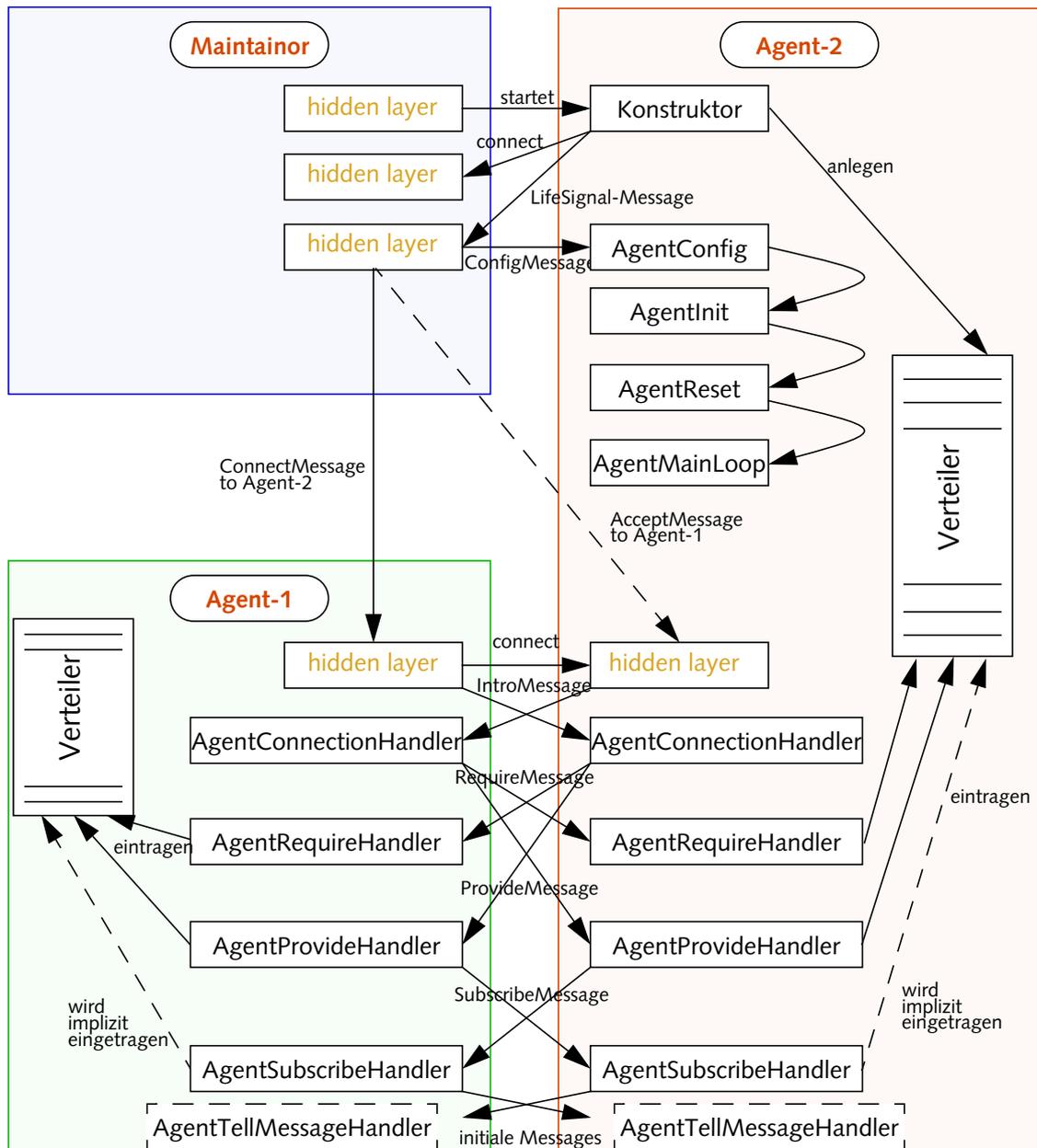
Im Abschnitt „Agenten“ auf Seite 77 wurden schon verschiedenen Agentenkonzepte vorgestellt. Wenn es hingegen um die konkrete Systemimplementierung geht, wird der Agentenbegriff auf autonome Programmeinheiten, die durch Nachrichten kommunizieren eingeschränkt. Dabei verwendet das Agentensystem das Paradigma der agentenbasierten Programmierung [Sho93]. Dieses Paradigma vermeidet ein großes Problem der populären verteilten objektorientierten Programmierung (engl. „*Distributed Object-Oriented Programming*“: DOOP). Da in der objektorientierten Sicht Objekte als eigenständige Entitäten mit klaren Schnittstellen dargestellt werden, hat sich DOOP dazu hinreißen lassen eine „Verteilungstransparenz“ zu postulieren, also eine Ebene auf der entfernte und lokale Objekte gleich anzusprechen sind. Obwohl Verteilungstransparenz schnell modelliert und einfach gefordert werden kann, kann sie nur sehr schwer und nur mit großen Nachteilen realisiert werden. Neben den mannigfaltigen Sicherheits- und somit auch Zuverlässigkeitsproblemen ergeben sich auch große Effizienzprobleme, da Netzwerktransport weder per se zuverlässig noch, verglichen mit lokalen Operationen, schnell ist. Auch ist es nicht absehbar, wann ein entferntes Objekt aktualisiert wird und wie lange das dauert. Es ist also per se nicht möglich allen Prozessen eine konsistente Sicht zu ermöglichen, die die Verteilungstransparenz ja gerade fordert (vgl. [GF99]).

Agentenbasierte Programmierung stellt den handelnden Agenten in den Vordergrund, der Information mit anderen Agenten und seiner Umwelt kommuniziert. In der Inter-Agentenkommunikation sollten die Information strukturiert sein, wie z.B. durch die „Knowledge Query and Manipulation Language“ (KQML) [KSI93]. Dabei werden die Nachrichten in sog. Performative (Frage, Mitteilung, Hypothese) unterteilt, die den Zweck der Nachricht darstellen.

Das in diesem Vorschlag präferierte Agentensystem ist von der Arbeitsgruppe „Wissensbasierte Systeme“ der Technischen Fakultät der Universität Bielefeld entwickelt und in Form von C++-Klassen implementiert worden. Ein Agent ist ein Objekt einer Agentenklasse, dessen Methoden Aktionen und Reaktionen auf die verschiedenen Performative codieren. Nachrichten sind ebenfalls Objekte, hier von Nachrichtenklassen, deren Methoden die Performative zum Versenden des Inhalts des jeweiligen Objekts und u.U. von Oberklassen ererbte Verfahren zur technischen Kodierung dieser Information enthält¹⁾. Theoretisch könnte so ein Programm mehrere Agenten enthalten, doch ist dies bei der Abbildung der Programmierkonzepte auf die Konzepte des Betriebssystems hinderlich und ist deshalb nicht vorgesehen. Die einzelnen Programme können Prozesse auf verschiedenen Rechnern einer administrativen Domäne haben. Bislang ist dieses System für Solaris 2.4 bis 7 und IRIX 5 bis 6.5 implementiert worden. Verwaltet wird dieses Mehrprogramm-*Multiagentensystem* (MAS) [Mue93] durch einen Verwaltungsprozeß, der die Agenten initial bekannt macht und sicher stellt, daß mehrere MAS sich nicht behindern. Abbildung 42 auf Seite 103 zeigt den Ablauf von Kommunikation in einem solchen Szenario.

Ein wichtiges Konzept stellen dabei die Verteiler und die Dienste dar. Jeder Agent kann an einen bestimmten anderen Agenten Nachrichten versenden oder aber an eine Gruppe von Agenten, die in einem sog. Verteiler eingetragen sind. Ein Agent kann sich mittels des „subscribe“-Performatives in den Verteiler eines anderen Agenten eintragen und mit „unsubscribe“ wieder austragen. Dienste sind Entitäten auf der Ebene des Systemdesigns. Ein Agent kann bestimmte Dienste, wie z.B. eine bestimmte Berechnung, entweder anbieten („provide“), oder kundtun, daß er ohne die Hilfe eines anderen Agenten nicht auskommen kann („require“). Typischerweise wird der Nachrichtenfluß für spezifische Dienste über ihnen zugeordnete Verteiler aufgebaut. Alle Informationen, die auf Grund eines bestimmten Dienstes kommuniziert werden, werden über den jeweiligen Verteiler abgewickelt. Die Struktur der Dienste und Ver-

1. Es wird also nicht das Objekt verschickt, sondern sein Inhalt. Es wird keine Verteilungstransparenz postuliert!



Alle Agenten sind über objektorientiert als Instanzen von Agentenklassen implementiert. Dabei sind einige Funktionen dem Agentenprogrammierer verborgen. Andere Methoden kann er überladen und so die Funktionalität implementieren. Für den proaktiven Teil steht eine Hauptschleife zur Verfügung. Mittels der (Message) Handler besteht die Möglichkeit auf Nachrichten zu reagieren.

Abbildung 42: Das WBS Multiagentensystem

teiler kann hierarchisch modelliert werden, da das Verteiler-Konzept objektorientiert via Klassenhierarchie implementiert ist.

Wenn auch die Nachrichtensemantik durch die KQML-Nachrichtentypen festgelegt ist, stellt dies noch keine Aussage über den Inhalt einer Nachricht dar, der prinzipiell beliebig kodiert sein kann.

7.1.2 Parallelität

Um effizient arbeiten zu können, mußte Abschied genommen werden vom traditionellen Verarbeitungsmodell, das ein Programm als eine Art Operator über einer Eingabemenge ansieht und eine Ausgabemenge liefert. Auch das Verarbeitungsmodell, das ereignisgetrieben arbeitet, muß erweitert werden, da Agenten auch immer eine proaktive Komponente haben sollen, also Ereignisse aus eigener „Überlegung“ anstoßen können sollen.

Damit in einem Programm mehrere Programmflüsse effizient realisiert werden können, wird die sog. Thread-Technologie (siehe [KSS96]) angewendet. Ein eigenständiger Programmfluß wird im User-Level des Betriebssystems, also dem Teil, über den der Benutzer Kontrolle hat, mit einem sog. Thread realisiert. Die dafür nötigen Routinen stehen in einer Benutzerbibliothek zur Verfügung. In diesem Projekt finden ausschließlich die im Rahmen des „Portable Operating System Interface (POSIX 1003.1/1996)“ ([IEEE1003.1], insbesondere Abschnitte 11, 13, 16, 17 und 18), genormten Varianten Verwendung¹). Es ist aber bis zu einem gewissen Grade den Implementatoren dieser Bibliotheken freigestellt, wie sie die geforderte Funktionalität realisieren. Das größte Problem für die Implementatoren besteht darin, die Flußverwaltung, das sog. „scheduling“, effizient zu realisieren. Die Verwaltung von Peripheriegeräten, Speicher und Rechenzeit die vornehmste Aufgabe des Betriebssystems, deshalb sollten solche Vorgänge soweit irgend möglich auf Konzepte und Verfahren des Betriebssystems abgebildet werden.

Eines der fortschrittlichsten Konzepte am Markt zu diesem Thema hat die Fa. SunSoft, Inc., eine Tochter der Fa. Sun Microsystems, Inc. Da das SunOS sog. „light weight processes“ (LWPs), auch „kernel threads“ genannt, im System-Level bietet, können Threads nach Bedarf oder automatisch auf LWPs abgebildet werden, die dann unter das „hierarchical priority scheduling“ des Betriebssystems fallen. Diese wiederum wer-

1. Dabei werden die von dem Betriebssystemherstellern bereitgestellten Bibliotheken verwendet, alternativ ständen noch unabhängige Distributionen, z.B. vom MIT zur Verfügung.

den automatisch auf mehrere Prozessoren verteilt, sofern der ausführende Rechner über mehrere Prozessoren in „symmetric multi processing“, SMP-Technologie (siehe z.B. [DS98]) verfügt. Das könnten heutzutage bis ca. 128 Prozessoren des Typs ULTRAsparc sein.

Wie die Problematik des parallelen Zugriffes auf Speichereinheiten vermieden wird, ist in Anhang „Thread-sichere Programmierung“ auf Seite 167 dargelegt.

7.1.3 Gemeinsam benutzte Bibliotheken

Um eine hohe Effizienz bei der Speichernutzung und einen gut wartbaren Quellcode zu erhalten werden systematisch „(dynamic) shared objects“, gemeinsam benutzte Funktionsbibliotheken, verwendet. Sie werden in den meisten Betriebssystemen nur einmalig in den Speicher geladen und doch von verschiedenen Programmen benutzt, dabei wird nur das Text-Segment gemeinsam benutzt, nicht aber die Datensegmente, so daß die einzelnen Programme nach wie vor unabhängig von einander bleiben.

Bibliotheken, die in Programmen verwendet werden, die die Thread-Technologie benutzen, müssen „thread safe“ sein. Dies bedeutet, daß in den Routinen keine Operationen durchgeführt oder Programmierkonzepte angewendet werden dürfen, die entweder durch das Thread-Programmierparadigma verboten sind oder aber zu ungeschützten kritischen Abschnitten führen. Ein typisches Problem ist die Verwendung von Zeigern auf statischen Variablen einer Funktion als Rückgabewert, die von mehreren Threads, z.B. leicht zeitversetzt, benutzt wird. In dem Thread, der die Funktion zuerst benutzte, tritt dann der Effekt auf, daß sich der Inhalt dieses Zeigers unversehens ändert. Der Entwurf von Systemen deterministischen Verhaltens wird damit sehr erschwert, daher wird ein solches Konstrukt als Fehler gewertet. Eine weitere Eigenschaft, die Bibliotheken aufweisen können, wird von Fa. Sun Microsystems, Inc. als *MT-Hot* bezeichnet. Dabei handelt es sich um Bibliotheken, die Routinen zur Verfügung stellen, die intern — für den Benutzer opaque — Thread-Technologie benutzen, ohne daß die sie benutzende Anwendung notwendiger Weise Threads benutzt. In diesem Sinne sind die Agentenbibliotheken als *MT-Hot* zu kennzeichnen.

7.2 Datenaufnahme

Wie in der Konzeption dargelegt, finden sich in der Literatur verschiedenste Methoden anthropometrische Daten aufzunehmen, wobei sich herausstellte, daß einige davon für online-Verarbeitung nicht tauglich sind.

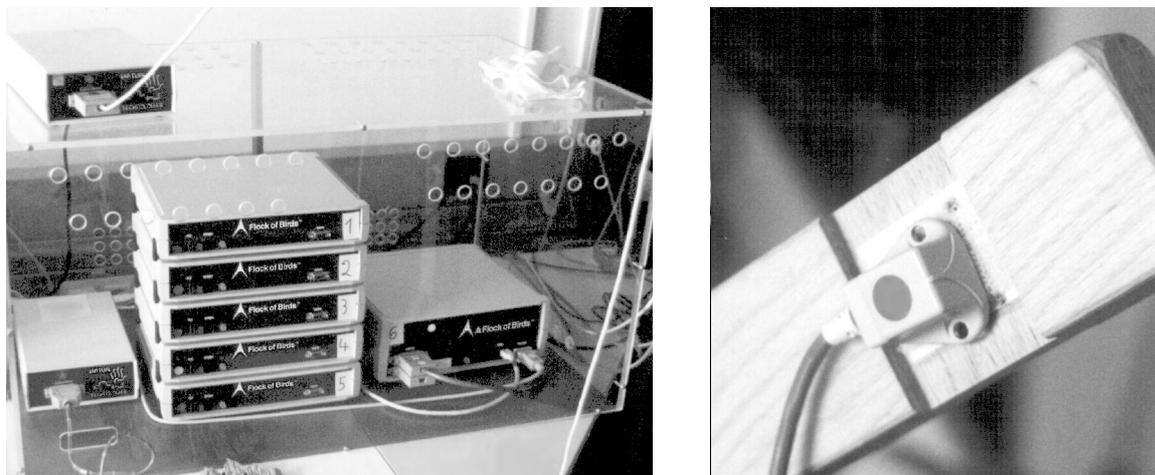
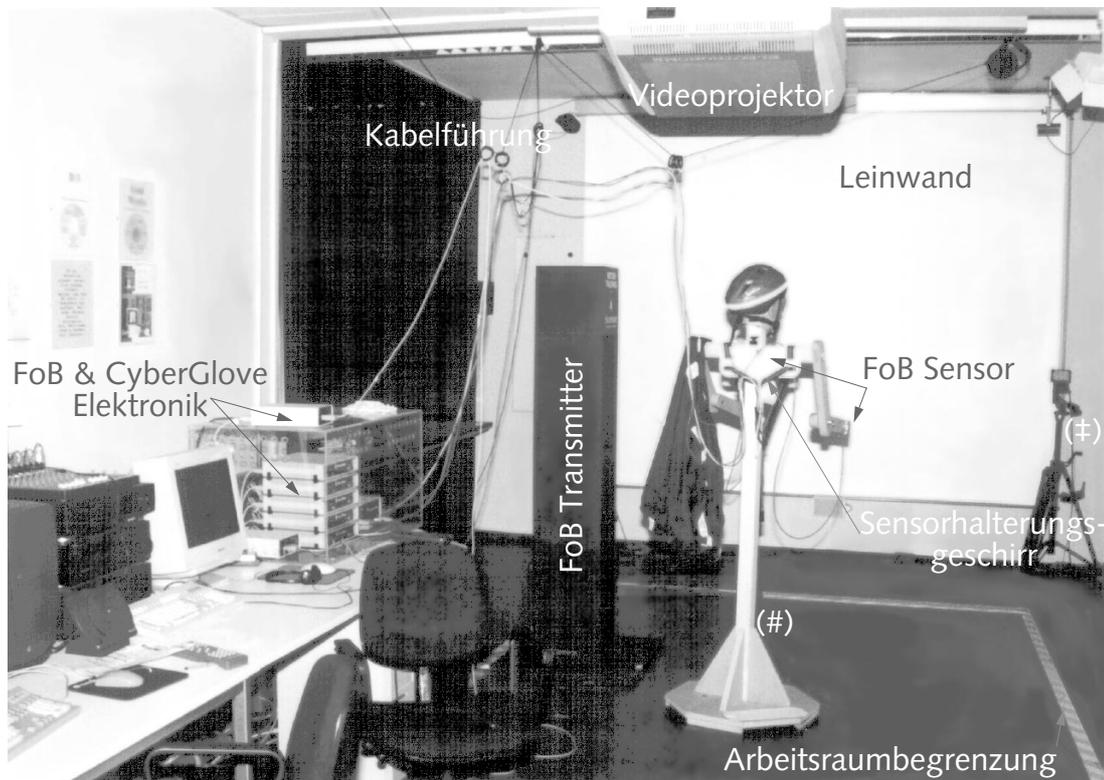


Abbildung 43: Flock of Birds und einzelner Sensor

Für die vorliegende Arbeit wurde sich aufgrund der hohen Abtastrate und Meßgenauigkeit sowie der kommerziellen Verfügbarkeit des Meßgerätes für das Produkt „Flock of Birds“ der Fa. Ascension Technologies Corporation, Burlington, Vermont, USA entschieden (siehe Abbildung 43). Dieses Gerät stellt eine maximale Abtastrate von 144 Hz unabhängig von der Anzahl der Sensoren zur Verfügung, hat einen Meßbereich von 10 Fuß also 3,048 m um den Transmitter und kann aus einem bis zu 14 Sensoren und aus einem bis vier Transmittern bestehen, wobei jede Kombination aus beiden Bereichen erlaubt ist. Dies reicht aus um den in Abbildung 44 auf Seite 107 zu sehenden Arbeitsraum vor der Großbildleinwand abzudecken. Die technischen Details werden im Anhang „Technik der Datenaufnahme“ auf Seite 171 genauer dargestellt.

Die Meßdaten werden von jedem Sensor oder wahlweise von einem als Zentrale (Master) auserkorenen Sensor wahlweise mittels EIA/TIA 232C, EIA/TIA 422 oder EIA/TIA 485 zu einem Rechner mit bis zu 115,2 kilobaud (310 kilobaud bei EIA/TIA 485) übertragen und einem Rechner empfangen der sie über einem im Zuge dieser Arbeit entwickelten Server in ein genormtes Datenformat konvertiert und dann an jeden weiteren Rechner versendet, der die Daten empfangen möchte. Als Repräsentationsformalismus kommt hier die „eXternal Data Representation language“, XDR zum Einsatz [Blo92]. XDR ist zusammen und für das „Remote Procedure Call protocol“, RPC, der Grundlage des „Network File Systems“, NFS von Fa. Sun Micro-



±) Eine der VisBoS-Kameras (vgl. [NR98b]), die andere #) „Stummer Proband“: Sensorhalter für Testzwecke in anthropometrischen Durchschnittsmaßen.

Abbildung 44: Labor für Künstliche Intelligenz und Virtuelle Realität in Bielefeld

systems entwickelt worden und mittlerweile Bestandteil vieler Standards im Bereich Netzwerkprotokolle.

Für das eigentliche Versenden der Daten wird das für das „Internet Protocol“, IP vorgeschlagene „Multicast Protocol“, auch bekannt als IP-Multicast benutzt [CS93][Ste97][Ste94]. Dabei handelt es sich um eine relativ neue Technologie, die es ermöglicht, ähnlich einem Radiosender, Nachrichten an eine quasi beliebige Anzahl von Zuhörern, im weiteren Klienten genannt, abzustrahlen. Der Klient benötigt dazu allerdings eine Stationskennung, die im IP-Multicast über reservierte IP-Adressbereiche zur Verfügung gestellt werden. Diese sog. Multicast-Adressen werden über die „Domain Name Services“, DNS verwaltet und als symbolische Namen analog zu Rechnernamen und Rechner-IP-Adresse verwendet. Unter einer jeden solchen Multicast-Adresse ist somit ein Datenstrom bestimmten Inhaltes zu empfangen. Der Mul-

ticastserver bietet die Meßdaten in verschiedenen Formaten an, dabei findet standardmäßig das Format „processed position matrix“ Verwendung. Dabei handelt es sich um eine Positionstriplet und eine drei Kreuz drei Orientierungsmatrix. Die Elemente dieser Strukturen sind im XDR-Datentyp „double“ repräsentiert, der dem gleichnamigen C-Datentyp weitgehend entspricht und Zentimeter als Einheit hat. Das „processed“ im Namen dieses Formates bezieht sich eben auf die implizite Umwandlung der Flock-Ticks in Zentimeter. Dabei wird aber nur mit einem festen Skalierungsfaktor multipliziert, nicht aber weitere Filter angewendet. Dies würde dem Konzept der Datenaufnahme widersprechen, die Daten des Flock of Birds unabhängig von der weiteren Verwendung allen Interessierten zur Verfügung zu stellen.

Da es sich bei den Konvertierungsschritten um simple Operationen linearer Komplexität handelt, fällt die hierfür benötigte Verarbeitungszeit in den Bereich weniger hundert Nanosekunden und kann somit vernachlässigt werden. Problematischer ist der Transport über das Internet. Hier sind nicht die Signal- oder Gatterlaufzeiten das Problem, sondern die Tatsache, daß im Internet keine exklusiven Verbindungen wie z.B. in der klassischen, analogen Telefonie, aufgebaut werden, sondern Datenpakete verschiedener Benutzer verzahnt nacheinander auf derselben Leitung verarbeitet werden. Multicastdatenpakete werden über das auf das IP aufbauende „User Datagram Protocol“, UDP abgewickelt. Ein solches Datentelegramm hat die Spezifikation, daß es ohne Rückmeldung versendet wird und daß es unter bestimmten Bedingungen an den Knotenpunkten des Netzes (router) kommentarlos weggeworfen werden kann. Der Sender hat also keine Information darüber, ob sein Datentelegramm empfangen wurde. Dies ist eine Voraussetzung für Multicasting, da die Anzahl der Klienten von 0 bis — theoretisch — „alle Rechner im Internet“ variieren kann. Doch schließt dies auch aus, daß ein Datentelegramm wiederholt werden kann, falls der Klient es nicht richtig empfangen hat. Auch der Klient kann so erst einmal nicht wissen, ob er alle Datentelegramme des Senders erhalten hat. Dieses Problem ist jedoch trivial durch die Vergabe geeigneter Seriennummern zu lösen. Damit weiß der Klient zwar, daß er etwas verpaßt hat, aber nicht, was es war. Dies ist aber auch kein echtes Problem, solange ein Nutzdatensatz in ein Datentelegramm paßt. Dann müssen die Datentelegramme nur mit einem Zeitstempel versehen sein; so stehen die Datentelegramme dann ein jedes für sich alleine.

Bei sehr starkem Netzverkehr können die Datentelegramme kurzzeitig verzögert werden, dies kann einige *100 ms* ausmachen. Auch dieser Effekt kann in der Praxis vernachlässigt werden, da zum einen mit der Netzwerkinfrastruktur vorgesorgt werden kann und da zum anderen, beim Einsatz von „Switching Hubs“¹⁾, der Klientenrechner

Teilnehmer an der lasterzeugenden Netzkommunikation sein muß, um merkliche Effekte zu erzielen. Daß letzteres auftritt, ist bei Multiuser/Multitasking-Rechnern zwar nicht ausgeschlossen, aber in der Regel plan- und somit vermeidbar.

In dieser Arbeit werden alle Daten von den Positionssensoren als die Daten aufgefaßt, die ein Klient in oben beschriebenen Aufbau *empfangen* hat.

7.3 Vorverarbeitung

Die von der Datenaufnahme versandten Daten können nun von beliebig vielen weiteren Prozessen empfangen werden. Dabei sind aus Gründen der Effizienz jedoch einige Überlegungen notwendig. Zuersteinmal können die im letzten Abschnitt vorgeschlagenen Methoden zur Konsistenzprüfung der Netzübertragung angewendet werden. Es kann jedes Datenpaket, das eine Zeitstempel trägt, der älter ist als der eines vorhergehenden Datenpaketes, verworfen werden, da es veraltet ist.

Weiterhin können die Daten Lauflängen-codiert werden, also nur die jeweiligen Änderungen unter Verwendung einer Hysterese (Änderungsschwellwert) weitergeleitet werden. Bei sinnvoll gewähltem Schwellwert, können dabei auch gleich Effekte, die durch das Rauschen der Sensoren (siehe Anhang „Technik der Datenaufnahme“ auf Seite 171) entstehen, ausgeblendet werden.

Wenn, wie in diesem Vorschlag, absolute Zeitstempel benutzt werden, müssen die Rechner, die die einzelnen Agenten beherbergen synchronisiert sein. Das geschieht im vorliegenden Fall über das Network Time Protocol (NTP) [Mil92]. Dabei ist eine Genauigkeit von ca. 500 µs für NTP in der hier verwendeten Version 4 gegeben (zum Vergleich: der System Call zum auslesen der Zeit benötigt ca. 1 µs) [Mil98].

Wenn das Rauschen der Muskulatur (Zittern) kein Parameter der weiteren Betrachtung sein soll, kann an dieser Stelle auch ein Low-Pass-Filter auf die Daten angewendet werden.

7.4 Körpermodell

In dem Abschnitt „Schicht 2b: Körpermodell“ auf Seite 82 wurden das geometrische und das dynamische Körpermodell eingeführt. Das gesamte Körpermodell soll a priori

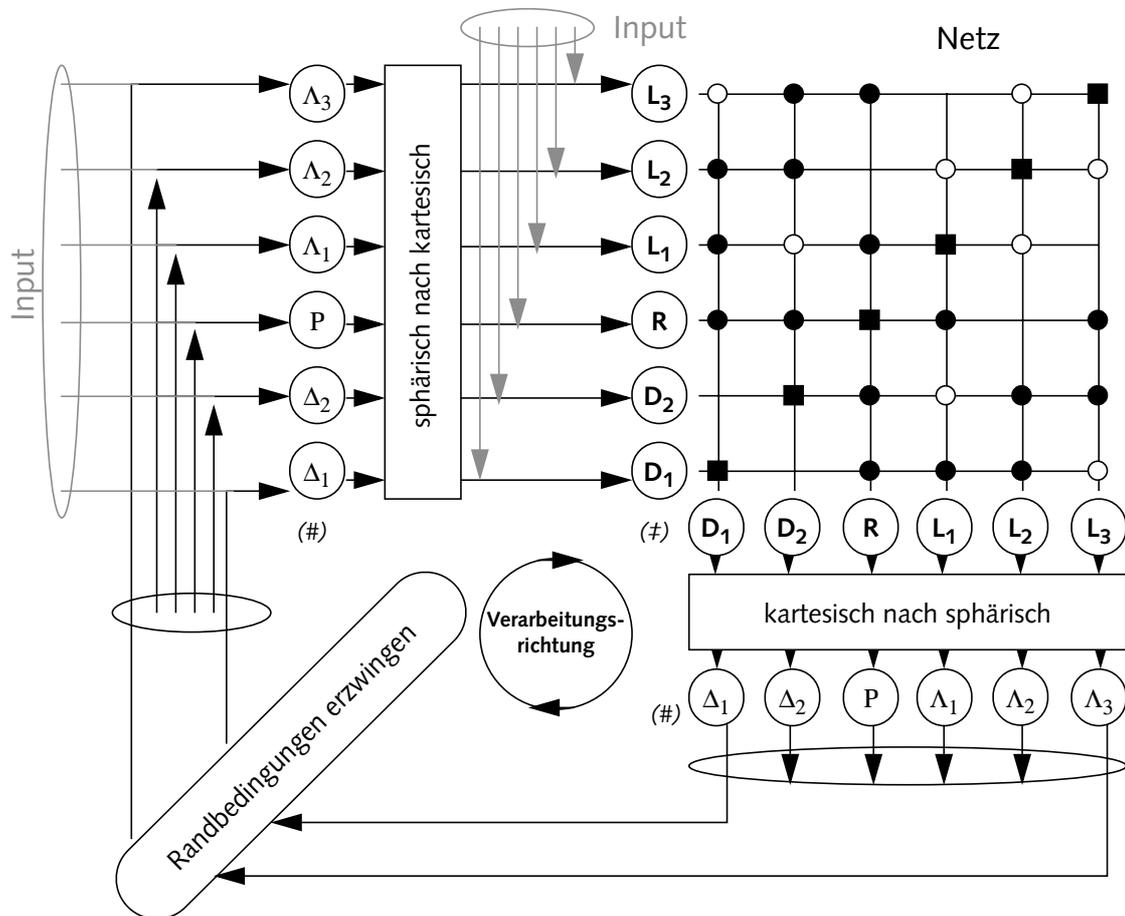
-
1. Bei dieser Technologie wird der Ethernet-Bus auf eine Backplane konzentriert und die einzelnen Geräte in Sternverkabelung betrieben. Dabei bekommt jeder Port nur noch die für ihn signifikante Information.
-

Wissen über den Oberkörper einbringen und eine Abstraktion von den realen Sensoren zu sog. virtuellen Sensoren vornehmen. Dabei wurde der geometrische Teil schon ausführlich dargestellt und Anforderungen für den dynamischen Teil formuliert. Dieser sollte robust und „anytime“-fähig das geometrische Modell in Konfigurations- und Ortsraum gegebenen Daten anpassen können. Biologischen Systemen scheint die Lösung kinematischer Aufgabenstellungen oft weniger Probleme zu bereiten. Zudem reagieren sie robust gegenüber auftretenden Sondersituationen. Erstaunlich ist auch, wie schnell Lösungen gefunden werden. Diese enorme Geschwindigkeit läßt sich nur auf die parallelisierte Signalverarbeitung zurückführen, die biologische Systeme mittels ihrer neuronalen Netze realisieren können. Ein biologisch motivierter Ansatz verfolgt daher auch das Ziel, einen parallelisierbaren Algorithmus zu entwickeln. In die engere Wahl kommen somit Verfahren, die ausgehend von einem gegebenen Zustand den Folgezustand auf Grund geänderter Datenlage berechnen.

Deshalb fiel die Wahl auf das „Mean of Multiple Computations Network“ (MMC-Netz) Verfahren von Ulrich Steinkühler und Holk Cruse [Ste94][CS93], das so die Basis für das *dynamische Körpermodell* bietet.

7.4.1 Mean of Multiple Computations Network (MMC-Netz)

Cruse und Steinkühler gehen bei ihrem Ansatz davon aus, daß allein die innere Geometrie eines redundanten kinematischen Systems ausreicht, um kinematische Problemstellungen zu lösen. Das von ihnen entwickelte Modell besteht aus einem rekurrenten Netzwerk, welches sich über die Problemdomäne entspannt (relaxiert) und nach einer gewissen Anzahl von Iterationen einen stabilen Zustand erreicht, der einer geometrisch korrekten Stellung des Manipulators entspricht. Dies funktioniert auch in Sonderfällen, beispielsweise wenn der Eingabevektor des Netzes das Problem nicht vollständig spezifiziert. Die dem Verfahren zugrunde liegende Idee ergibt sich aus der Überlegung, daß aufgrund der Redundanz des kinematischen Systems die zu ermittelnden Größen auf verschiedene Art und Weise berechnet werden können, indem man ihre wechselseitigen geometrischen Beziehungen ausnutzt. Aus den einzelnen Ergebnissen wird für jede Größe das arithmetische Mittel gebildet und bei der nächsten Iteration als Startwert verwendet. Diese Verfahrensweise motiviert den Namen des Modells: mean of multiple computation (Mittel aus mehrfacher Berechnung). Spezielle Zusatzbedingungen, wie beispielsweise Beschränkungen der Gelenkwinkel können nach der Mittelwertbildung auf die berechneten Größen angewendet werden, um das Ergebnis des nächsten Iterationszyklus zu beeinflussen. Auf diese Weise kann das Netz zur Lösung kinematischer Probleme eingesetzt werden, wobei es die Existenz etwaiger Begrenzungen sowohl im Gelenk- als auch im Arbeits-



†) Lateinische Buchstaben für die kartesische Notation #) Griechische Buchstaben für die sphärische Notation

Abbildung 45: Das „Mean of Multiple Computations“-Netz

raum des Manipulators bewältigen kann. Über die verschiedenen Schritte entspannt sich das Netz zu einer die Randbedingungen möglichst gut erfüllenden Konfiguration. In Abbildung 45 auf Seite 111 ist eine Graphik des Netzes zu sehen, in deren Netzkomponente ein leerer Kreis für ein negatives Gewicht, ein voller Kreis für ein positives Gewicht und ein Quadrat für ein Rückkopplungsgewicht steht. Die einzelnen Größen selbst sind wiederum Vektoren, typischerweise aus \mathbb{R}^3 .

Genauer gesagt, wird bei der Konstruktion eines solchen Netzes von einer Darstellung des zu modellierenden Armes, wie in Abbildung 46 dargestellt, ausgegangen, in der die Segmente als Vektoren dargestellt sind. Dieser Vektorzug bildet von seinem Ur-

sprung (o.B.d.A. der Nullpunkt) bis zu seinem Greifpunkt G einen gerichteten, azyklischen Graphen des Grades eins. Dieser Graph kann immer zu einem gerichteten, zyklischen Graphen erweitert werden, in den vom Nullpunkt bis zum Endpunkt der Segmentkette (Greifpunkt) eine Kante R in Richtung des Endpunktes eingefügt wird. Weiterhin kann dieser Graph zu einem vollständigen Graphen erweitert werden, dabei sollen alle Kanten vorzugsweise vom Nullpunkt in Richtung Greifpunkt orientiert sein. Die so entstandenen Kanten können wieder als Vektoren interpretiert werden und werden wie folgt bezeichnet: Die ursprünglichen n Segmente als L_1 bis L_n , der eingefügte Greifvektor als R und die Vektoren, die den alten Graphen nun zu einem vollständigen Graphen ergänzen, als D_x . Die so erhaltenen Vektoren können ebenfalls in einer Winkel-Länge-Darstellung repräsentiert werden, hier mit griechischem (Λ, P, Δ) , entsprechend den lateinischen Buchstaben (L, R, D) , bezeichnet.

Im nächsten Schritt wird versucht, eine Reihe von einfachen Gleichungen aus dem Graphen abzuleiten. Dies geschieht, indem Kreise in dem Graphen gefunden werden und je nach Orientierung der Kanten die entsprechenden Vektoren addiert (mit der Kantenorientierung) oder subtrahiert (gegen die Kantenorientierung) werden. Da ein Kreis auf seinen Anfangspunkt zurückführt, entspricht ein so gefundener Term der Null. Aus obigem Graphen lassen sich also sieben Gleichungen der Form $L_3 + L_2 + L_1 - R = 0$, $L_2 + R - D_2 - D_1 = 0$ usw. (siehe Abbildung 47) ableiten.

Im nächsten Schritt bei der Erstellung des Netzes werden alle Variablen der Reihe nach betrachtet und jedesmal eine Liste von Gleichungen erstellt, in denen die jeweilige Va-

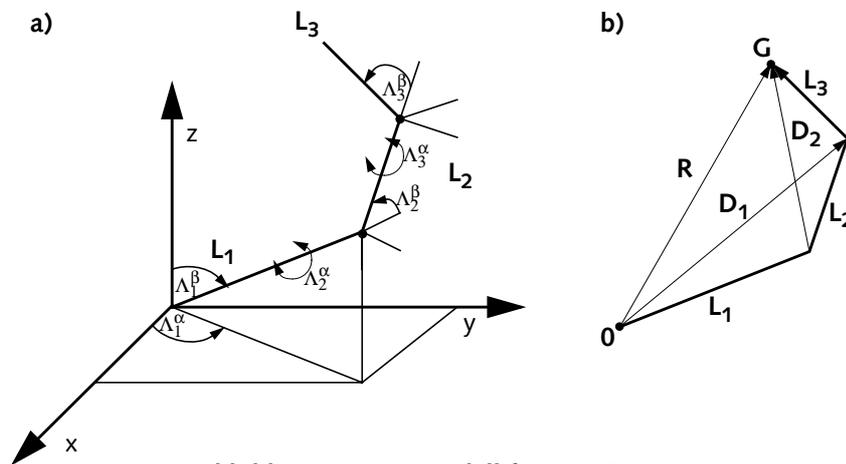


Abbildung 46: Armmodell für MMC-Netz

L_1	$+L_2$	$+L_3$			$-R$	$= 0$
	$+L_2$		$-D_1$	$-D_2$	$+R$	$= 0$
L_1		$-L_3$	$-D_1$	$+D_2$		$= 0$
L_1	$+L_2$		$-D_1$			$= 0$
	$+L_2$	$+L_3$		$-D_2$		$= 0$
		$+L_3$	$+D_1$		$-R$	$= 0$
L_1				$+D_2$	$-R$	$= 0$

Abbildung 47: Grundgleichungen des MMC-Netzes

riable vorkommt. Dann wird jede dieser Gleichungen zu dieser Variablen hin umgeformt und das arithmetische Mittel berechnet. Z.B. für L_3 : $\bar{L}_3 = \frac{1}{4}(-2L_2 + 2R + 2D_2 - 2D_1)$.

Mit dieser Methode lassen sich für alle Variablen Mittelwertsterme bilden, die die Grundlage für den Aufbau des Netzes bilden. Führt man dabei die Division mit 4 explizit durch, so nehmen die einzelnen Koeffizienten ausschließlich die diskreten Werte $-0,5$; 0 und $+0,5$ an.

Zu guter letzt bleibt nur noch der Selbst- oder Rückkopplungsfaktor d zu erklären. Dieser wird als eine Low-Pass-Dämpfung eingebaut, um Überschwingungen zu vermeiden. Für das L_3 -Beispiel ergibt sich so $\bar{L}_3 = \frac{1}{(4+d)}(-2L_2 + 2R + 2D_2 - 2D_1 + dL_3)$.

Wenn, wie vorliegend, $d = 5$ gewählt wird, und alle Gewichte in das rechteckige Schema eines Netzes eingetragen werden, ergeben sich so vier Gewichtsfaktoren. Einmal Null, um anzuzeigen, daß diese Variable nicht in einen Term eingeht, dann $2/9$ und $-2/9$ als positive und negative Kopplung sowie 5 als Selbstkopplungsfaktor.

Wenn das Netz mit einem Satz Werten gestartet wird, entspannt es sich derart, daß eine möglichst günstige Konfiguration eingenommen wird. Wenn nun in jedem Schritt eine Randbedingung in Form eines Intervalls für jede Variable erzwungen wird, kann so eine günstige Armkonfiguration gefunden werden.

ABSCHÄTZUNG DER KOMPLEXITÄT IN ABHÄNGIGKEIT VON DER SEGMENTZAHL. Hier ist ein kleiner mathematischer Einschub sinnvoll, der es im weiteren erlauben wird, eine Kosten-Nutzen-Abschätzung (Komplexitätsabschätzung) für einige Parameter des

MMC-Netzes durchzuführen: Das oben dargestellte System mit drei Segmenten kann relativ einfach auf einen aus beliebig vielen Segmenten bestehenden redundanten Manipulator erweitert werden. Für jedes hinzukommende Gelenk werden neue Vektoren eingeführt, welche die aufeinander folgenden Gelenke miteinander verbinden. Auf diese Weise entsteht eine Vektorkette von der Basis des Systems (o.B.d.A. der Nullpunkt des Koordinatensystems) bis zum Arbeitspunkt des Endeffektors R. Zusätzlich führt man Diagonalen ein, die jeden Joint mit allen anderen Joints verbinden. Die einzelnen Vektoren bilden geschlossene Wege, für die sich, wie oben gezeigt, lineare Gleichungen aufstellen lassen. Der neue Wert eines Vektors ergibt sich aus dem Mittelwert der Gleichungen, in denen er enthalten ist. Nach dem MMC-Prinzip fließen damit bei der Berechnung auch die aktuellen Werte der anderen Segmente mit ein.

Etwas formaler kann die Darstellung des Verfahrens auch unter einer graphentheoretischen Betrachtung behandelt werden. Bezeichnet man mit j die Anzahl der Gelenke eines idealisierten Manipulators, so kann das dazugehörige System von Vektoren als vollständiger ungerichteter Graph G mit $n = j + 1$ Knoten betrachtet werden. Die Anzahl der signifikanten Gleichungen $K(G)$ errechnet sich aus der Summe aller knotendisjunkten Hamilton-Kreise¹⁾ in G und seinen einzelnen Teil-Graphen bis zur Länge j .

Es existieren $\binom{n}{k}$ Möglichkeiten, aus einem Graphen G mit Knotenzahl n einen Teilgraphen G' mit k Knoten auszuwählen. Die Anzahl der knotendisjunkten Hamilton-Kreise im Graphen G' beträgt $\frac{(k-1)!}{2}$. Damit ergibt sich für die Anzahl der relevanten (s.o.) Kreise in G der wie folgt zu berechnende Wert $K(G)$:

$$K(G) = \frac{1}{2} \sum_{k=3}^n \binom{n}{k} (k-1)! \quad (\text{Gl. 6})$$

Das heißt, daß für jede Variable eine gewisse Anzahl Gleichungen gefunden werden kann. Diese Anzahl Gleichungen entspricht der Anzahl der Kreise im Graphen $K_i(G)$,

$$K_i(G) = \frac{n \times (n+1)}{2} \quad (\text{Gl. 7})$$

an denen ein (beliebiger aber fester) Knoten i beteiligt ist.

1. Ein Hamilton-Kreis bezeichnet einen geschlossenen Weg in einem Graphen (Kreis), bei dem jeder Knoten genau einmal passiert wird.

In Abbildung 48 sind beide Funktionen zusammen mit 10^x in einem logarithmischen Diagramm dargestellt. Wie dort leicht abzulesen ist, steigt die Anzahl der Kreise (Gleichungen) zwar mit wachsendem n sehr stark (exponentiell) an, die Anzahl der Gleichungen pro Variable hingegen bleibt, verglichen damit, gering (polynomiell).

Diese Komplexitätsaussagen betreffen allerdings auf den ersten Blick nur die Erstellung des Netzes, und haben keinen Einfluß auf die Laufzeit bzw. Größe des Netzes, die ja nur von der Anzahl der Variablen des Modells mit n^2 abhängt. Bei näherer Betrachtung wird jedoch klar, daß einerseits eine exponentiell wachsende Komplexität bei der Erstellungsphase große Probleme bereiten kann, da auch die Erstellung des Netzes algorithmisch — hier mit $O(x^n)$ — erfolgt und andererseits mit wachsender Zahl von Gleichungen für eine Variable die Gewichtungsfaktoren sehr klein werden können, was aufgrund der iterativen Natur des Netzes zu numerischen Problemen führen kann.

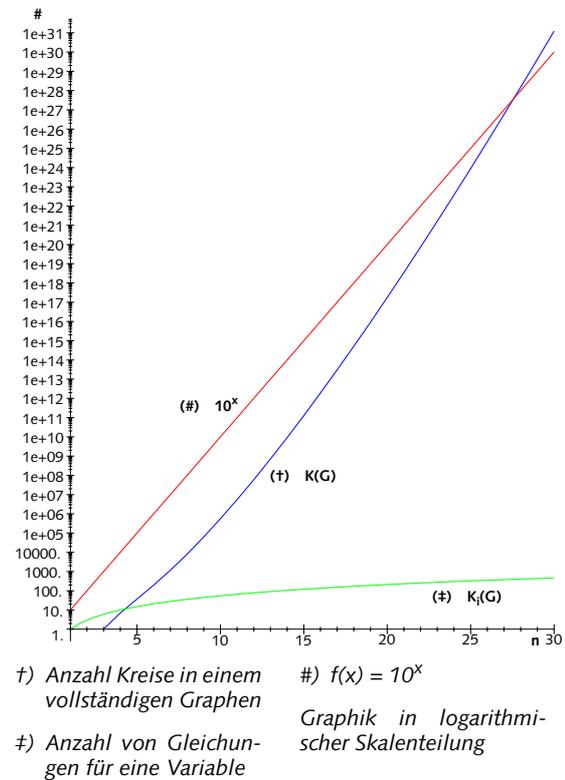


Abbildung 48: Benötigte Gleichungen für eine Variable

Bislang wurde der rein lineare Teil des MMC-Netzes beschrieben. Wie Steinkühler und Cruse beweisen konnten, konvergiert diese Netz [Ste94], allerdings mit der ungünstigen Eigenschaft, daß die Segmentlängen nicht beibehalten werden.

Daher haben sie eine nichtlineare Erweiterung des Netzes vorgeschlagen, in der die einzelnen Segmente nicht nur in kartesischer, sondern auch in einer sphärischen (zwei Winkel eine Länge) Darstellung konsistent beschrieben werden. So kann auch auf die Segmentlängen eine Randbedingung gelegt werden, die eine konstante Segmentlänge erzwingt. Für diese nichtlineare Erweiterung konnte die Konvergenz noch nicht bewiesen werden, doch zeigen ausführliche Tests von Steinkühler und Cruse einerseits und Tests, die in einer Diplomarbeit der Arbeitsgruppe „Wissensbasierte Systeme“

der Technischen Fakultät der Universität Bielefeld durchgeführt wurden, [Alt98], daß typischerweise nach nur 30 Iterationen eine in Anwendungen brauchbare Konfiguration eingenommen wird. Der Abstand Ist- und Zielvektor lag dabei in der Größenordnung des Meß- (Gestenerkennung) bzw. Darstellungsfehlers (Computergraphik).

7.4.2 Anwendung des MMC-Netzes auf das Körpermodell

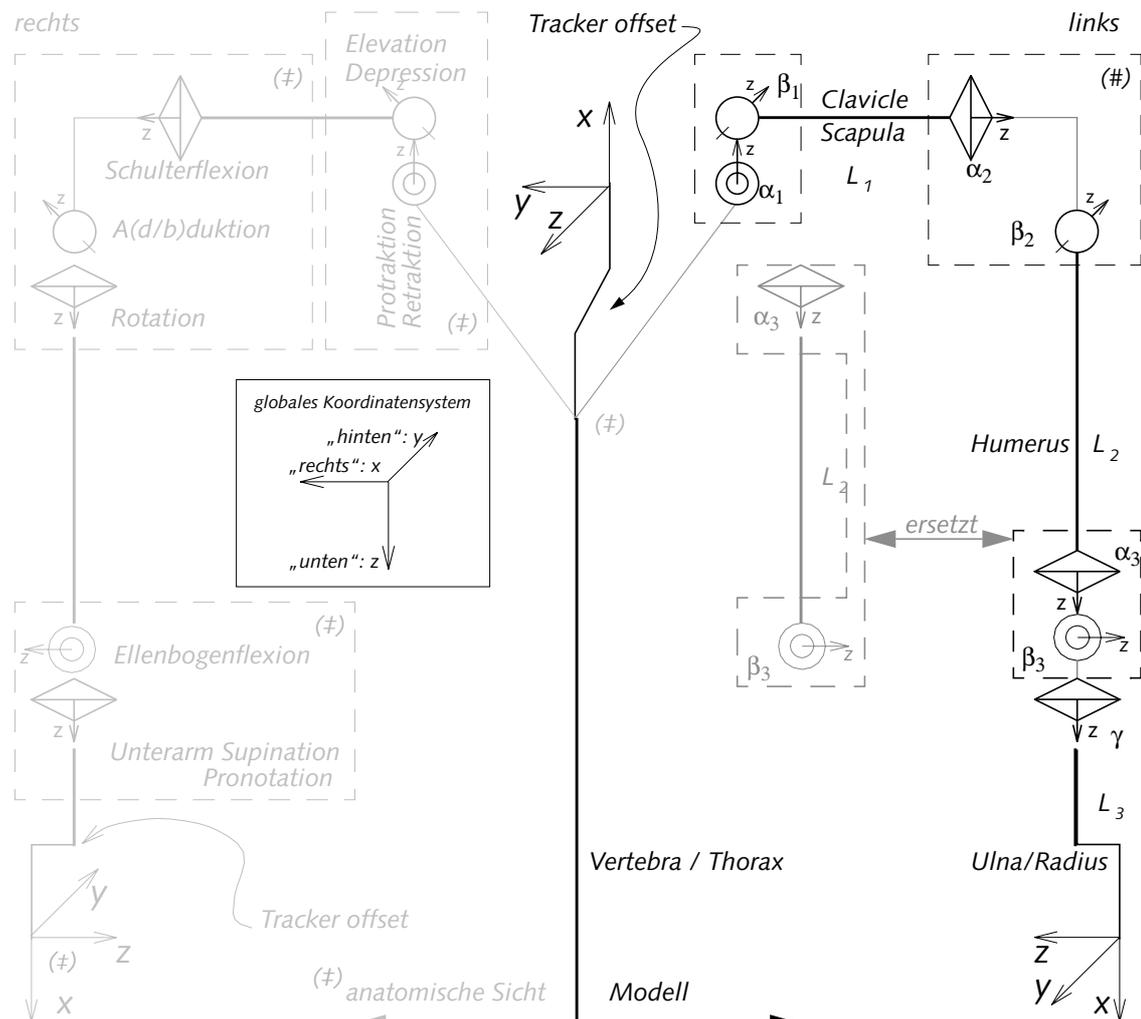
Wie in Abbildung 49 auf Seite 117 dargestellt, können nun die Parameter des MMC auf das geometrische Körpermodell abgebildet werden. Vereinfachend wandert die Schulterrotation dabei an den Ellenbogen. Die Darstellungen sind äquivalent, da der Oberarm als rotationssymmetrisch angenommen wird¹⁾. Der Winkel γ in Abbildung 49 wird dabei nicht von dem MMC modelliert, kann aber dem Sensor direkt entnommen werden, da dieser nicht nur die drei Freiheitsgrade der Position sondern noch zusätzlich die drei Freiheitsgrade der Rotation mißt. Der Sensor zwischen den Schultern, der auf oder nahe dem letzten Halswirbel C7 oder vertebra prominens anzubringen ist, stellt die Positionsreferenz auf den Probanden im Arbeitsraum dar und ist der Null- bzw. Ausgangspunkt in dem Modell.

7.4.3 Längenbezeichner im Körpermodell

In den weiteren Abschnitten werden einige Längenbezeichner benutzt die sich direkt oder mittelbar aus den virtuellen Sensoren des Körpermodells ablesen lassen. Dabei werden die Segmentlängen des Körpermodells nach einer Heuristik konsistent aus der Größe (engl. *Stature*) des Probanden berechnet. Diese Heuristik verwendet Durchschnittswerte für die relative Länge der Segmente für jeweils den weiblichen und den männlichen Teil verschiedener Populationen, um daraus dann wieder die absolute Segmentlänge zu berechnen. Die dafür nötigen Meßverfahren, Segment- und Winkeldefinitionen sind gemäß [Phe86], [Win79], [DIN33402T1], [DIN33402T2b] und [NASA94] gewählt und die Daten aus [Phe86], [Con72], [DGF59], [DIN33402T2] und [NASA94] entnommen. Alle Vorfaktoren beziehen sich auf den absoluten Größenwert, hier als *Stature* (H) bezeichnet.

Ein unabhängiger Kontrollagent überwacht das Körpermodell und für den Fall das Inkonsistenzen in einer Belegung beobachtet werden, wird eine bessere ausgewählt. Wenn z.B. der Punkt G (aus Abbildung 46b) weiter als die modellierte Armlänge vom

1. Diese Annahme macht praktisch nur dann Probleme, wenn ein Sensor am Oberarm befestigt wird und der Sensor auch Rotationen messen soll. Doch da der Oberarm mit vielen großen Muskeln ausgestattet ist, ist er — wie schon im Abschnitt „Schicht 1: Datenaufnahme“ auf Seite 78 dargelegt — kein geeigneter Ort einen Sensor zu befestigen.



‡) Siehe Abbildung 37, „Geometrisches Körpermodell nach [FW98]“ auf Seite 88

#) Die gestichelten Kästchen fassen Winkel einer Winkel-Länge-Darstellung zusammen.

Nur bezeichnete Segmente haben eine von Null verschiedene Länge. Die dünnen Linien bezeichnen Pseudo-seg-

mente der Länge Null. Die Bezeichner sind nach Abbildung 45, „Das „Mean of Multiple Computations“-Netz“ auf Seite 111. gewählt. Das dunkelgraue Segment wurde ersetzt. Wird, wie in diesem Modell, davon ausgegangen, daß L_2 rotationssymmetrisch ist, sind die Darstellungen äquivalent.

Abbildung 49: Anwendung des MMC auf das Körpermodell

Körper entfernt ist, ist anzunehmen, das es sich um einen weiblichen Probanden handelt, da Frauen im Allgemeineren eine größere relative Armlänge haben. Auf diese

weise lassen sich weitere kleine Heuristiken finden um zwischen den Belegungen der Variablen des Modells umzuschalten.

Name	Bezeichnete Funktion
Stature	Absolute Größe des Probanden
EyeHeight	Abstand Boden-Augen
ChinHeight	Abstand Boden-Kinn
ShoulderHeight	Abstand Boden-Schulter
ChestHeight	Abstand Boden-Brust
StomachHeight	Abstand Boden-Bauch
ElbowHeight	Abstand Boden-Ellenbogen
HipHeight	Abstand Boden-Hüfte
WristHeight	Abstand Boden-Handgelenk
FingertipHeight	Abstand Boden-Fingerspitzen
Chin2TopOfHead	Abstand Kinn-Kopfoberseite
MaxHeadDiameter	Abstand Kinn zum anderen Ende des Kopfes
ShoulderBreadth	(biacromial) Abstand zwischen den Schultergelenken
ClavicleLength	(acromial) Abstand von der Wirbelsäule zur Schulter.
ShoulderElbow	Länge des Humerus (Oberarmes)
ElbowFingertipLength	Abstand Ellenbogen – Fingerspitzen
HandLength	Abstand Handgelenk – Fingerspitzen
SittingHeight	Abstand Gesäß – Vertex (Kopfoberseite)
ChestDepth	Brusttiefe (Min. Abstand zweier untereinander und zur coronalen Ebene parallelen, den Brustkorb je dorsal und ventral begrenzenden Ebenen)

Tabelle 14: Bezeichner im Körpermodell

Bei den Bezeichnern in Tabelle 14 auf Seite 118 wird von einem stehenden Probanden in „Halbacht“-Stellung (amerikanische Version, mit ausgestreckten Fingern) ausgegangen.

Zusammen ermöglichen das geometrische und das dynamische Körpermodell eine Repräsentation der oberen Gliedmaßen. Dabei sorgt das geometrische Modell für die Benutzung plausibler Segmentlängen und das dynamische Modell für die Berechnung sinnvoller Winkel und Übergänge.

7.5 Merkmalerkenner

In diesem Abschnitt wird auf jene kleine Agenten eingegangen, die zum ersten Mal ordinale Information zu nominalen Symbolen codieren. Dazu wird dargestellt in welche Symbolklassen sich diese Symbole unterteilen lassen und wie die codierende Abbildung in Termen des Körpermodells hier vorgeschlagen wird.

Die Merkmalerkenner fassen Bereiche von Signalzuständen zu einer Hypothese zusammen. Die Bereiche erstrecken sich typischerweise über nur einer Dimension oder Größe. Dabei operieren die Merkmalerkenner über eine Menge von Symbolklassen, die zusammen mit ihren Symbolen, im weiteren vorgestellt werden. Die Summe dieser in den Symbolklassen verwendeten Symbole formt den Symbolvorrat eines Notationssystems, das im folgenden HNS+ genannt wird.

GESTENREPRÄSENTATION. Um Gesten adäquat repräsentieren zu können, muß eine Menge von zu notierenden Merkmalen ausgewählt werden. Dabei wird von den Atomen von HamNoSys (Siehe „Hamburger Notationssystem“ auf Seite 47) ausgehend eine Untermenge namens HNS+ gebildet. Die Mächtigkeit dieser Untermenge ist ausgehend von den Semantikfunktionen der Symbole gewählt. Sie umfaßt zunächst einmal die verhältnismäßig einfach zu erkennenden Merkmale oder einfache Merkmalskomplexe. Das Gesamtsystem ist jedoch darauf ausgelegt, diese Untermenge auf ganz HamNoSys erweitern zu können. Eine Übersicht über HNS+ findet sich wegen ihrer Länge im Anhand als Tabelle 23, „Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente“ auf Seite 159. Dabei werden neben den HamNoSys-Glyphen auch die in dem System normalerweise benutzen ASCII-Bezeichner angegeben.

Da die Semantik der HamNoSys-Zeichen keine geometrisch exakten Grenzen für die Gültigkeitsbereiche der einzelnen Bezeichner festlegt, es sich also um vage Konzepte handelt, müssen mit Hilfe von anthropometrischer Statistik Annahmen über diese Be-

reiche formuliert werden und mit einem Verarbeitungsverfahren, das unsicheres Wissen zu handhaben vermag, gearbeitet werden.

Im folgenden werden die die Symbole der einzelnen Klassen und ihre Abbildung auf den Sensordatenraum mittels des Körpermodells einzeln aufgeführt. In den Tabellen wird neben den HNS+-Namen auch die HamNoSys-Glyphen aufgeführt.

7.5.1 Ortsangabe im Gestikulationsraum

Der Raumausschnitt, in dem Gesten physikalisch stattfinden können wird Gestikulationsraum genannt. Relativ zum Körper kann so in den drei Raumdimension „oben-unten“ (Höheneinteilung), „nahe – entfernt“ (Abstand zum Körper) und „links – rechts“ (Abstand zu den Körpernormalen) der Ort der Hand bezeichnet werden.

HÖHENEINTEILUNG. Die Höheneinteilung unterscheidet sieben Höhenbereiche wie in Tabelle 15 dargestellt. Dabei wird der Konfidenzwert als relativer Abstand (prozen-

Nr.	Name		Bereich ^a
1	Above head	○	Stature – URH^b
2	Head	○	ChinHeight – Stature
3	Shoulder	☐	ChestHeight – ChinHeight
4	Chest	☐	ElbowHeight – ChestHeight
5	Stomach	☐	HipHeight – ElbowHeight
6	Below Stomach	☐	FingertipHeight – HipHeight
7	Rückweisungsbereich		URH – $+\infty \wedge -\infty$ – FingertipHeight

Tabelle 15: Höheneinteilung im Gestikulationsraum

a. Die Bezeichner decken sich mit denen in Tabelle 14 auf Seite 118.

b. **UpperReachHeight (URH)** lokaler Bezeichner für (**ShoulderHeight - WristHeight**)

tuale Abweichung) von der Mittelebene des Intervalls, hier als prototypischer Vertreter des Bereiches aufgefaßt, dargestellt. Somit ist der Konfidenzwert 1-normiert.

ABSTAND ZUM KÖRPER. Für den Abstand zum Körper sind vier Kategorien vorgesehen. Dabei wird in Kauf genommen, daß durch Verrechnung einiger anthropometrischer Merkmale sich der Fehler erhöht (siehe Tabelle 16).

Nr.	Name		Bereich ^a
1	superficial signing space	χ	0 – ChestDepth
2	proximal signing space	⋈	ChestDepth – (ChestDepth + HandLength)
3	neutral signing space	⊘	(ChestDepth + HandLength) – (ChestDepth + ShoulderElbowLength)
4	distal signing space	↗	(ChestDepth + ShoulderElbowLength) – (ShoulderHeight - FingertipHeight)
5	Rückweisungsbereich		(ShoulderHeight - FingertipHeight) – ∞

Tabelle 16: Abstandseinteilung im Gestikulationsraum

a. Die Bezeichner decken sich mit denen in Tabelle 14 auf Seite 118.

ABSTAND ZU DEN KÖRPERNORMALEN: TRANSVERSALE SEGMENTE. Auch die transversale Ebene wird gemäß Abbildung 50 auf Seite 122 in verschiedene Zonen eingeteilt. Dabei werden, entsprechend des HamNoSys-Vorbildes, die in Tabelle 17 dargestellten

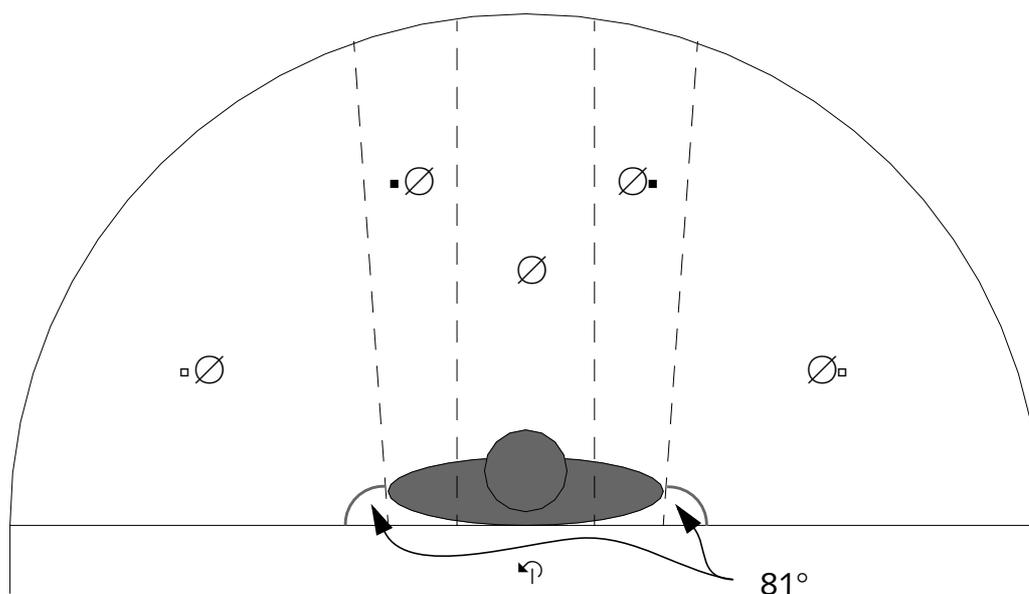
Nr.	Name		Bereich
6	far left	◻∅	ventral alles links der um 81° gekippten Ebene links der linken Schulter
1	left	◼∅	ventral zwischen der Ebene links des Thorax und der um 81° gekippten Ebene links der linken Schulter

Tabelle 17: Positionen im Gestikulationsraum

Nr.	Name		Bereich
2	middle	∅	ventral zwischen den Schultern
3	right	∅▪	ventral zwischen der Ebene rechts des Thorax und der um 81° gekippten Ebene rechts der rechten Schulter
4	far right	∅▪	ventral alles rechts der um 81° gekippten Ebene rechts der rechten Schulter
5	behind	↶	dorsal alles hinter (posterior) der Ebene, die parallel zur coronalen Ebene am hintersten Punkt des Körpers liegt

Tabelle 17: Positionen im Gestikulationsraum (Forts.)

Zonen unterschieden. Die Zonen sind zwar de facto durch den Arbeitsraum des Probanden begrenzt, wie in der Graphik durch den Halbkreis angedeutet wird, doch wird



Diese Abbildung ist ein Auszug der Abbildung 21, „HamNoSys: Einteilung der transversalen Ebene“ auf Seite 53 und bezieht sich nur auf die Bezeichner aus Tabelle 17 auf Seite 121.

Abbildung 50: Einteilung der transversalen Ebene

diese Grenze hier nicht noch einmal getestet, da dies schon in dem in Abschnitt „Abstand zum Körper“ auf Seite 121 dargestellten Erkener behandelt wird.

Da diese drei Raumdimensionen auf den Achsen des Koordinatensystems des Körpermodells liegen, können hier die Komponenten des Ortsvektors unabhängig voneinander einzeln auf die jeweiligen Intervalle trivialerweise abgebildet werden.

7.5.2 Bewegungen im Gestikulationsraum

Für sprachbegleitete Gestik sind einfache Bewegungen wichtig, erst wenn ein starker Zeichencharakter bei einer Geste auftritt, müssen unter Umständen auch komplexere Trajektorien betrachtet werden. Daher beschränkt sich die Erkennung auf den geradlinigen Anteil von Bewegungen. Einige komplexere Trajektorien können daraus im folgenden Integrationsschritt inferiert werden.

GERADLINIGE BEWEGUNGEN. Diese Bewegungen können immer als Bewegungsanteile auf den drei Hauptebenen — coronaler, transversaler und sagittaler Ebene — betrachtet werden. Dabei wird jede Ebene in acht gleichgroße Segmente, ausgehend von ihrem Schnittpunkt, zerlegt. Nach Abzug der doppelt bezeichneten Segmente ergibt sich hier eine Menge von acht Segmenten auf der coronalen Ebene, sechs Segmenten auf der transversalen Ebene und vier Segmenten auf der sagittalen Ebene — also zusammen 18 Segmente. Dabei werden die in Abbildung 51 dargestellten Einteilungen und die Einteilungen wie in Tabelle 18 auf Seite 124 benutzt. Die Erkennung wird hier durch den Vergleich mit 18 Einheitsrepräsentanten vorgenommen.

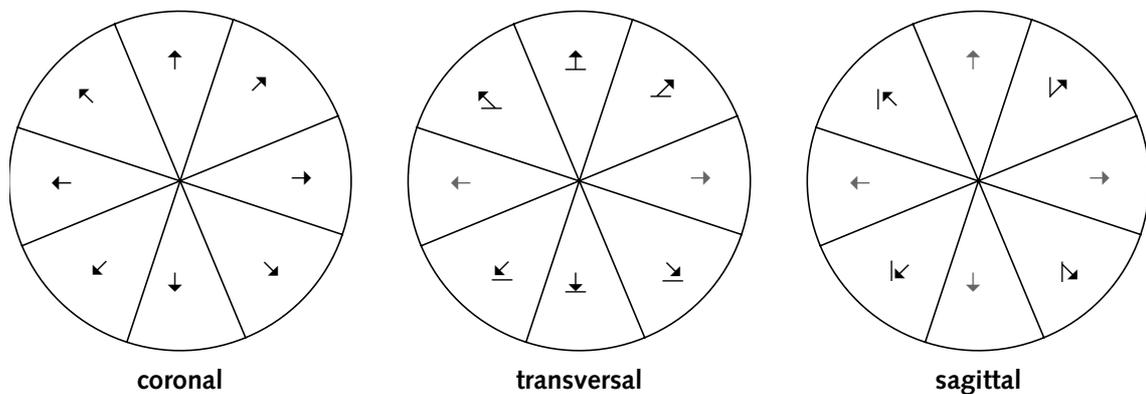


Abbildung 51: Bewegung: Einteilung der Hauptebenen

Nr. ^a	Name		Bereich
1	MoveL	←	coronal: $0 - \pi/4$
2	MoveLD	↙	coronal: $\pi/4 - \pi/2$
3	MoveD	↓	coronal: $\pi/2 - 3/4 \pi$
4	MoveRD	↘	coronal: $3/4 \pi - \pi$
5	MoveR	→	coronal: $\pi - 5/4 \pi$
6	MoveRU	↗	coronal: $5/4 \pi - 6/4 \pi$
7	MoveU	↑	coronal: $6/4 \pi - 7/4 \pi$
8	MoveLU	↖	coronal: $7/4 \pi - 2 \pi$
1	MoveA	⊕	transversal: $0 - \pi/4$
2	MoveAL	↖	transversal: $\pi/4 - \pi/2$
3,(4)	MoveTL	↙	transversal: $3/4 \pi - \pi$
4,(5)	MoveT	⊕	transversal: $\pi - 5/4 \pi$
5,(6)	MoveTR	↘	transversal: $5/4 \pi - 6/4 \pi$
6,(8)	MoveAR	↗	transversal: $7/4 \pi - 2 \pi$
1,(2)	MoveAU	↗	sagittal: $\pi/4 - \pi/2$
2,(4)	MoveTU	↖	sagittal: $3/4 \pi - \pi$
3,(6)	MoveTD	↙	sagittal: $5/4 \pi - 6/4 \pi$
4,(8)	MoveAD	↘	sagittal: $7/4 \pi - 2 \pi$

Tabelle 18: Bewegungsrichtungen im Gestikulationsraum

a. Die Nummer des Segmentes im Kreis (positive Richtung) ist in Klammern angegeben, wenn sie nicht mit der Nummer des Symbols übereinstimmt.

7.5.3 Bewegungsart im Gestikulationsraum

Die Bewegungscharakteristika wie Geschwindigkeit und Ruck sind relativ schwer in Symbole zu fassen, da es einerseits schwer ist, verlässliche Daten über die Wahrnehmung von Geschwindigkeit und Ruck zu erhalten und andererseits das Signalrauschen im richtigen Maß auszufiltern, um die Erkennung nicht auf technische Artefakte anspringen zu lassen (siehe dazu auch Anhang „Störungen bei der Zuweisung von Zeitstempeln“ auf Seite 175).

GESCHWINDIGKEIT ALS QUANTISIERUNG DES BEWEGUNGSVEKTORS. Die Geschwindigkeit, sofern eine Bewegung auftritt, wird durch die drei in Tabelle 19 vorgestellten Symbole

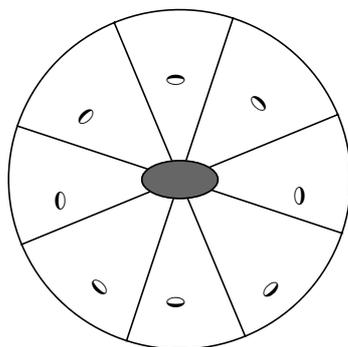
Nr.	Name		Bereich
1	MoveFast	*	20,0 cm/s – <i>TOOMUCHSPEED</i> ^a
2	MoveNorm		10,0 cm/s – 20,0 cm/s
3	MoveSlow	–	3,3 cm/s – 10,0 cm/s
4	keine Bewegung	∅	0 cm/s – 3,3 cm/s
5	Rückweisungsklasse		<i>TOOMUCHSPEED</i> – $+\infty$

Tabelle 19: Geschwindigkeit

- a. **TOOMUCHSPEED** ist ein Systemparameter, der zur Laufzeit eingestellt werden kann, da die sinnvolle Grenze zum Fehlerbereich hin stark von der jeweiligen Anwendung abhängt.

repräsentiert.

RUCK ALS WEITERE ABLEITUNG DER GESCHWINDIGKEIT. Die dritte Ableitung des Ortes wird in der Physik auch „Ruck“ genannt. Wenn ein bestimmter Schwellwert (z.B. 4,0 cm/s³) überschritten wird, wird das Symbol „**MoveHalt**“ („||“) emittiert. Die anderen Bewegungsmodi, nämlich „**MoveTense**“ („x“) und „**MoveRest**“ („?“), benötigen entweder Wissen, das in der Form der Bewegung nicht enthalten ist, oder einen weiteren Kontext, der erst in der Integrationsphase gegeben ist.



Querschnittsebene Unterarm

Abbildung 52: Handflächenorientierung:
Einteilung der Ebene

7.5.4 Handflächenorientierung

Ähnlich wie die Bewegungsrichtungen wird auch die Handflächenausrichtung in acht Segmenten auf einer zum Unterarm senkrechtstehenden Ebene codiert (siehe Abbildung 52 auf Seite 126). Wie an dieser Beschreibung deutlich wird, handelt es sich trotz des Namens eigentlich um die Pronation/Supination, die an der Hand mit gestrecktem (geraden) Handgelenk betrachtet wird.

7.5.5 Die Konfiguration der Hand

Die Konfiguration der Hand wird durch einen direkten Mustervergleich der Exoskelettsensoren mit vorgegebenen Winkel- und Winkelgeschwindigkeitsdaten vorgenommen. Dabei werden die Flexionssensoren zu einem Streckungsgrad des Fingers verrechnet. In Abbildung 53 ist die Hand in Pronationshaltung mit den Sensoren des Exoskelettes verzeichnet. Jedem zu erkennendem HNS+-Symbol ist ein einzelner Agent zugeordnet. Wird der zuvor beschriebene Wertebereich in Winkel und Winkelgeschwindigkeit betreten oder verlassen, meldet der Agent dieses Ereignis weiter.

7.6 Integrationsverfahren

Im folgenden wird nun dargelegt, wie die mit den dargestellten Mitteln erzeugten einfachen Symbole zu Symbolen komplexerer Semantik (sog. komplexeren Symbolen) integriert werden können. Dabei ist es sinnvoll, alle Integrationsinstanzen im System auf einem gemeinsamen Grundprinzip aufzubauen. Dieses Prinzip wird im folgenden *Integrationsrahmen* genannt. Eine Darstellung dieses Integrationsrahmens,

achtet werden muß. Jede generierbare Hypothese ist aus Basishypothesen, zwischen denen einschränkende Randbedingungen bestehen, zusammengesetzt. Die Basishypothesen bilden die Antezedenzen der (einer) Regel zur Produktion der neuen Hypothese, und daher muß mindestens eine Instanz jeder Basishypothese im Gedächtnis vorliegen, damit die Konsequenz der Regel ausgeführt werden kann. Im Falle einer zeitlich strukturierten Geste etwa, in der Nachzeitigkeitsbedingung vorkommen, kann die Gestenhypothese erst mit der Assertion der zeitlich zuletzt auftretenden Basishypothese generiert werden. Es ist keine Inferenz möglich, die bereits ein Gestensymbol produzierte, obwohl die Geste noch nicht vollständig durchgeführt wurde. Eine Lösung dieses Problems ist, dafür zu sorgen, daß jede Hypothese zu jedem Zeitpunkt mindestens einmal im Gedächtnis instantiiert ist, auch wenn ihre Konfidenz null ist. Dies wird durch Instantiierung einer sog. *persistenten Instanz* jeder vorkommenden Hypothesenklasse bei der Definition neuer Hypothesen erreicht. Persistente Instanzen haben die Konfidenz null und erstrecken sich über einen unendlich großen Zeitraum und können nicht gelöscht werden, verhalten sich aber ansonsten wie jede andere Regel.

Zusammen mit dem Prinzip des zeitlich begrenzten Speicherns erlaubt diese Art der schritthaltenden Verarbeitung eine Architektur, die ohne ein vollständiges, von Expertensystemen bekannten, Truth Maintenance System (TMS) auskommt. TMS speichern Ursachen, die zur Assertion eines Faktums geführt haben. Damit ist es möglich, ein Faktum beim Wegfallen seiner Ursache zu löschen. Ein komplette Inferenzkette läßt sich auf diese Weise rückgängig machen. Im hier vorgestellten Ansatz werden falsche bzw. sehr unsichere Hypothesen einerseits von anderen Hypothesen mit einem höheren Konfidenzwert überlagert, andererseits fallen sie nach einiger Zeit aus dem Speicher (dem Gedächtnis), wenn sie zuvor nicht zu einer komplexeren Hypothese integriert worden sind; damit erübrigt es sich, sie explizit zu löschen. Ein gewisses „belief update“ läßt sich trotzdem nicht vermeiden, wenn auch noch unvollständige Zeitintervalle zur Integration betrachtet werden sollen.

7.6.2 Wissen über Zeitintervalle

Zur Durchführung einer temporalen Integration ist es notwendig, den zeitlichen Gültigkeitsbereich von Hypothesen zu definieren. Dazu wird jeder Hypothese ein Start und optional ein Endzeitpunkt als zusätzliche Komponente mitgegeben, die zusammen das Gültigkeitsintervall beschreiben.

Abhängig von den Eigenschaften der Hypothesenklassen lassen sich verschiedene Mechanismen zur Behandlung des Gültigkeitsintervalls anwenden. Dazu ist es nütz-

lich, den Konfigurationsraum K aller möglichen Eingaben über alle Sensoren in der Zeit zu betrachten. Gibt es eine Hypothesenklasse H , die unabhängig von der Konfiguration $k \in K$ immer instantiiert wird, dann heißt H vollständig. Ein Beispiel: Sei H die Klasse, die mit der Flexion und Extension des Ellenbogens assoziiert ist. H habe zwei Unterklassen H_1 und H_2 , die Flexion bzw. Extension repräsentieren. Offenbar ist H dann vollständig, da der Ellenbogen sich immer entweder in Flexion oder in Extension befindet, andere Möglichkeiten gibt es nicht. Daher gibt es immer eine Hypothese $h \in H$. Sei, im Gegensatz dazu, H die Klasse der definierten Gesten und besitze H die beiden Unterklassen Zeigegeste und Drehgeste, dann ist H nicht vollständig, da der Benutzer auch andere Gesten oder gar keine Geste vollführen kann. Ferner heißt eine Hypothesenklasse H eindeutig, wenn es semantisch keine zwei Instanzen h_1, h_2 von H geben kann, deren Gültigkeitsintervall überlappt. Im oben gewählten Beispiel der Ellenbogenbeugung ist H eindeutig („entweder oder“), im Falle von Gestenklassen jedoch nicht eindeutig, wie man am obigen Beispiel der konkurrierenden Hypothesen sehen kann.

Für vollständige und eindeutige Klassen bietet sich auf der technischen Seite das Verfahren an, statt eines kompletten Gültigkeitsintervalls nur den Startzeitpunkt mitzuteilen und anzunehmen, daß die Hypothese bis zum Widerruf durch eine andere Instanz derselben Klasse gültig ist. Kommt eine andere Instanz hinzu, so kann der Endzeitpunkt der alten Hypothese auf den Startzeitpunkt der neuen gesetzt werden. Da die Hypothesen der unteren Ebene diese geforderten Eigenschaften oft aufwiesen, erspart dieses Vorgehen gerade auf den unteren Ebenen des Integrationsdreiecks Kommunikationsaufwand, da dort viele Hypothesen in kleinen Zeitabständen emittiert werden. Wird eine Instanz aus einer nicht vollständigen Hypothesenklasse auf diese Weise auf den Stand der Zeit gebracht, wird sie mit dem nun vollständig spezifizierten Intervall und dem weiteren Inhalt der alten Instanz assertiert und die unvollständige, alte Version wird gelöscht. In diesem Sinne erfährt eine Hypothese ein „belief update“.

7.6.3 Informationsverlust bei Integration zu komplexeren Symbolen

Bei der Klassifizierung von Signalen und der Integration von Symbolen zu komplexeren Symbolen kann Information verloren gehen. So sagt ein Tupel ${}^t(1; -0,5; 0)$ in einem gegebenen Koordinatensystem (hier ${}^t(\text{rechts, hinten, unten})$) mehr aus als ein Symbol „...“ oder „**LocRightSide**“. Um aber diese Information „retten“ zu können, ist es sinnvoll, sowohl kategorisches als auch räumliches Wissen zu repräsentieren. Eine kategorische Wissensdarstellung in Form von Hypothesen mit festgelegten Bedeutungen bietet den Vorteil, daß auf alle Hypothesen dieselben Mechanismen zur

Symbolmanipulation angewendet werden können, unabhängig von der Bedeutung. Auf der anderen Seite gibt es allerdings Informationen, die sich nur sehr schwer symbolisch repräsentieren lassen, dazu zählen unter anderem präzise Ortsangaben. Gerade diese Art der räumlichen Information kann mit deiktischen Gesten besonders gut übermittelt werden. Als einfaches Beispiel sei hier eine Zeigegeste angeführt. Nicht nur die Information, daß eine Zeigegeste vorliegt, ist von Interesse, sondern auch die exakte Richtung des Zeigestrahls, um damit mögliche Referenten zu identifizieren. Zusammenfassend kann gesagt werden, daß kategorisches Wissen Antwort auf die Frage „Was für ein Konzept liegt vor?“ gibt, während sich mit räumlichem Wissen die Frage „Wie kann dieses Konzept quantitativ charakterisiert werden?“ beantworten läßt. In dieser Sichtweise ist die räumliche Information eine Erweiterung der kategorischen Information. Bei der Repräsentation von Hypothesen schlägt es sich darin nieder, daß jede Instanz einer Hypothesenklasse mit quantitativer Information annotiert werden kann.

Da die räumliche Information von den Integratoren nicht manipuliert wird, sondern der Auflösung durch andere, spezialisierte Agenten bedarf, die sie in kategorische Information umwandeln können, reicht es aus, sie als Komponente einfach mitzuführen und auf sie dann, wenn die Auflösung der räumlichen Information notwendig ist, zuzugreifen. Diese Komponente wird im folgenden auch „pay-load“ oder *Nutzlast* genannt. Diese Nutzlast beschreibt also die Fundierung eines Symbols im Signal und behandelt insofern das Problem des „symbol grounding“ [Har90] (siehe auch Abschnitt „Harnad’sche symbolische Repräsentation“ auf Seite 27).

Wenn nun die zeitlich sortierte Folge der Symbole genommen und die Nutzlast betrachtet wird, ergibt sich eine indirekte Einteilung des Signalflusses in aussagekräftige Passagen. Dabei hängt die Aussagekraft von der Definition des nutzlasttragenden Symbols ab.

7.6.4 HNS+ – Syntax

Wie schon erwähnt, können die Symbole von HNS+ Bestandteile der Symbolhierarchie in der Integrationshierarchie sein. Aber auch zur *Definition* höherer Symbole werden HNS+-Ausdrücke benutzt. HNS+ selbst ist eine Sprache, deren Grammatik sich in Backus-Naur-Form (BNF) niederschreiben läßt. Ein gültiger HNS+-Ausdruck wird mit dieser BNF beschrieben:

⟨HNS+-Ausdruck⟩ →	⟨Konfiguration⟩ ⟨Symmetrie⟩ ⟨Konfiguration⟩ ⟨Konfiguration⟩ ⟨Aktion⟩ ⟨Symmetrie⟩ ⟨Konfiguration⟩ ⟨Aktion⟩
⟨Symmetrie⟩ →	· ·̄ ·̇ ·̈ ·̉ ·̊ ·̋
⟨Konfiguration⟩ →	⟨Handform⟩ ⟨Handstellung⟩ ⟨Lokation⟩ ⟨Handstellung⟩ ⟨Lokation⟩ ⟨Handform⟩ ⟨Lokation⟩ ⟨Handform⟩ ⟨Handstellung⟩ ⟨Handform⟩ ⟨Handstellung⟩ ⟨Lokation⟩
⟨Handform⟩ →	⟨Grundform⟩ ⟨Grundform⟩ ⟨Daumenstellung⟩ ⟨Grundform⟩ ⟨Fingerbeugung⟩ ⟨Grundform⟩ ⟨Daumenstellung⟩ ⟨Fingerbeugung⟩ ⟨Verbindungsform⟩ ⟨Verbindungsform⟩ ⟨Öffnungsweite⟩ ⟨Verbindungsform⟩ ⟨Fingerbeugung⟩ ⟨Verbindungsform⟩ ⟨Öffnungsweite⟩ ⟨Fingerbeugung⟩
⟨Grundform⟩ →	○ ◯ ◰ ◱ ◲ ◳
⟨Verbindungsform⟩ →	◁ ▷ ◂ ▸ ▹ ►
⟨Daumenstellung⟩ →	↙ ↘
⟨Fingerbeugung⟩ →	┌ ^ ▽
⟨Öffnungsweite⟩ →	└ ∨
⟨Handstellung⟩ →	⟨Fingeransatzrichtung⟩ ⟨Handflächenorientierung⟩ ⟨Fingeransatzrichtung⟩ ⟨Handflächenorientierung⟩
⟨Fingeransatzrichtung⟩ →	< r l > - + ^ v △ ▽ ⊥ ∞ ⊥ ⊥ ⊥ ⊥ ⊥
⟨Handflächenorientierung⟩ →	○ ◯ ◰ ◱ ◲ ◳ ◴
⟨Lokation⟩ →	⟨Körperebene⟩ ⟨Abstand⟩ ⟨Körperebene⟩ ⟨Abstand⟩
⟨Körperebene⟩ →	◻ ◼ ◽ ◾ ◿
⟨Höhe⟩ →	○ ◯ ◰ ◱ ◲ ◳
⟨Abstand⟩ →	↔) (↔
⟨Aktion⟩ →	[⟨Aktionen⟩] [⟨Aktionen⟩] ⟨Bewegung⟩
⟨Aktionen⟩ →	⟨Aktion⟩ ⟨Aktion⟩ ⟨Aktionen⟩
⟨Bewegung⟩ →	⟨absolute Bewegung⟩ ⟨relative Bewegung⟩ ⟨einfache Bewegung⟩ ⟨Konfigurationsübergang⟩
⟨absolute Bewegung⟩ →	⟨Lokation⟩ ⟨einfache Bewegung⟩ ⟨Lokation⟩
⟨relative Bewegung⟩ →	⟨Lokation⟩ ⟨einfache Bewegung⟩
⟨Konfigurationsübergang⟩ →	⟨Handform⟩ ↘ ⟨Handform⟩ ⟨Handstellung⟩ ↘ ⟨Handstellung⟩ ⟨Handform⟩ ⟨Handstellung⟩ ↘ ⟨Handform⟩ ⟨Handstellung⟩

denzwert und einem Zeitstempel versehen sind. Wie in Abschnitt „Zeitbegriffe“ auf Seite 31 für den menschlichen Informationsverarbeitungsprozeß dargelegt wurde, wird Information wachsenden Komplexitätsgrades auch in Verarbeitungsschritten mit wachsender Verarbeitungsdauer verarbeitet. Ein jeder Integrator steht für einen Komplexitätsgrad, benutzt aber den gleichen Meta-Mechanismus zur Integration, den Integrationsrahmen. Um dieses Modell adäquat umzusetzen, muß also jeder Integrator auf einem Zeitfenster arbeiten und die inferierte Information nach oben weiterleiten.

Dieses Zeitfenster wird durch eine *Abklingfunktion* realisiert. Wenn ein neues Faktum instantiiert wird, werden die Konfidenzwerte entsprechend ihres Alters (Zeitstempels) abgewertet. Unterschreitet der Konfidenzwert eine gewisse Signifikanzschwelle, so wird das Faktum (bzw. Symbol, bzw. Instanz) gelöscht. Eine Abklingfunktion hat folgende drei Eigenschaften:

1. Sie bildet auf einen Gewichtungsfaktor zwischen null und eins ab.
2. Sie ist monoton fallend, mit $f(0) = 1$.
3. Sie unterschreitet nach einer endlichen Zeit t_{max} die Signifikanzschwelle.

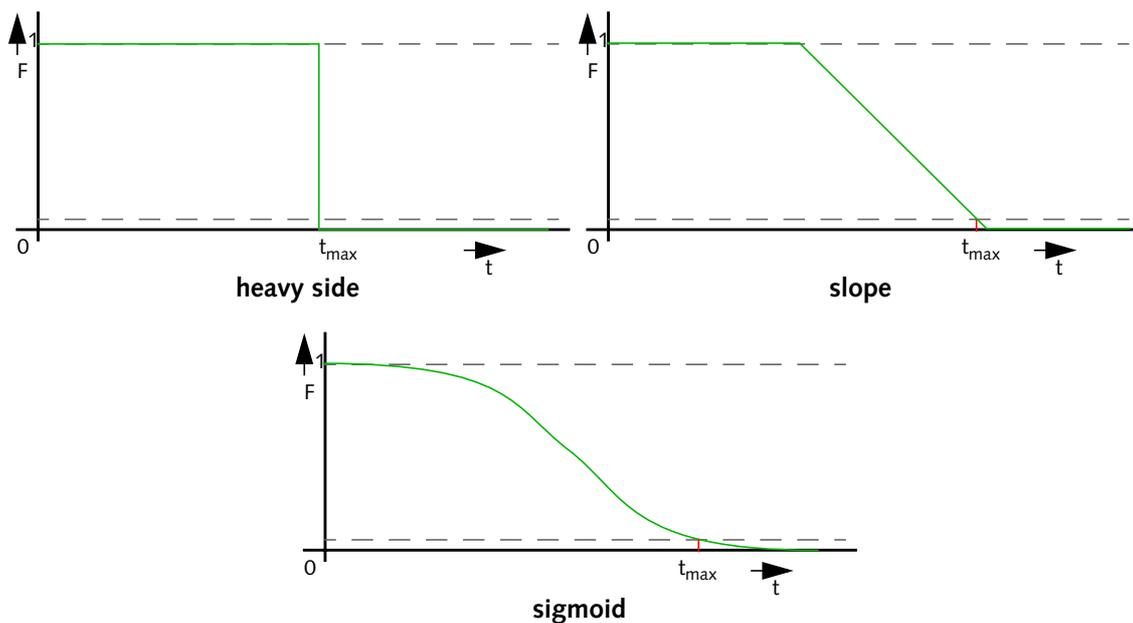


Abbildung 54: Abklingfunktionen

In Abbildung 54 auf Seite 133 sind einige Abklingfunktionen skizziert. Für den Pöpel'schen Ansatz ist die Rechteckfunktion für t_{max} („heavy side“) als Abklingfunktion zu wählen.

7.6.7 Segmentierung

Die Segmentierung des Gestenstroms wird hier also über eine Theorie der menschlichen Wahrnehmung Modelliert. Dies erscheint adäquat, da der intendierte Rezipient menschlicher Kommunikation im Normalfall ebenfalls ein Mensch ist.

Um eventuelle Schwächen in diesem Modell auffangen zu können, ist es notwendig auch andere Hinweise auf Segmentgrenzen auszunutzen, soweit sie einem bekannt sind [SFL99], und einen Mechanismus zu implementieren, der Segmentierungshinweise, die mittels eigener Segmentierungsnachrichten versendet werden auszuwerten. Tritt ein solcher Segmentierungshinweis auf, wie z.B. der von Harling und Edwards in [HE96] auf Handspannung beruhende Segmentierungshinweis, muß dies im Integrator berücksichtigt werden. Als weitere Segmentierungshinweise sollen für die weitere Forschungsarbeit rhythmische und prosodische Informationen des Sprachkanals herangezogen werden.

HANDSPANNUNG ALS SEGMENTIERUNGSHINWEIS. Das in [HE96] vorgestellte Modell beruht auf einer simplifizierten Vorstellung der Finger einer Hand, wie in Abbildung 55 dargestellt. Dabei bezeichnet $\Theta_{n,t} \in [0 \dots \pi/2]$ Krümmung des Fingers n zur Zeit t und d_n die (konstante) Länge des Fingers n .

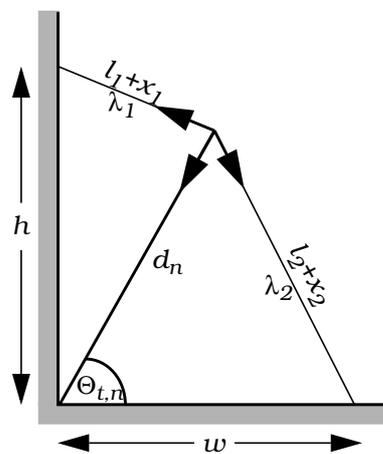


Abbildung 55: Fingermodell nach [HE96]

Die Fingerspannung $T_{n,t}$ berechnet sich nach (Gleichung 8), wobei x_1 (Gleichung 9)

$$T_{n,t} = \frac{\lambda_1 x_1}{l_1} \sqrt{1 - \frac{h^2 (\cos \Theta_{n,t})^2}{(l_1 + x_1)^2}} + \frac{\lambda_2 x_2 (d_n - w (\cos \Theta_{n,t}))}{l_2 + x_2} \quad (Gl. 8)$$

$$x_1 = \sqrt{h^2 (\cos \Theta_{n,t})^2 + (d_n - h \sin \Theta_{n,t})^2} - l_1 \quad (Gl. 9)$$

und x_2 (Gleichung 10) als die Ausdehnung elastischer Fäden gemäß dem Hook'schen

$$x_2 = \sqrt{w^2 (\sin \Theta_{n,t})^2 + (d_n - w \cos \Theta_{n,t})^2} - l_2 \quad (Gl. 10)$$

Gesetz ist. Die Handspannung H_t insgesamt ist dann die Summe über alle Fingerspannungen (Gleichung 11).

$$H_t = \sum_{n=1}^5 T_{n,t} \quad (Gl. 11)$$

Durch Simplifizierung des Modells durch einfache Wahl von $l_1, l_2, \lambda_1, \lambda_2, w, h$ zu 1, werden die Gleichungen, ohne eine wirkliche Beschränkung einzugehen, einfacher berechenbar. Die Fingerlänge wurde daraufhin gemäß den Angaben über den Cyber-Glove [VT94] zu den Werten in Tabelle 20 berechnet.

	Daumen	Zeigef.	Mittelf.	Ringf.	kleiner F.
Länge	8,8 cm	9,4 cm	10,4 cm	9,6 cm	6,9 cm

Tabelle 20: Fingerlängen

Grad	Korrelation	$\bar{\Delta}$	$\sigma(\bar{\Delta})$
2	0,999916	8,02e-09	0,00137
	0,999915	-2,86e-09	0,00138
	0,999914	6,11e-09	0,00139
	0,999915	-8,34e-09	0,00139
	0,999923	8,59e-09	0,00131
3	0,999917	9,17e-09	0,00137
	0,999916	-1,40e-09	0,00138
	0,999914	8,280e-10	0,00139
	0,999915	-6,68e-09	0,00139
	0,999923	4,20e-09	0,00131
4	~1,00 ^a	8,72e-09	4,12e-06
	~1,00	3,18e-10	5,071e-06
	~1,00	-6,36e-11	5,64e-06
	~1,00	-1,46e-08	4,91e-06
	~1,00	3,50e-09	8,33e-07
5	~1,00	1,08e-08	4,36e-06
	~1,00	-3,50e-09	6,10e-06
	~1,00	2,54e-10	5,65e-06
	~1,00	-9,23e-09	7,94e-06
	~1,00	-7,64e-10	5,85e-06

Tabelle 21: Fehler der approximierten Polynome (pro Finger)

- a. Durch die numerischer Rechnung ergibt sich eine endliche Darstellungsgenauigkeit. Da der Fehler nicht wirklich gleich 0 ist, ist die Korrelation nicht wirklich gleich 1.

Trotz dieser Vereinfachung enthält der Term immer noch trigonometrische Funktionen, die in der Berechnung teuer sind. Um diese aber nicht berechnen zu müssen wurden die Terme numerisch auf dem Intervall $]0.. \pi/2]$ durch effizient berechenbare Polynome approximiert. Dabei wurden verschiedene generische Polynome der Grade 2 bis 5 untersucht. In Tabelle 21 sind der Grad dieser Polynome und die Korrelation der Originaldaten mit den approximierten Daten sowie der mittlere Fehler der dabei entstandenen Abweichung und dessen Standardabweichung aufgeführt. Unschwer

läßt sich ablesen, das die Polynome nützliche, für viel Fälle schon in ihrer einfachsten Form hinreichende Näherungen bieten.

Grad	Korrelation	$\bar{\Delta}$	$\sigma(\bar{\Delta})$
2	0,999945	-2,09e-10	0,001224
3	0,999946	-1,61e-10	0,001224
4	$\sim 1,00^a$	4,54e-10	4,81e-06
5	$\sim 1,00$	2,96e-10	4,81e-06

Tabelle 22: Fehler der approximierten Polynome (Fingerlänge 1)

- a. Durch die numerischer Rechnung ergibt sich eine endliche Darstellungsgenauigkeit. Da der Fehler nicht wirklich gleich 0 ist, ist die Korrelation nicht wirklich gleich 1.

Das Fingerspannungsmodell ist dermaßen abstrakt, daß auch die Länge der einzelnen Finger vernachlässigt werden kann. Mit $\forall n \in 1..5: d(n) = 1$ ergibt sich ein universelles Fingerspannungsmodell. Wird die Approximation wiederholt, so ergeben sich ähnliche Funktionen mit ähnlichen Fehlerwerten, wie in Tabelle 22 auf Seite 137 abzulesen ist.

Für $T_{n,t}$ ergeben sich im ersten Fall fünf Terme (2. Grades), die im folgenden als C-Ausdrücke angegeben sind:

$$(0.1461128673E2+(-0.1059488697E1+0.674423331*x)*x)*x \quad (G1. 12)$$

$$(0.158105351E2+(-0.1060103947E1+0.6748144595*x)*x)*x \quad (G1. 13)$$

$$(0.1780955096E2+(-0.1060825265E1+0.6752732074*x)*x)*x \quad (G1. 14)$$

$$(0.1621031364E2+(-0.106027201E1+0.6749213905*x)*x)*x \quad (G1. 15)$$

$$(0.1081503111E2+(-0.1055699525E1+0.6720137954*x)*x)*x \quad (G1. 16)$$

Und im weiter vereinfachten Fall ergibt sich für $T_{n,t}$ als Näherung mit einem Polynom 2. Grades für die universelle Fingerspannung folgender C-Ausdruck:

$$(0.2998823648+(-0.1216636577E1+0.7743213677*x)*x)*x \quad (G1. 17)$$

Diese nach Horner-Schema angegebenen Terme lassen sich effizient berechnen. In Anhang „Näherung der Handspannungsfunktion“ auf Seite 177 wird die Näherung genauer dargestellt und es sind Graphen der Funktion $T_{n,t}$ und ihrer Näherungen sowie die weiteren Näherungspolynome (3. bis 5. Grad) angegeben.

WIRKUNGSWEISE DER SEGMENTIERUNGSHINWEISE. Wenn ein Segmentierungshinweis empfangen wird, ist in dem Zeitmodell des Integrationsrahmes klar, das alle Information, die vor dem Segmentierungshinweis liegt, veraltet ist. Ebenso soll alle Informationen die jünger als der Segmentierungshinweis sind ohne die ältere Informationen betrachtet werden.

Dies läßt sich durch einfaches Löschen aller Informationen, die älter als der Segmentierungshinweis sind, erreichen.

7.7 Implementierungsaspekte

Im folgenden wird dargelegt mit welchem Werkzeug die soeben dargestellte Integration implementiert werden kann. Dabei wird auf ein spezielles Expertensystemwerkzeug eingegangen und seine Fakten- und Regelbasis in einem dynamischen Speicher erörtert.

Die Implementierung des Integrationsrahmens umfaßt die Kernfunktion, bestehend aus dem Expertensystem-Tool CLIPS zusammen mit dem Basiswissen, das alle Integratoren teilen, einem C++-API, das die Schnittstelle zum Integrator bildet, sowie die äußere Hülle mit einem C++-API, dem Integrator-Agenten, der alle agentenbezogenen Funktionen enthält, und der Anbindung an das Gesamtsystem dient.

Die wesentlichen Kriterien zur Auswahl eines Produktionssystems für den Integrationsrahmen sind:

- einfache Einbindung in C/C++-Programme
- hohe Verarbeitungsgeschwindigkeit
- direkte Unterstützung zum Aufbau von Klassenhierarchien (*is-a*-Hierarchien)

CLIPS. Das Expertensystem-Tool CLIPS (C-Language Integrated Production System) erfüllt die gestellten Anforderungen und findet daher als Herzstück des Integratoragenten Verwendung. CLIPS wurde vom Software Technology Branch des NASA/Lyndon B. Johnson Space Center entwickelt und existiert seit 1986 [Gia94].

Seit der ersten Version hat CLIPS eine starke Weiterentwicklung erfahren, die zu der hier benutzten Version 6.05 geführt haben [GR93].

CLIPS ist kein reines Produktionssystem, da es neben der Formulierung des Wissens als Regeln auch die Repräsentation von prozeduralem Wissen in Form von definierbaren Funktionen, sowie objektorientiertes Programmieren unterstützt. CLIPS enthält eine eigene Sprache, in der Regeln, Funktionen, Methoden, usw. notiert werden können. Diese Sprache wird interpretiert, so daß zur Laufzeit Konstrukte hinzugefügt, gelöscht oder verändert werden können. Die objektorientierte Sprache innerhalb von CLIPS heißt COOL (CLIPS object oriented language). Hier ist ein einfaches Beispiel für eine Regel, notiert in CLIPS:

```
(defrule duck
  (animal-is duck)
=>
  (assert (sound-is quack)))
```

Das Konstrukt definiert die Regel **duck**, die als Vorbedingung (**animal-is duck**) besitzt, und als Konsequenz (**sound-is quack**) in die Faktenbasis assertiert.

CLIPS ist vollständig in C implementiert und liegt im Quelltext vor. Die Manipulation der Fakten- und Regelbasis sowie aller anderen Konstrukte kann statt in der CLIPS-Sprache auch mittels des C-APIs erfolgen, so daß der Interpretationsaufwand entfällt. Ebenso lassen sich neue Funktionen auch direkt in C programmieren und als CLIPS-Funktion registrieren. Das erlaubt eine beliebige Erweiterung der CLIPS-Funktionalität mit eigenen Funktionen. Die CLIPS-Funktionen sind als Laufzeitbibliothek verfügbar, wodurch eine einfache Integration in eigene C/C++-Programme möglich ist.

In jedem Produktionssystem muß für die pattern-matching-Phase ein Vergleich der Regel-Vorbedingungen mit den aktuell im Speicher befindlichen Fakten stattfinden, damit die aktivierbaren Regeln auf die Agenda gesetzt werden können. CLIPS bedient sich dazu des RETE-Verfahrens [For82], das unter der Voraussetzung, daß sich von Inferenzschritt zu Inferenzschritt nur wenige Fakten ändern, besonders effizient ist. RETE basiert auf einem Sortiernetzwerk, das aus allen Vorbedingungen aller Regeln gebildet wird. Im wesentlichen besteht es aus Prüfknoten, durch die getestet wird, ob sich bei der Assertion eines neuen Faktums notwendige Bedingungen erfüllen. Das Netzwerk speichert erfüllte Teile einer Vorbedingung ab, so daß ihre Gültigkeit bei einem nachfolgenden Inferenzschritt nicht mehr untersucht werden muß.

BASISWISSEN. Das Basiswissen liegt in Form mehrerer Dateien in der COOL vor, und ist für alle Integrator-Agenten auf allen Ebenen des Integrationsdreiecks identisch. Es besteht aus der Definition verschiedener Basis-Klassen, zu denen auch die Klasse **Hypothesis** zählt. Die symbolische Information, die zwischen den Integratoren kommuniziert wird, liegt immer als Instanz einer Klasse vor, die von der Oberklasse **Hypothesis** abgeleitet ist. Die Klasse **Hypothesis** ist von **Interval** und **Timeless-Hypothesis** abgeleitet. **Interval** steuert den zeitlichen Aspekt bei, während in **Timeless-Hypothesis** u.a. die Unsicherheit mittels Konfidenzwerten modelliert wird. Im Anhang „Die Klassen in COOL“ auf Seite 166 ist die COOL-Implementation dieser Klasse zu finden.

INTEGRATOR. Der Integrator baut auf der CLIPS-Bibliothek auf. Er ist als C++-Klasse ausgeführt und implementiert ein Produktionssystem mit dynamischem (d.h. zeitlich begrenztem) Speicher und einem Regel-Konstruktor, wie er in Abschnitt „Regel-Konstruktor“ auf Seite 141 beschrieben wird.

ZUGRIFF AUF DIE FAKTEN- UND REGELBASIS . Die Methoden zur Manipulation der Fakten- und Regelbasis umfassen das Laden von Konstrukten auf einer Textdatei (load), das Löschen aller Konstrukte (clear), das Löschen aller Fakten (reset) und das Asserrieren neuer Instanzen (**newInstance**). Diese Methoden bilden die Schnittstelle zum CLIPS-Interpreter und bedienen sich der entsprechenden CLIPS-Funktionen.

Die Methode **newInstance** bekommt als Parameter eine Zeichenkette, die den Namen der Hypothesenklasse und die Komponenten der Instanz in einer Slot-Filler-Notation enthält, und durch diese Syntax definiert ist:

```
String = Klasse Slots Slots = {"(" Slotname Slotinhalt ")"}  
Klasse = <bekannte Klasse, z.B. "FBrol">  
Slotname = <bekannter Slot der Klasse>  
Slotinhalt = <zum Typ von Slotname passender Inhalt>
```

Zum Beispiel:

```
String= "FB3rol (begin-timestamp 10 50)(end-timestamp 24 80)  
(confidence 0.8)"
```

Die Syntax ist bewußt an COOL angelehnt, damit die Zeichenkette mit wenigen Modifikationen an den CLIPS-Parser weitergegeben werden kann. Dies erspart eine eigene Parsing-Routine.

DYNAMISCHER SPEICHER. In CLIPS ist die Faktenbasis nur durch expliziten Zugriff (Zufügen, Löschen, usw.) veränderbar, eine zeitliche Dynamik ist nicht enthalten. Diese ist mit der Methode `applyDecayFunction` des Integrators implementiert. In der Methode wird eine Abklingfunktion auf jede Instanz im Gedächtnis angewendet, die **Timeless-Hypothesis** als Oberklasse hat und nicht als persistent markiert ist. Die Funktion modelliert eine Abnahme der Konfidenz, anders ausgedrückt, eine Abnahme der Präsenz einer Hypothese im Gedächtnis. Das Argument der Funktion ist das Alter einer Hypothese. Liefert die Funktion einen negativen Wert, dann ist die Hypothese überaltert, und wird aus dem Gedächtnis entfernt. Im einfachsten Fall ist es eine Schwellwertfunktion, die 1 ergibt für Argumente, die kleiner als die Speicherdauer sind, und andernfalls -1 . Es läßt sich aber auch eine kontinuierliche Abnahme der Konfidenz durch die Wahl einer anderen Funktion einstellen. Neben der Schwellwertfunktion ist daher noch eine sigmoide-Funktion und eine lineare Funktion wählbar.

Der Algorithmus der Methode `applyDecayFunction` in Pseudocode:

```

WHILE NOT alle Instanzen von Timeless-Hypothesis bearbeitet DO
  nimm nächste Instanz von Timeless-Hypothesis -> hypo
  IF hypo nicht persistent THEN
    Alter(hypo) = (aktuelle Zeit - Assertionszeit (hypo))
    neue Konfidenz(hypo) = Basiskonfidenz(hypo) *
                          Abklingfunktion (Alter(hypo))
    IF neue Konfidenz(hypo) < 0.0 THEN
      lösche hypo
    ENDIF
  ENDIF
ENDWHILE

```

REGEL-KONSTRUKTOR. Der Regel-Konstruktor ist mittels der Methode `ruleConstructor` implementiert. Sie erwartet als Parameter einen Ausdruck, der das neue Symbol beschreibt, und einen Namen für das neue Symbol. Der beschreibende Ausdruck muß die folgende Syntax besitzen:

```

Ausdruck      = Methode Oberklasse {"(" Ausdruck ")"} | Klasse
Methode       = <COOL; liefert Instanz von "Oberklasse">
Oberklasse    = <COOL-Klasse>
Klasse        = <COOL-Klasse>

```

Beispiel:

Ausdruck = **PARALLEL Hypothesis LocStretched**
(SEQUENCE Hypothesis LocLeft LocRight)

Die Symbolbeschreibung kann beliebig geschachtelt sein. Der Regel-Konstruktor erstellt aus der Symbolbeschreibung die notwendigen CLIPS-Regeln. Die benutzten Methoden, im Beispiel sind das **PARALLEL** und **SEQUENCE**, müssen vom Systemdesigner als Strukturwissen in CLIPS implementiert werden. Sie sind Bestandteil der Wissensbasis.

Während der Ausführung des Regel-Konstruktors werden zwei besondere Datenstrukturen aufgebaut: Einerseits speichert der Algorithmus alle Basishypothesen der Regeln in einer Struktur ab, die somit alle benötigten Symbole erfaßt, andererseits werden alle definierten Symbole, also alle erzeugbaren Symbole, in einer weiteren Struktur abgelegt. Die benötigten Symbole gebraucht der Integrator als Eingabe, um die produzierbaren Symbole mittels der Regeln inferieren zu können.

7.8 Zusammenfassung

In diesem Kapitel wurden die Lösungen und Realisierungen der im vorhergehenden Kapitel auf theoretischer Basis aufgezeigten Probleme dargelegt. Das ganze System ist als Multiagenten-System realisiert und benutzt intern parallelisierte Agenten. Die Datenaufnahme erfolgt mittels spezieller elektromagnetischer Sensoren deren Werte über ein abstraktes Körpermodell verallgemeinert werden, so daß seine Parameter als „virtuelle Sensoren“ aufgefaßt werden können. Dieses Modell ist auf der Basis des MMC-Netzes realisiert. Auf den Daten des Körpermodells arbeiten die Merkmalerkennung, die für ein Merkmal eine Klasse von Symbolen emittieren können, die durch einen Zeitstempel, einem Konfidenzwert und dem Rohdatum, das die Erkennung auslöste, markiert sind. Dies geschieht in dem Formalismus einer Hypothesen-Nachrichtenklasse. Die mittels dieses Formalismus versendeten Symbole können von den nachgeschalteten Integratoren aufgrund einer Ihnen jeweils mitzugebenen Wissensbasis zu neuen Symbolen integriert werden, die wieder dem Formalismus genügen. Daher können die Integratoren nach Belieben kaskadiert werden. Eine wesentliche Eigenschaft des Integrationsrahmens in dem die Integratoren realisiert sind, ist der Regel-Konstruktor, der die Erkennungsregeln aus der Symboldefinition erzeugt. Auf diese Weise ist eine Infrastruktur realisiert, die nur auf die Merkmale und Symbole testet, die von der obersten Stufe nachgefragt wurden. Dabei wird die Definition dieser Symbole über die einzelnen Stufen der Integration dekomponiert und jeweils als Menge von Untersymbolen baumartig an die unteren Stufen weitergeleitet.

Kapitel 8

Anwendung

In den vorangehenden Kapiteln wurde ein Rahmensystem zur Gestenerkennung vorgeschlagen. In diesem Kapitel werden nun zwei Prototypen vorgestellt, die in diesem Rahmen stehen.

8.1 Einfache Gestenerkennung

In diesem Szenario zur einfachen Gestenerkennung (Prototyp A) sind die Agenten wie in Abbildung 56 (siehe auch Abbildung 39, „Vorschlag einer Systemarchitektur“ auf Seite 98) dargestellt konfiguriert. Da hier die Applikation durch eine einfache Anzeige

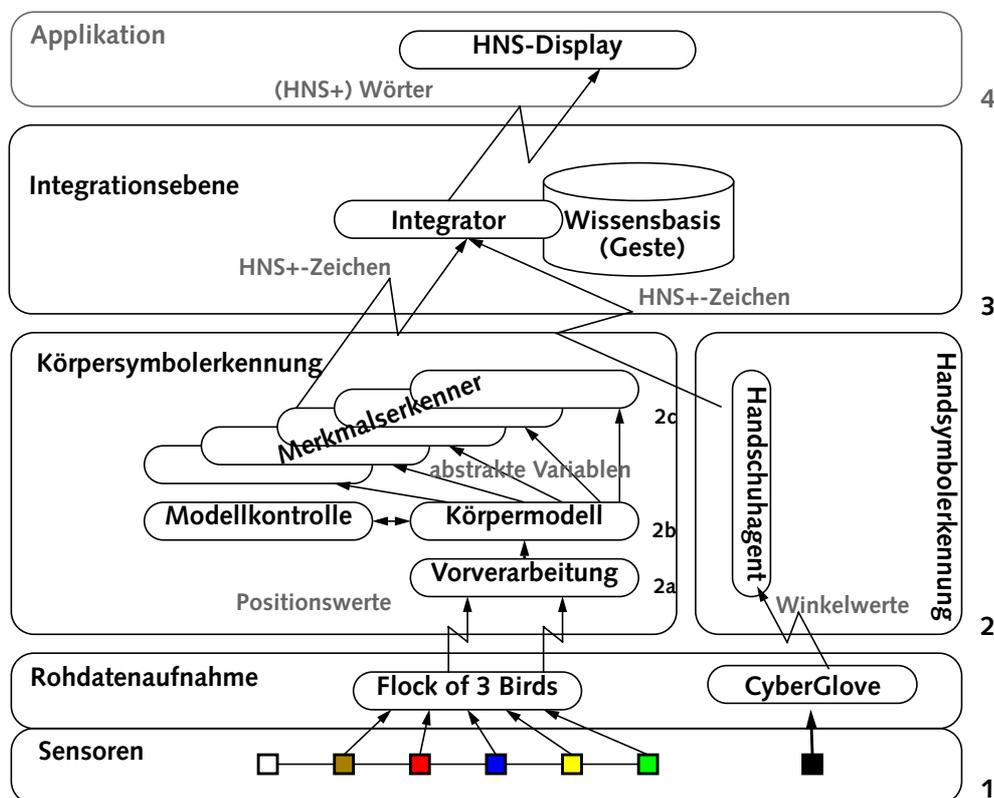


Abbildung 56: Prototyp A: Systemkonfiguration

simuliert wird, ist die de facto oberste Ebene der Integrator. Ihm kommt also die Aufgabe zu, das Gesamtsystem zu konfigurieren. Dazu wird ihm z.B. folgende Wissensbasis als sog. „Meta-Rules“ mitgegeben:

```
;; Time-stamp: <99/03/15 17:18:53 martinf>
;; Contains basic meta rules, indicating pointing, and waving
DISTRIBUTOR PointAndWave
[ BSifinger LocChest EFinA ] Pointing PARALLEL Hypothesis BSifinger LocChest EFinA
[ BSifinger LocChest EFinAU ] Pointing PARALLEL Hypothesis BSifinger LocChest EFinA
[ BSifinger LocChest EFinAD ] Pointing PARALLEL Hypothesis BSifinger LocChest EFinA
[ BSifinger LocChest EFinAL ] Pointing PARALLEL Hypothesis BSifinger LocChest EFinA
[ BSifinger LocChest EFinAR ] Pointing PARALLEL Hypothesis BSifinger LocChest EFinA
[ BSflat LocHead LocSide ] Waving PARALLEL Hypothesis BSflat LocHead LocRight
[ BSflat LocShoulder ] Waving PARALLEL Hypothesis BSflat LocShoulder
[ BSflat LocHead ] Waving PARALLEL Hypothesis BSflat LocHead
```

Die Zeichenfolge in den eckigen Klammern bezeichnen nur die Codefolge, die zur Ausgabe (engl.: „pretty print“) der in dieser Zeile definierten Hypothese Verwendung finden soll. Die eigentliche Definition beginnt danach mit dem Namen der hier überladenen Definition (siehe auch Abschnitt „Regel-Konstruktor“ auf Seite 141). In Kapitalen ist dann der Name der temporalen Relation angegeben, in der die Konstituenten vorliegen müssen. Daraufhin ist dann noch der Name der Oberklasse angegeben, von der sich die hier definierte Klasse ableitet. Alle weiteren Eintragungen sind Konstituenten, die selbst wieder aus um den Hypothesennamen verkürzten Definitionen (Rekursion) oder atomaren Hypothesenkonstituenten (Rekursionsverankerung), hier HNS+-Zeichen, bestehen. Wird in mehreren Zeilen ein und der selbe Name verwendet, so werden die Definitionen Überladen bzw. die aus ihnen generierbaren Erkennungsregeln mit einem logischen „oder“-Operator verknüpft.

Der Integratoragent verfügt so über eine Liste der von ihm verwertbaren Symbole, die er von den anderen Agenten anfordert („require“). Daraufhin aktivieren sich die Merkmalerkennungsagenten, die solche Hypothesen liefern können und tragen den Integratoragenten auf dem entsprechenden Verteile für diese Hypothesenklasse ein. Es sind also nur die Merkmalerkenner aktiv, die benötigt werden. Findet einer der aktiven Merkmalerkenner nun eine seiner Observablen in einem Bereich vor, der zu einem der angefragten Hypothesenklassen gehört, teilt er dies dem Integrator mittels einer zeitgestempelten Hypothese mit. Der Integrator kann nun die erhaltenen Hypothesen gemäß den aus der Wissensbasis generierten Regeln bearbeiten und eine neue Hypothese emittieren, die diesmal aus einer anderen, komplexeren Notationsssystem besteht (in diesem Beispiel nur mit den Marken „**Pointing**“ und „**Waving**“).

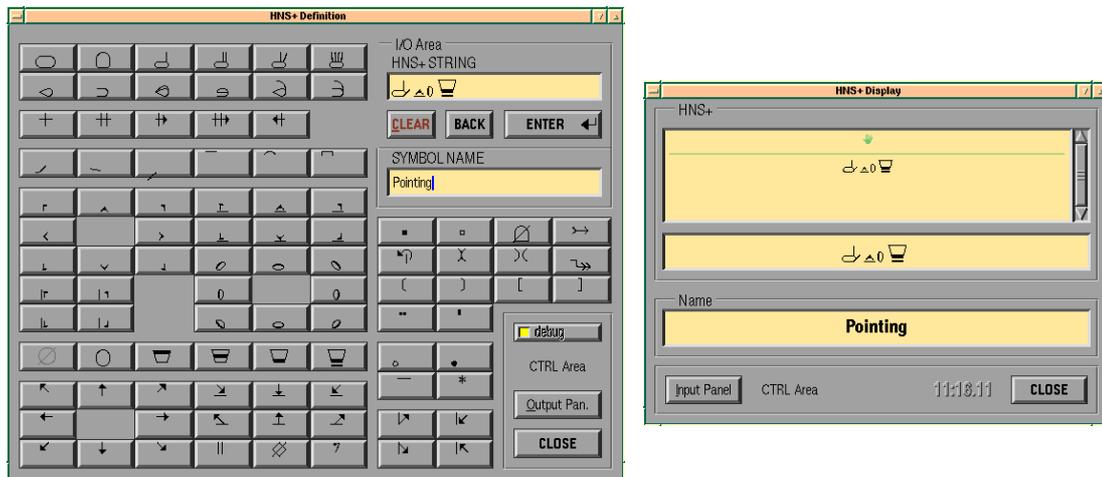


Abbildung 57: Symboldefinition zur Laufzeit

In Abbildung 57 ist die Symboldefinition zur Laufzeit dargestellt. Hier wird ein „Fingerschuß aus der Hüfte“ als „**Pointing**“ definiert. Dabei stellt „ \downarrow “ eine Handkonfiguration in Pistolenhaltung dar, „ \triangle “ bedeutet von „vom Körper weg“, „0“ eine Handfläche die nach links zeigt und „ \square “ bedeutet eine Handposition unter der Bauchhöhe (siehe auch Abschnitt „Hamburger Notationssystem“ auf Seite 47). Dies wird als „Meta-Rule“ „[BSifinger ThExt EFinA PalmL LocUnderstomach] Pointing Hypothesis PARALLEL BSifinger ThExt EFinA PalmL LocUnderstomach“ codiert zum Integrator gesendet, der daraus dann die Regel

```
(defrule MAIN::gen23-Pointing-rule
  ?gen24-match <- (object (is-a BSifinger))
  ?gen25-match <- (object (is-a ThExt))
  ?gen26-match <- (object (is-a EFinA))
  ?gen27-match <- (object (is-a PalmL))
  ?gen28-match <- (object (is-a LocUnderstomach))
  =>
  (PARALLEL Pointing "BSifinger ThExt EFinA PalmL
    LocUnderstomach" ?gen24-match ?gen25-match
    ?gen26-match?gen27-match ?gen28-match))
```

generiert, die bei Vorliegen von Hypothesen der Klassen „BSifinger“ „ThExt“ „EFinA“ „PalmL“ und „LocUnderstomach“ im Zeitfenster die Methode „**PARALLEL**“ aufrufen, die einen bestimmten Parallelitätsbegriff abtestet. Erfüllen die

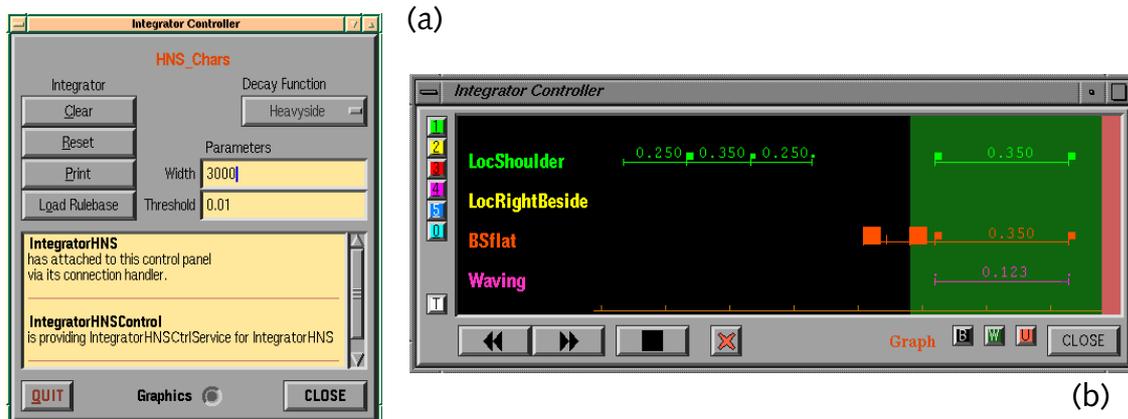


Abbildung 58: Integrator Visualisierung

Hypothesen die Bedingungen von „**PARALLELEL**“, wird dort dann eine neue Hypothese „**Pointing**“ berechnet, assertiert und jenen Agenten mitgeteilt, die sie nachgefragt („required“) hatten.

Eine Visualisierung des Integrationsvorganges liefert der Kontrollagent des Integrators, wie in Abbildung 58 zu sehen. Dabei wurde ein einfacheres Beispiel gewählt: Es wird „**Waving**“ aus „**LocShoulder**“ und „**BSflat**“ integriert. In Abbildung 58a ist das Fenster dargestellt über das der Integrator gesteuert wird und z.B. die Abklingfunktion (engl. Decay Function) gewählt und parametrisiert wird. In Abbildung 58b ist ein Schnappschuß des Visualisierungsfensters zu sehen, das bei jeder neuer Nachricht entsprechend dem Zeitstempel nach links rollt, um so den Zeitverlauf darzustellen. Dabei ist der Bereich, der im Integrationsfenster liegt leicht farblich hinterlegt. Der Zeitbereich zwischen dem „jetzt“ (rechter Bildrand) bis zu dem ersten Auftreten von Informationen stellt die Zeitverzögerung des Systems dar und ist etwas heller farblich hinterlegt. Das Zeitfenster wird vom linken Rand dieses Bereiches an gerechnet, um den Maschinenabhängigen Effekt zu kompensieren. Die einzelnen Hypothesen, deren Klassennamen am linken Bildrand verzeichnet sind, werden entweder als Quadrate oder Strecken dargestellt. Bei den Strecken wird das Hypothesengewicht textuell oberhalb der Strecke dargestellt, bei den Quadraten ist die Größe proportional zum Gewicht. Quadrate bezeichnen Hypothesen, bei denen nur der Beginn eines Ereignisses beschrieben wird, Strecken bezeichnen Zeiträume. Wie in Abschnitt „Eindeutigkeit von Hypothesen“ auf Seite 127 beschrieben ist, können zwei Zeitpunkte unter gewissen Bedingungen zu einem Zeitraum integriert werden. Zu beachten bleibt, das der Integrator Hypothesen uniform behandelt, also keinen prinzipiellen Unterschied

zwischen Eingabe- und Ausgabe-Hypothesen annimmt. In dem hier abgebildeten Beispiel werden die Hypothesengewichte multiplikativ zusammengefaßt, jedoch sind auch Verrechnungen durch Mittelwertbildung oder „Fuzzy Logic“ [KY95] je nach gewünschter weiterer Verwendung sinnvoll.

Bei der Implementation dieses Systems zeigte sich, daß die Integration stark von der Performanz des Agentensystems (und der Maschinen auf denen es implementiert ist) abhängt. Werden nämlich *einzelne* Nachrichten derart verzögert, daß sie erst hinter dem Integrationsfenster eintreffen, werden sie sofort gelöscht ohne für die Integration beachtet zu werden. Ein weiteres Problem ergibt sich, wenn mehr Nachrichten mit neuen Hypothesen eintreffen, als verarbeitet werden können. Zwar werden diese Hypothesen dann nach und nach abgearbeitet, aber es entsteht ein „Loch“ in der Erkennung, da einige Nachrichten durch das kontinuierliche Fortschreiten des Integrationsfensters verworfen werden. Verschärft wird dieses Problem durch die Tatsache, daß die Verarbeitungszeit überproportional zur Anzahl der Hypothesen im Speicher wächst. Daher muß bei der Dimensionierung des Systems strikt auf eine Minimierung der auszutauschenden Hypothesen hingearbeitet werden.

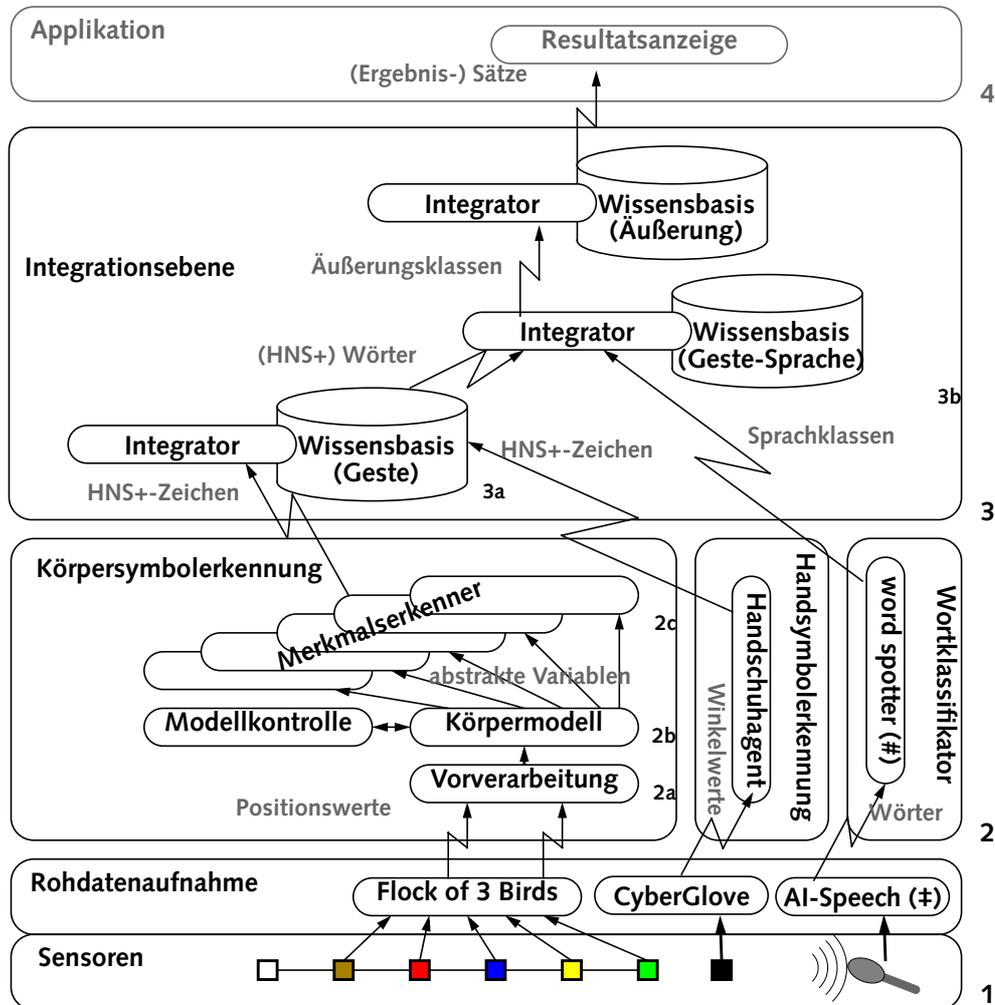
8.2 Einfache multimodale Integration

In diesem Szenario (Prototyp B), in dem aufgezeigt wird, das auch multimediale Integration prinzipiell möglich ist, sind die Agenten wie in Abbildung 59 auf Seite 148 (siehe auch Abbildung 39, „Vorschlag einer Systemarchitektur“ auf Seite 98) dargestellt konfiguriert.

In diesem Szenario wurde neben der Gestenverarbeitung (siehe Abschnitt „Einfache Gestenerkennung“ auf Seite 143) auch ein sprachanalytisches Werkzeug benutzt. Dabei handelt es sich um einen Forschungsprototypen eines Spracherkennungssystems der Arbeitsgruppe „Angewandte Informatik“ der Universität Bielefeld [FSKS98]. Die von diesem System inkrementell erkannten besten Wortketten werden via Netzwerk¹⁾ an einen sog. „word spotter“ geleitet. Dieser Agent klassifiziert die erhaltenen Wörter in mehrere Wortklassen mittels einer einfachen Hash-Tabelle (vgl. [CLR90], Chapter 12), wie im folgenden aufgeführt ist²⁾.

-
1. In diesem Falle eine direkte TCP-Socket Verbindung via IP, kein Multicast wie bei den Bewegungssensoren.
 2. Alle anderen Wörter werden stillschweigend ignoriert.
-

Die so ermittelten Hypothesen der Wordklassen „Object“, „Location“ und „Transaction“ (vgl. Abbildung 60 auf Seite 149) werden dann den Agenten, die sie anfordern („require“) zur Verfügung gestellt (siehe auch [SFL99]).



‡) Das mit „AI-Speech“ bezeichnete Modul ist ein Spracherkennungsprototyp der Arbeitsgruppe „Angewandte Informatik“, der Technischen Fakultät der Universität Bielefeld.

#) Das als „word spotter“ gekennzeichnete Modul bildet Wörter auf zuvor definierte Klassen ab.

— Die Integriationsebene ist nun mehrstufig.

Abbildung 59: Prototyp B: Systemkonfiguration

BEGIN OBJECT	BEGIN LOCATION	BEGIN TRANSACTION
das	da	stell
dieses	dort	stelle
teil	END	schieb
ding		schiebe
rohr		END
END		

Abbildung 60: Wortklassen und ihre Instanzen

Dabei wird ein extrem einfacher Satz von Regeln verwendet wie z.B. „**ObjectReference PARALLEL Hypothesis Pointing SpeechObject**“, die eine Referenz auf ein Objekt als Hypothese der Äußerungsklasse „**ObjectReference**“ durch das Zusammentreffen einer HNS+-Wortklasse („**Pointing**“) mit einer Sprachhypothese der Klasse „**Object**“ („**SpeechObject**“) beschreibt. Hypothesen der Äußerungsklassen können dann durch Regeln wie „**ptt_seq SEQUENCE Hypothesis ActionReference ObjectReference ObjectReference**“ in Ergebnissätze integriert werden, die dann angezeigt werden können oder von weiteren Agenten angefordert werden können.

Dieser stark vereinfachte Prototyp zeigt zwar, daß Information verschiedenster Quellen mit einander integriert werden kann, führt aber auch Probleme und Grenzen des Ansatzes vor Augen. Es wird klar, das es nicht einfach ist, das jeweilige Integrationswissen zu formulieren und über die Integratoren strukturiert zu verteilen. Dabei stellt sich zentral die Frage, welche Strukturebenen welche zeitlichen Randbedingungen haben — also wie die Abklingfunktion beschaffen und parametrisiert sein muß. Für die unteren Stufen der Integration wird dies noch verhältnismäßig gut durch die Modelle die im Kapitel „Zeitbegriffe“ auf Seite 31 dargelegt, für weitere Ebenen müssen weitere Forschungsergebnisse aus Linguistik und Psychologie herangezogen werden.

8.3 Zusammenfassung

In diesem Kapitel werden exemplarisch zwei mit dem Rahmensystem prototypisch implementierte Systeme vorgestellt, anhand derer die Funktionsprinzipien des Rahmensystems einerseits und der jeweiligen Aufgabe andererseits erläutert werden. Durch die beiden Systeme wird dargelegt daß das Rahmensystem an verschiedene Anforderungen anpaßbar ist und zur Laufzeit rekonfiguriert werden kann. Die Wissensbasen der einzelnen Integratoren sind voneinander getrennt und beschreiben Kommunikationsstruktur auf verschiedenen Ebenen. Es zeigt sich, das eine detaillierte

linguistische Kenntnis der zeitlichen Randbedingungen dieser Struktur zur Formulierung dieser Wissensbasen notwendig ist.

Kapitel 9

Résumé

In diesem Kapitel findet sich ein kritischer Rückblick mit einer Zusammenfassung, die noch einmal in kondensierter Form die einzelnen Aspekte der Arbeit unter verschiedenen Gesichtspunkten aufgreift, sowie ein Ausblick auf neue Forschungsfragen und nun beschreibbar erscheinende Wege gibt.

9.1 Wesentliche Systemeigenschaften

Das vorgestellte System unterscheidet sich von anderen Systemen durch das Zusammenspiel verschiedener Eigenschaften, die im folgenden noch einmal aufgeführt werden.

AUFGABENORIENTIERTE KOMMUNIKATIONSSTRUKTUR. Welche Information nun zur Erfüllung der Aufgabe der Applikation wirklich benötigt wird, kann diese in Form von Kompositionsregeln der Integrationsstufe mitteilen. Diese Kompositionsregeln werden in Integrationsregel umgeformt und aus diesen Integrationsregeln wieder Kompositionsregeln für nachfolgende Stufen generiert. Dies setzt sich bis auf die Ebene der Merkmalerkenner fort, die nur solche Merkmale aus dem Signalmodell extrahieren, die auch wirklich benötigt werden.

SCHNELLE DATENREDUKTION. Um die Integratoren, die als Produktionssysteme realisiert sind, performant zu halten, wird eine schrittweise Datenreduktion in den einzelnen Integrationsstufen vorgenommen. Zusammen mit dem sich ausdehnenden betrachteten Zeitintervall, läßt sich so ein beschränktes Gedächtnis realisieren, auf dem effiziente Suche als eine Vorbedingung für die Regelauswertung möglich ist.

SKALIERBARKEIT DER INTEGRATOREN. Die Symbolintegration ist hierarchisch organisiert. Dabei wird immer der gleiche Integrationsrahmen mit für die jeweilige Stufe spezifischem Integrationswissen gefüllt. Trotzdem ist der formale Type der Eingabe der gleiche, wie der der Ausgabe, so daß ein Integrator auf der Ausgabe eines anderen arbeiten kann.

INFORMATIONSTECHNOLOGISCH ADÄQUATE IMPLEMENTIERUNG. In diesem System ist eine Implementierung verwirklicht, die einzelne Zeit- und Prozeßflußverteilungsvorgänge

direkt auf Primitive des Betriebssystems abbildet und so eine Performanz erreicht, die für eine schritthaltende Bearbeitung hinreichend ist.

BEGRÜNDBARE SEMANTIK DER SYMBOLE. Die in diesem System verwendeten Symbole leiten sich von der Autorität der Experten des Instituts für Deutsche Gebärdensprache und Kommunikation Gehörloser in Hamburg her. Dies hat den Vorteil, daß so Expertenwissen modelliert werden kann und Probleme und Ergebnisse auf einer einem weiteren Personenkreis zugänglichen Basis diskutiert werden kann.

APPLIKATIONSUNABHÄNGIGKEIT DES ANSATZES. Das hier vorgestellte System ist so konzipiert, daß es seine Dienste in verschiedenen Szenarien anbieten kann. Einzig die Wissensbasis für die Integratoren muß jeweils an die entsprechenden Gegebenheiten angepaßt werden.

9.2 Lösungen

In den ersten Kapiteln konnte ein Beitrag zur Klärung der Struktur von Gesten durch die Vorstellung eines Versuches einer Taxonomie geleistet werden. Dabei wurden die Grundlagen zur Schärfung der Definition der Symbole gelegt, die in der Integration benutzt werden. Dadurch konnte eine Theorie zur Erzeugung fundierter und verständlicher Symbole für und in dem Integrationsmechanismus gelegt werden.

Der vorgestellte Integrationsmechanismus stellt eine Realisation eines merkmalsbasierten Ansatzes in Kombination mit einem wissensbasierten System dar. Dabei sind die zu integrierenden Symbole zur Laufzeit definierbar.

Auf der Ebene der Datenerfassung wurde das Konzept der „virtuellen Sensoren“, durch die Verwendung eines abstrakten Körpermodells realisiert. Dabei kommt das aus der Biokybernetik abgeleitete MMC-Netz zum Einsatz.

Als Implementationsbasis wird ein Multiagenten-System vorgeschlagen, das durch seine Nachrichtenstruktur und -semantik eine weitere Strukturebene realisiert und Daten sowie Konfigurationsanforderungen expliziert.

Die hier vorgestellten Systeme benutzen parallelisierbare Algorithmen, die mittels der im Anhang dargelegten Technologien zur Parallelisierung von Programmabläufen implementiert wurden.

9.3 Kritik

Diese Arbeit konnte nur den Grund legen für eine systematische Erforschung der merkmalsbasierten Gestenerkennung. Eine systematische Untersuchung der Anwendungstauglichkeit in verschiedenen Anwendungsszenarien steht aus, da das System nur in Form der im Kapitel „Anwendung“ auf Seite 143 dargestellten Prototypen getestet wurde. Auch wäre es wünschenswert Lernverfahren in der Integrationsebene anzuwenden, sie sollten neben der vorgegebenen Symboldefinition in der Lage sein, eigene Symbole zu kreieren. Dadurch könnten Symbole mit direkter Fundierung („grounding“), neben den vordefinierten entstehen und somit eine robustere und an den jeweiligen Benutzer besser angepaßte Erkennung ermöglichen.

In dem Integrationsrahmen bleibt die Behandlung der Vor- und Nachzeitigkeit aufwendig. Die Abwicklung über die beiden Funktionen **PARALLEL** und **SEQUENCE** ist zwar sehr allgemein, jedoch ist die **SEQUENCE**-Funktion aufgrund des Intervallkalküls recht komplex. Es wäre eine sinnvolle Arbeit, zu untersuchen, inwieweit sich diese Struktur implizit in die Struktur der Hierarchie der Integratoren überführen läßt, ohne das kognitionswissenschaftliche Grundmodell von Pöppel zu verletzen.

9.4 Abschließende Bemerkungen

Das in den letzten Jahren drastisch steigende Interesse an neuen Wegen in der Mensch–Maschine Kommunikation hat in kurzer Zeit an verschiedenen Stellen der Forschungswelt das Interesse auf Gesten und ihr Potential gelenkt. Als ein Indikator für diese Entwicklung kann die Schaffung einer neuen Konferenzreihe, den sog. Gestenworkshops, die nun in 18 monatigem Abstand einen Konzentrationspunkt für die wissenschaftliche Diskussion in Europa bieten, dienen. Dabei wird deutlich, daß nicht nur bekannte Verfahren auf die neue Domäne angewendet werden, sondern auch neue Wege beschritten werden. Auch in den USA finden vermehrt Workshops zu diesem Thema statt. Obwohl es mittlerweile viele Arbeitsgruppen gibt, die sich dem Thema Geste aus der einen oder anderen Perspektive annehmen, steht dieses Forschungsgebiet noch an seinen Anfängen. Es ist nicht mit dem Stand der Spracherkennung zu vergleichen, da die dort gefundenen Verfahren aus den beschriebenen Unterschieden von Sprache und Gestik nicht direkt übertragbar sind und nur für Zeichensprachen adäquat erscheinen. Welche Eingabeformen in einem System zu verwenden sind, ist eine Entscheidung, die nur durch exakte Analyse der Anforderungen und der Kenntnisse und Fähigkeiten der Benutzer zu entscheiden. Die Forschung auf dem Gebiet der alternativen Schnittstellen dient in erster Linie dazu neue Perspektiven und Erkenntnisse über eben jene Alternativen zu bieten.

Anhänge

1.1 Multimodal versus Multimedial

Lenzmann führt in [Len98], Abschnitte 5.2 und 5.3, die Begriffe „Modalität“, „Medium“, „multimediales System“ und „multimodales System“ systematisch ein. Ihre Definitionen seien hier kondensiert wiedergegeben.

Definition: Medium: Als Medium wird einerseits das materielle Objekt bezeichnet, das Information eines bestimmten Typs enthält, als auch den zur Übertragung notwendigen Kommunikationskanal [May93], [Len98].

Definition: Modalität: Eine Interaktionstechnik, die dem Benutzer erlaubt, ein bestimmtes Ziel zu erreichen. Dabei wird eine Modalität formal als ein Tupel bestehend aus einem physikalischen Gerät d mit einer Interaktionssprache L zu $\langle d, L \rangle$ definiert [NC95].

Definition: Multimodales System: Ein multimodales System ist durch seine Fähigkeit definiert, mit dem Benutzer über viele verschiedene Modalitäten zu kommunizieren, die so zusätzliche, ergänzende teilweise sogar redundante Informationen, die eine Auflösung von Mehrdeutigkeiten sowohl aus der Benutzer- wie auch aus der Systemsicht gestatten.

Definition: Multimedia-System: Analog ist ein Multimedia-System über seine Unterstützung verschiedener Medien definiert.

Eine einfache Aneinanderreihung oder Kombination von Modalitäten und Medien ist dabei aber nicht hinreichend, vielmehr muß deren Kooperation sorgfältig koordiniert werden, wofür sowohl kontextuelle als auch temporale Kriterien zu berücksichtigen sind.

1.2 Prinzipien der Softwareentwicklung

Allgemein lassen sich die folgenden Grundsätze für die Entwicklung von Softwaresystemen angeben.

MODULARITÄT. Das Gesamtsystem soll in mehrere möglichst eigenständige Module zerfallen, die getrennt entwickelt werden können.

AUFGABENTEILUNG. Gemäß dem alten Entwicklungsprinzip „Single Source — Single Source of Failure“, soll jede Funktion möglichst nur einmal implementiert sein.

SCHNITTSTELLEN. Die einzelnen Module müssen klar definierte Schnittstellen aufweisen.

„PRINCIPLE OF LEAST SURPRISE“. Alle Module sollen die ihnen zugewiesene Funktion, möglichst ohne Seiteneffekte erfüllen.

PARALLELITÄT. Es soll die Möglichkeit gegeben sein, daß Abläufe, die algorithmisch unabhängig sind auch tatsächlich parallel ausgeführt werden können.

PORTABLE CODE. Da sich insbesondere Forschungssysteme über ihre gesamte Nutzungsdauer in Entwicklung befinden und sich die Betriebssysteme ständig weiter entwickeln, ist es unabdingbar, das nach den strengen Grundsätzen der portablen Entwicklung gearbeitet wird. Auf diese Weise wird die rechner-system-übergreifende Entwicklung quasi automatisch mitgelöst.

1.3 Agentenorientierte Programmierung

Allein durch die Dezentralisierung von Agentensystemen ergeben sich eine Reihe von Vorteilen gegenüber den zentralen Ansätzen der traditionellen künstliche Intelligenz. Nach [Mue93] sind das im Einzelnen:

- *Robustheit:* Fällt ein Agent im System aus, so ist nicht das ganze System inoperabel, sondern nur Teile desselben. Verteilte Systeme können entschieden einfacher auf Ausfälle einzelner Einheiten vorbereitet werden.
- *Effizienzgewinn:* Durch die parallele Verarbeitung ist eine Verbesserung der Bearbeitungsgeschwindigkeit auf jeden Fall dann zu erwarten, wenn die Anwendung auch physikalisch verteilt ist, d.h. wenn sie auf mehrere Prozessoren aufgeteilt ist.
- *Offenheit:* Da Agenten in verteilten Systemen über definierten Schnittstellen kommunizieren, ist es ein leichtes, Agenten auszutauschen oder neue Agenten an das System anzukoppeln. Hierdurch wird auch eine Dezentralisierung der Entwicklungsarbeit möglich.
- *Vielfältigkeit:* Durch die verteilte Architektur sind VKI-Systeme nicht hinsichtlich ihrer Hardwareplattform oder Programmiersprache eingeschränkt. Unabhängig von der Modellierung sind verschiedene Operationalisierungsszenarien möglich.

- *Skalierbarkeit*: Innerhalb einer Domäne unterscheiden sich Anwendungsgebiete oft nur durch ihre Komplexität. Ein Vorteil ist es dann, wenn die Systeme skalierbar sind, d.h. an die Komplexität der Anwendung anpaßbar sind. Verteilte Systeme bieten gerade diese Möglichkeit, indem für einzelne Aufgaben bestimmte Agenten modelliert werden, deren Anzahl ohne Probleme verändert werden kann. Außerdem können einzelne Agenten durch Untersysteme ausgetauscht werden und so deren Aufgabe weiter verteilen.
- *Wiederverwendbarkeit*: Einige Systemkomponenten können sehr leicht auch in anderen Systemen ohne Änderungen wiederverwendet werden.
- *Wissenserwerb*: Bei verteilten Systemen wird durch die Aufteilung der Domäne in Unterdomänen der Wissenserwerb erleichtert, da dieses die Spezifikationen der Gesamtdomäne einschränkt.
- *Adaptivität*: Agenten-Systeme sind schon alleine deshalb adaptiver als konventionelle Systeme, weil die Interaktion auf Kommunikation mit einem Partner beruht. Dadurch reagieren die Systeme flexibel auf Eingriffe durch menschliche Partner.
- *Flexibilität*: Bei steigender Agentenzahl erhöht sich die Emergenz (vgl. [Min85]), durch die man an dem System Verhaltensweisen beobachten kann, die nicht vordefiniert waren.

Diese Vorteile setzen allerdings für ein konkretes System eine sorgfältige Modellierung voraus. Ein System, das nicht über eine einheitliche Schnittstelle verfügt, wird schlecht erweiterbar sein, ebenso wie ein System keine Verbesserung der Effizienz liefert, wenn die Aktivierung der einzelnen Komponenten wie bei einem konventionellen System seriell erfolgt.

Klassisch werden bei der Agentenmodellierung folgende Ansätze verfolgt [Mue93]:

ZUSTANDSORIENTIERTE MODELLIERUNG. Bei der zustandsorientierten Modellierung befindet sich jeder einzelne Agent in einem Zustand, der sich durch Ereignisse ändern kann. Das gesamte System wird daher durch seinen Zustand beschrieben, der sich aus der Summe der Einzelzustände ergibt. Die Interaktionen der Agenten in dem System zeigen sich in dem Reagieren auf Ereignisse und dem Synchronisieren der Aktionen. Um ein System zustandsorientiert zu modellieren, müssen also die Zustände und deren Übergangsfunktionen beschrieben werden.

OBJEKTORIENTIERTE MODELLIERUNG. Im Gegensatz zur zustandsorientierten Modellierung sind die Zustände in der objektorientierten Modellierung nach außen hin nicht sichtbar, da sie in den Agentenobjekten gekapselt sind. Betrachtet werden demnach nur die Interaktionen unter den Agenten, die sich auf das Empfangen und Abgeben (Senden) beschränken. Durch Beschreibung dieser Kommunikationsprozesse wird das System modelliert.

ABSICHTENORIENTIERTE MODELLIERUNG. Bei der absichtenorientierten Modellierung handeln die Agenten motiviert, d.h. sie verfügen über Intentionen und Glauben. Agenten werden anhand ihrer mentalen Zustände beschrieben, die allerdings rein virtuell sind und nicht mit Zuständen von Maschinen zu verwechseln sind, mit denen z.B. die zustandsorientierte Modellierung realisiert werden kann. Zu diesen mentalen Zuständen werden Überzeugungen (beliefs), Verpflichtungen (commitments), Entscheidungen (decisions) und Fähigkeiten (capabilities) gezählt. Hier läßt sich auch der Begriff der BDI-Agenten festmachen: Sie arbeiten auf der Basis von Überzeugungen (Beliefs), Wünschen (Desires) und Absichten (Intentions).

ROLLENORIENTIERTE MODELLIERUNG. Dieser Ansatz ist bildet Begriffe der Soziologie auf Agenten ab: Die Agenten werden durch ihre Rollen innerhalb der Gesellschaft beschrieben. Ein Agent kann z.B. ein Erzeuger, Verbraucher, Vermittler, Händler oder Kunde sein. Da sich die Rollen aufgrund von Wechselwirkungen innerhalb des Systems auch ändern können, kann ein Agent auch mehrere Rollen zu einem Zeitpunkt innehaben (vgl. [Len98]).

Diese theoretischen Überlegungen können durchaus auch gleichzeitig angewendet werden, dieses wird um so wichtiger, je mehr Komplexitätsgrade der Information ein Agentensystem überdeckt. Rollenorientierte Modellierung ist adäquat für höhere, d.h. komplexere, Ebenen, während sie auf der Merkmalsebene mehr verschleiern würde, als zu einer durchsichtigen Struktur zu führen.

1.4 HamNoSys und SGIM

Aufzählungssymbole aus der SGIM-C++-Bibliothek, ihre numerische Entsprechung und der ihnen am meisten entsprechende HamNoSys-Bezeichner. Dabei ist zu beachten, daß die hier verwendeten Symbole Elemente von HNS+ und nicht HamNoSys sind und daß somit ihre Bedeutung leicht von HamNoSys abweichen, z.B. dadurch, daß sie in weit weniger Kontexten stehen können.

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
1	○	BSfist	
2	○	BSflat	
3	㇀	BSifinger	
4	㇁	BSimcfinger	
5	㇂	BSimofinger	
6	㇃	BSffinger	
7	㇄	BSdfistc	
8	㇅	BSdfisto	
9	㇆	BSdflatc	
10	㇇	BSdflato	
11	㇈	BSdifingerc	
12	㇉	BSdifingero	
13	∅	ThNorm	Auch leerer Glyph
14	↙	ThExt	
15	˘	ThAcr	
16	↘	ThDown	
17		OpNorm	Auch leerer Glyph
18	↙	OpWide	
19	˘	OpNarrow	
20	˘	FBangle	
21	˘	FBround	

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
22	𐀀	FBbent	
23	𐀁	FB2angle	
24	𐀂	FB2round	
25	𐀃	FB2bent	
26	𐀄	FB2rol	veraltet für FB2round
27	𐀅	FB2str	
28	𐀆	FB3angle	
29	𐀇	FB3round	
30	𐀈	FB3bent	
31	𐀉	FB3rol	veraltet für FB3round
32	𐀊	FB3str	
33	𐀋	FB4angle	
34	𐀌	FB4round	
35	𐀍	FB4bent	
36	𐀎	FB4rol	veraltet für FB4round
37	𐀏	FB4str	
38	𐀐	FB5angle	
39	𐀑	FB5round	
40	𐀒	FB5bent	
41	𐀓	FB5rol	veraltet für FB5round
42	𐀔	FB5str	

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente (Forts.)

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
43	◀	EFinL	
44	↵	EFinLU	
45	↳	EFinLD	
46	▶	EFinR	
47	↶	EFinRU	
48	↷	EFinRD	
49	^	EFinU	
50	∨	EFinD	
51	⤴	EFinA	
52	⤵	EFinAL	
53	↵	EFinAR	
54	∨	EFinT	
55	⤴	EFinTL	
56	⤵	EFinTR	
57	↵	EFinAU	
58	↳	EFinAD	
59	↵	EFinTU	
60	↳	EFinTD	
61	◊	PalmL	
62	◊	PalmLU	
63	◊	PalmLD	

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente (Forts.)

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
64	◊	PalmR	
65	◊	PalmRU	
66	◊	PalmRD	
67	◊	PalmU	
68	◊	PalmD	
69	○	LocHead	
70	☐	LocShoulder	
71	☐	LocChest	
72	☐	LocStomach	
73	☐	LocUnderstomach	
74	▪	LocSide	
75	◻	LocBeside	
76	◻◊	LocLeftBeside	
77	◊◻	LocRightBeside	
78	▪◊	LocLeft	
79	◊▪	LocRight	
80	↶	LocBackof	
81	χ	LocContact	
82)((LocNear	
83		LocNorm	keine HNS-Entsprechung
84	↷	LocStretched	

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente (Forts.)

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
85	←	MoveL	
86	↖	MoveLU	
87	↙	MoveLD	
88	→	MoveR	
89	↗	MoveRU	
90	↘	MoveRD	
91	↑	MoveU	
92	↓	MoveD	
93	↕	MoveA	
94	↖	MoveAL	
95	↗	MoveAR	
96	↘	MoveAU	
97	↙	MoveAD	
98	⊥	MoveT	
99	↙	MoveTL	
100	↘	MoveTR	
101	↖	MoveTU	
102	↙	MoveTD	
103	•	MoveLarge	
104	◦	MoveSmall	
105		MoveNorm	keine HNS-Entsprechung

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente (Forts.)

Id	HamNo-Sys-Glyph	SGIM-Name	Kommentar
106	*	MoveFast	
107	—	MoveSlow	
108	×	MoveTense	
109	7	MoveRest	
110		MoveHalt	
111	(BrackSeqL	
112)	BrackSeqR	
113	[BrackParL	
114]	BrackParR	
115	↘	ShapeC	
116	..	SymMirror	
117	:	SymParallel	
118	+	Repeat	
119	⦶	RepeatS	
120	↗	RepeatC	
121	↗⦶	RepeatSC	
122	↖⦶	RepeatR	

Tabelle 23: Bezeichner der SGIM-Library und ihre HamNoSys Äquivalente (Forts.)

1.5 Tabelle der Faktoren im Körpermodell

Drei Beispielmengen von Referenzwerten, die nach [Phe86], [Con72], [DGF59], [DIN33402T2], [NASA94] ermittelt wurden und vom Kontrollagenten nach der Verletzung von Konsistenzbedingungen zum Heilen dieser ausgetauscht werden.

Name	US-Bürger		Nordeuropäer		Mittelmeeranrainer	
	m	w	m	w	m	w
Stature [in mm]	1755	1625	1740	1640	1715	1600
EyeHeightFactor	0,936	0,938	0,936	0,933	0,936	0,938
ChinHeightFactor	0,870	0,875	0,870	0,875	0,870	0,875
ShoulderHeightFactor	0,819	0,825	0,830	0,825	0,814	0,825
ChestHeightFactor	0,725	0,725	0,725	0,725	0,725	0,725
StomachHeightFactor	0,630	0,632	0,632	0,624	0,621	0,632
ElbowHeightFactor	0,630	0,632	0,632	0,624	0,621	0,632
HipHeightFactor	0,530	0,524	0,511	0,505	0,511	0,524
WristHeightFactor	0,485	0,480	0,489	0,489	0,477	0,480
FingertipHeightFactor	0,357	0,370	0,357	0,370	0,371	0,370
Chin2TopOfHead-Factor	0,130	0,125	0,130	0,125	0,130	0,125
MaxHeadDiameterFactor	0,132	0,132	0,132	0,132	0,132	0,132
ShoulderBreadth-Factor	0,259	0,245	0,259	0,245	0,259	0,212
ShoulderElbow-LengthFactor	0,189	0,193	0,198	0,183	0,193	0,172
ElbowFingertip-LengthFactor	0,273	0,262	0,261	0,254	0,250	0,257
HandLengthFactor	0,128	0,110	0,118	0,119	0,106	0,108

Tabelle 24: Beispielwerte für konsistente Belegung des Körpermodells

Name	US-Bürger		Nordeuropäer		Mittelmeeranrainer	
	m	w	m	w	m	w
SittingHeightFactor	0,520	0,533	0,520	0,533	0,520	0,530
ChestDepthFactor	0,145	0,157	0,126	0,147	0,143	0,157

Tabelle 24: Beispielwerte für konsistente Belegung des Körpermodells (Forts.)

Die hier dargestellten Werte sind die Mittelwerte (50%-Perzentil, da Normalverteilung) für US-Bürger, Nordeuropäer, und (europäische) Bewohner des Mittelmeerraums. Wobei bei letzteren nur Relativdaten für den männlichen Bevölkerungsanteil vorliegen. Daher sind die Relativdaten für den weiblichen Bevölkerungsanteil aus allgemeinen Daten [Phe86] und von den US-Bürgerinnen kopierten Daten komplettiert worden.

1.6 Die Klassen in COOL

Die CLIPS-Notation ist stark an typischen Slot-and-Filler-Notationen aus der KI angelehnt. Für die Anwendung des Integrationsansatzes in einer bestimmten Domäne ist die Klasse **Hypothesis** abzuleiten und durch weitere domänenspezifische Slots zu ergänzen.

1.6.1 Timeless-Hypothesis

Klasse für zeitlose Hypothesen. **USER** ist die Generische Oberklasse in COOL. Von Klassen, die eine abstrakte Rolle haben, können keine Objekte instantiiert werden.

```
(defclass Timeless-Hypothesis (is-a USER)
  (role abstract)
  (slot persistent
    (type SYMBOL)
    (create-accessor write)
    (default NO))
  (slot confidence
    (type FLOAT)
    (create-accessor read-write))
  (slot exclusion-tag
    (type SYMBOL))
```

```
(default NONE)
(create-accessor read-write))
(multislot assertion-time
(create-accessor read)
(default-dynamic (hires_time))))
```

1.6.2 Interval

Klasse für ein Intervall in der Zeit. Der end-Zeitstempel wird bei der Instantiierung eines Objektes auf unendlich gesetzt, d.h. „bis auf weiteres gültig“.

```
(defclass Interval (is-a USER)
(role abstract)
(multislot begin-timestamp
(create-accessor read-write))
(multislot end-timestamp
(default (create$ INF INF))
(create-accessor read-write)))
```

1.6.3 Hypothesis

Von Klassen, die eine konkrete Rolle spielen, können durchaus Instanzen gebildet werden. Die letzte Zeile besagt, daß Objekte dieser Klasse in das pattern-matching mit einbezogen werden.

```
(defclass Hypothesis (is-a Timeless-Hypothesis Interval)
(role concrete)
(pattern-match reactive))
```

1.7 Thread-sichere Programmierung

Um in einer multi thread-Umgebung arbeiten zu können, sind im wesentlichen zwei Probleme zu Umgehen:

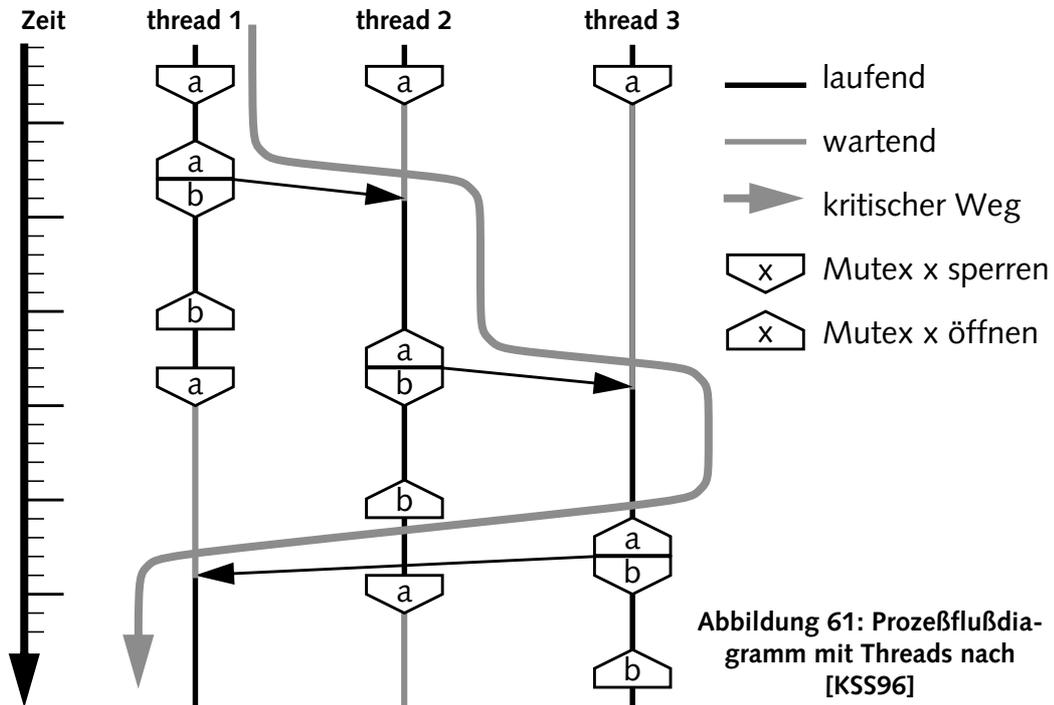
1. Der gleichzeitige Zugriff auf kritische Abschnitte („coherency violation“), und
2. das zyklische Warten auf die Freigrabe von kritischen Abschnitten („deadlock“).

Einer der in der Praxis am häufigsten vorkommenden kritischen Abschnitte, ist der zugriff auf Speicher, den mehrer Threads potentiell gemeinsam nutzen. Dabei heißt gemeinsam, daß zwar mehrere Threads auf den Speicher zugreifen, aber dies nie gleichzeitig.

Wo liegt das Problem beim gleichzeitigen Speicherzugriff? Es gibt mindestens zwei. Zum einen das bekannte Leser-Schreiber Problem: Der eine schreibt ein 2 Byte großes Datum, der andere liest „gleichzeitig“. Unter Umständen erhält der Leser ein Byte altes Datum und das andere Byte ist schon das neue Datum. Dies kann sehr fatal — z.B. im Kontext von Indizes und Zeigern — sein, wie mit leichter Überlegung schnell nachvollzogen werden kann. Das andere Problem liegt in der Art und Weise wie moderne Arbeitsplatzrechner den von-Neumann-Flaschenhals, den Adress- und Datenbus, realisieren. Da auch der dahinterliegende Speicher langsam oder teuer ist, wird Speicher in Zwischenspeicherhierarchien („cache hierarchies“) eingebaut, sodass schnell auf kleine Speichereinheiten zugegriffen werden kann, die Kopien von Abschnitten der darunterliegenden Speicherhierarchiestufe enthalten. Wird ein Datum außerhalb der Speichereinheit angefordert, wird die mit einer Kopie der Umfelds der angeforderten Adresse überschrieben („cache miss“). Nach „unten“ — in Richtung langsamerer Speicher — wird dies dann vom Betriebssystem bis auf die Größe von (typischerweise) 4 K-Blöcken auch auf die Festspeicher (z.B. Festplatte) abgebildet, sodass ein Datum im schlimmsten Fall von dem Festspeicher eingelesen werden muß („page fault“). Die beiden Ereignisse „cache miss“ und „page fault“ sind relativ zu den Zugriffsparametern der betrachteten Hierarchiestufe sehr teuer, d.h. es treten Faktoren von 2 bis 1000 auf. Nach „oben“ hin endet die Hierarchie bei den Registern der Instruktionspipelines, die meist parallel ausgelegt sind (vgl. „superscalar processor“ und „superpipelined processor“ z.B. in [DS98]), wie z.B. beim MIPS R8000. Um geschriebene Daten bei einem „cache miss“ nicht zu verlieren sorgen sog. Kohärenzverfahren für ein zurückschreiben („write back“) aller Daten in dem Zwischenspeicher auf eine der weiter „unten“ liegenden Stufen¹). Es läßt sich leicht ausmachen was passiert, wenn mehrere Threads auf einem Prozessor ohne den schützenden Rahmen des Programmkontextes „write backs“ auslösen. Bei genauerer Betrachtung ist dies auch das Leser-Schreiber-Problem, nur auf einer anderen Ebene. Das typische Resultat unter UNIX ist ein **sigbus**, ein „bus error“, der zur unbedingten, sofortigen Terminierung des Prozesses führt.

Um den gleichzeitigen Zugriff verhindern zu können, ist in Kontext der Threads auch mehrere Synchronisationsverfahren definiert. Zum einen Variablen zur Steuerung des gegenseitigen Ausschluß, („mutual exclusion“) Mutex genannt, zum anderen Bedingungsvariablen und zusätzlich die schon klassischen Semaphore (siehe [Tan94]) stehen auch zur Verfügung. Die Idee ist in jedem Falle die, für einen kritischen Abschnitt

1. Bei multiprozessor Systemen handelt es sich um ganze Kohärenzprotokolle, einem sehr komplexen Algorithmus.



einen Zustandsindikator zu haben, mittels dem zu entscheiden ist, ob ein Thread in den kritischen Abschnitt eintreten darf oder auf dessen Freigabe warten muß.

Ziel ist es mindestens einen durchgehenden Weg zu schaffen, den der Prozeßfluß beschreiten kann, den sog. kritischen Weg („critical path“). Idealerweise hat der kritische Weg genausoviele parallele Wegstrecken, wie parallele Prozessoren vorhanden sind. Das heißt, daß die Abarbeitung tatsächlich parallel vonstatten gehen kann. In Abbildung 61 ist ein Prozeßflußdiagramm dargestellt, in dem der Prozeßfluß über mehrer Threads an einigen Mutexen vorbeiläuft. Dieses Diagramm ist insofern idealisiert, da auch Threads dem normalen Algorithmus zur Zuweisung von Rechenzeit auf den Prozessoren unterliegen („scheduler“). Hier wird also eine auf einen Prozeß idealisierte Sichtweise benutzt.

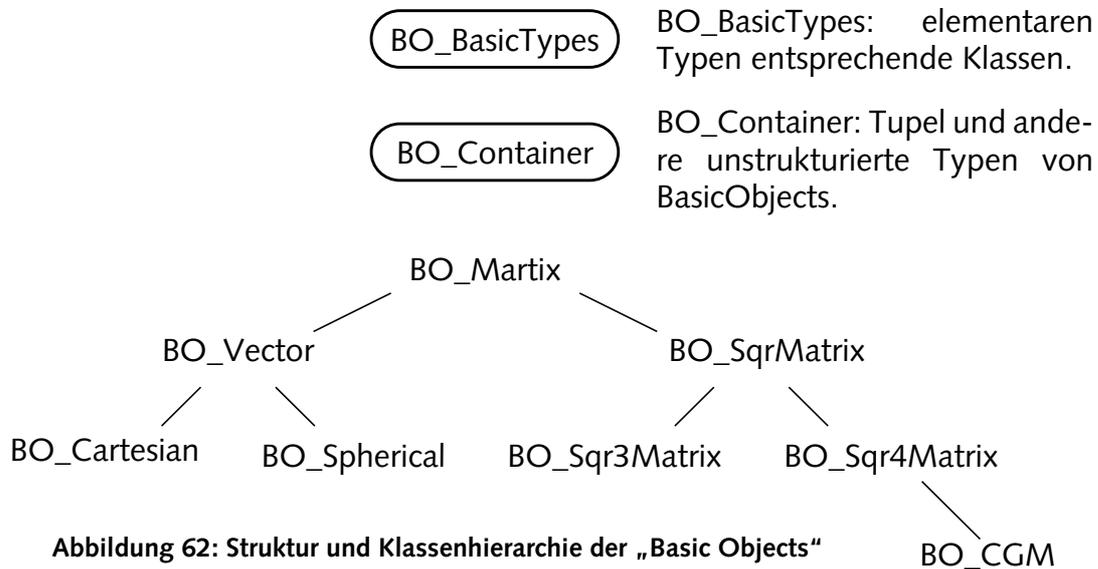
Wie dieses Diagramm deutlich macht, liegt die Verantwortung zur Vermeidung von „deadlocks“ bei dem Programmierer. Da die meisten Programmiersprachen keine Konstrukte zur Modellierung von Parallelität haben, muß also ein algorithmischer Weg besprochen werden. Die wichtigsten und auch einfachsten Verfahren sind:

„**PIPELINING**“. Die Aufgabe wird in einzelne Schritte zerlegt die nacheinander ausgeführt werden können. Dabei ist die Ausgabe eines Schrittes die Eingabe des nächsten Schrittes. Die Parallelisierungen entspricht der Stufigkeit der Pipeline.

EXPLIZITE PARALLELITÄT. Zwei Threads laufen Parallel ohne auf gemeinsame Daten zuzugreifen.

BARRIERE. Die Threads laufen alle auf einen expliziten Synchronisationspunkt, bei dem sie dann sequentiell auf strittige Ressourcen zugreifen.

Zur Realisierung des ausschließlichen Speicherzugriffs wurde die an der Arbeitsgruppe Wissensbasierte Systeme maßgeblich von Martin Hoffhenke und Martin Fröhlich entwickelte Klassenbibliothek namens „Basic Objects“ und die Bibliothek „Locked Basic Objects“ erweitert. In Abbildung 62 ist eine Übersicht über die Klassen der „Basic Objects“ gegeben. Dabei handelt es sich z.B. um Klassen zur Verarbeitung von mathematischen Matrizen. Für die häufig vorkommende untertypen spezialisiert wurden, wie z.B. die quadratischen Matrizen, die 4x4 Matrizen und die Transformationsmatrizen, wie sie in der Computergraphik und Robotik benötigt werden. In Abbildung 63 wird dargestellt, durch welches Vererbungsprinzip die „Locked Basic Objects“ geschaffen werden.



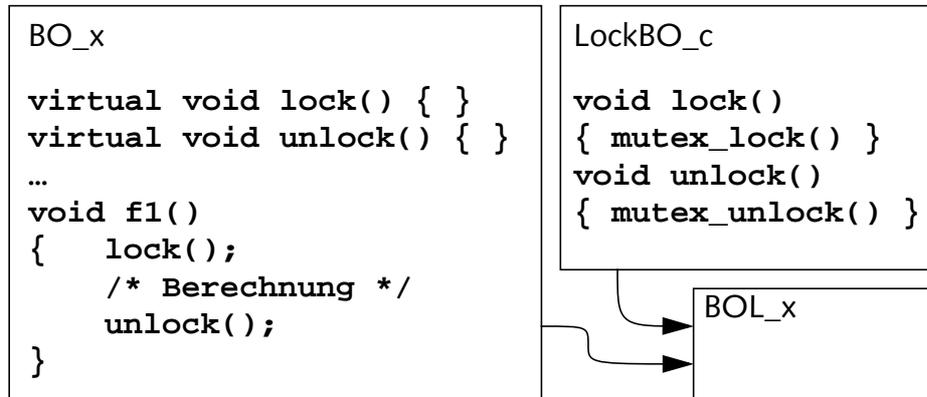


Abbildung 63: Klassenhierarchie eines „Locked Basic Objects“

Im Prinzip ist der Ausschlußmechanismus über die Funktionen `lock` und `unlock` immer schon in den Methoden der `BO_x` Klassen realisiert. Über Mehrfachvererbung werden jedoch die leeren Funktionen der `BO_x` Klassen in den `BOL_x` Klassen durch „scharfe“ Varianten ersetzt. Dabei benutzt `LockBO_c` das Mutex-Verfahren nicht direkt, sondern realisiert sog. „recursive locking“ (vgl. [KSS96]), das es einem Thread kontrolliert erlaubt, eine Marke mehr als einmal zu setzen und erst dann freigibt, wenn die Marke genauso häufig zurückgesetzt wurde. Das ist bei rekursiven Methoden erforderlich und so implementiert, da mit diesem Modell viel älter, d.h. längst bestehender und getesteter Code, erschlossen werden mußte.

Eine weitere Methode könnte auch durch eine eigene Implementierung des `new_handlers` erfolgen

1.8 Technik der Datenaufnahme

Vorliegend kommen drei bis fünf „Birds“ genannte Sensoren obigen Typs und ein sog. „Extended Range Transmitter“, ERT für einen erweiterten Meßbereich von 92 cm auf 305 cm von Fa. Ascension (s.o.) zum Einsatz. Diese sind jeweils über eine serielle EIA/TIA 232C vollduplex Leitung mittels einer 7-fach seriellen Schnittstellenkarte (Serial Parallel Controller (SP/C) X1008A von Fa. Sun Microsystems) an eine Sun SPARCstation 10 (sun4m) geleitet. Auf diesem Rechner kann nun ein Serverdaemon gestartet werden, der den ganzen Flock konfiguriert und dann die Meßdaten der einzelnen Birds empfängt. Der Flock wird hierbei in eine vom Hersteller vorgeschlagene Grundkonfiguration versetzt, bei der schon einige Filter und Verarbeitungsschritte hardwareseitig vorgenommen werden, was einerseits Auswirkungen auf die ermittel-

ten Daten hat, andererseits eine Verzögerung in der Datenausgabe von ca. 100 ms bis 500 ms zur Folge hat.

In einer realen Umgebung (siehe Abbildung 64) müssen zunächst einmal die Angaben des Datenblattes überprüft werden. Dazu wurde eine Meßreihe von 131 Meßstellen auf einem 6 x 8 x 4 (BTH)–Raster von demnach 192 möglichen Meßstellen benutzt. Einige der Meßstellen waren unzugänglich und konnten daher nicht benutzt werden. Die Meßpunkte, die dem Fußboden am nächsten waren — sie lagen in einer Höhe von 50 cm — wiesen eine dermaßen hohe Abweichung auf, daß sie für die zu berechnenden Abweichungsfunktion nicht benutzt wurden. Dieser Entschluß begründet sich mit folgender Überlegung:

Zum einen verletzen die Wände sowie Fußboden und Decke die Spezifikationen des Arbeitsbereiches der Sensoren, in denen es explizit heißt:



Abbildung 64: Labor für Künstliche Intelligenz und Virtuelle Realität in Bielefeld

„All specifications are valid at 30 °C ±10 °C in an environment void of large metal objects and electromagnetic frequencies, other than the power line.“ [Asc96], Seite 116

Doch die Wände des Labors bestehen aus einer Stützkonstruktion aus Aluminiumträgern, auf die Gipsplatten montiert sind, und Fußboden wie Decke bestehen aus Stahlbeton. Unter der Decke verlaufen große metallene Kabelhalter auf denen dutzende von Energie-, Netzwerk-, Video- und Telephonkabeln liegen und dazu noch ein Dreiröhrenvideoprojektor auf einem Stahlrohrgestell angebracht ist. Damit liegen die Regionen nahe Boden, Decke und Wänden nicht im Arbeitsbereich der Sensoren.

Zum anderen wird von einem aufrechtstehenden, erwachsenen Probanden ausgegangen. Befänden sich seine Handgelenke (tiefste mögliche Sensorposition wird im weiteren dargestellt) auf einer Höhe von 50 cm, so wäre der Proband nach bewährter Heuristik [Con72] ca. 133 cm groß¹⁾ und somit nicht in unserer Zielgruppe.

An den 131 Meßpunkten mit den durch Messung mit dem Maßband bekannten Koordinaten wurden jeweils zehn bis 15 Meßwerte in einem Abstand von ca. einer viertel Sekunde aufgenommen. Die so erhaltenen Trippelpaare von klassisch und mit den Sensoren gemessenen Raumkoordinaten wurden als Stützstellen für eine $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ Ausgleichsfunktion angesehen, und mittels dem Verfahren der kleinsten Fehlerquadrate wurden die Parameter einer quadratischen Funktionsgrundform bestimmt. Dabei errechnete sich folgende Funktion, die in Abbildung 65 graphisch dargestellt ist.

$$\vec{f}(\vec{x}) = \vec{a}x_1^2 + \vec{b}x_2^2 + \vec{c}x_3^2 + \vec{d}x_1x_2 + \vec{e}x_1x_3 + \vec{f}x_2x_3 + \vec{g}x_1 + \vec{h}x_2 + \vec{i}x_3 + \vec{j}$$

Die Elemente der einzelnen Koeffizientenvektoren errechneten sich dabei zu den Werten, die in Tabelle 25 auf Seite 175 dargelegt sind.

Ein weiteres Problem — neben dem oben beschriebenen Einfluß einer metallhaltigen Umgebung — ist das in den Daten enthaltene Rauschen. Hier wurde in einem einfachen Experiment jeweils einer der Sensoren in einer festen Position gehalten und seine Daten — hier 1762 Datensätze — über die Zeit aufgenommen. Die Graphen der Meßdaten finden sich in Abbildung 66 und Abbildung 67 auf Seite 176. Die offen-

1. Dies ist nach [Phe86] die Größe eines 9jährigen Jungen nach britischem 50% Quantil. Der Wert liegt im 0,1% Quantil für die Größe der weiblichen Bevölkerung Deutschlands

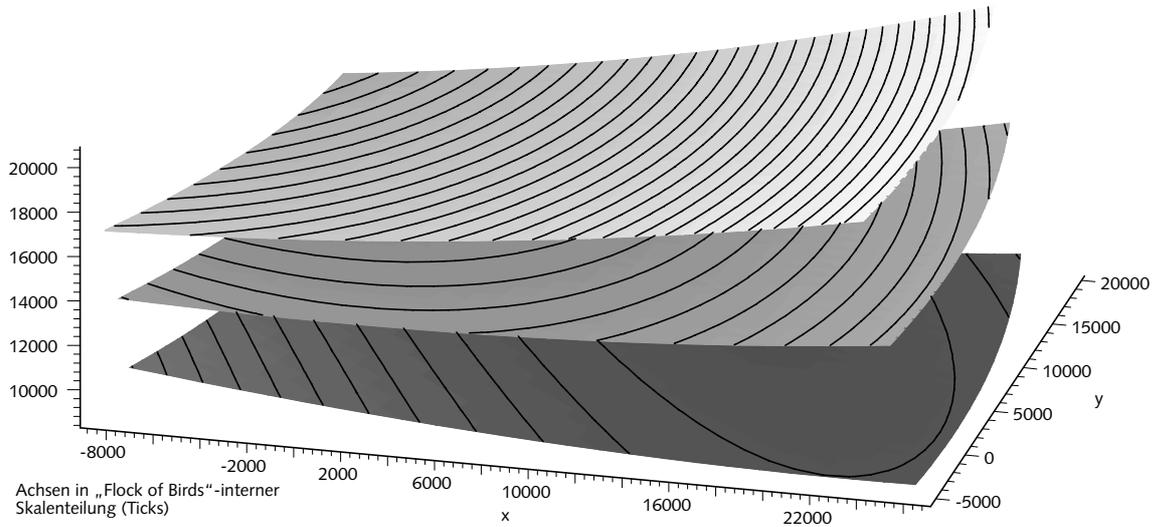


Abbildung 65: Graph der Ausgleichsfunktion nach [FW98]

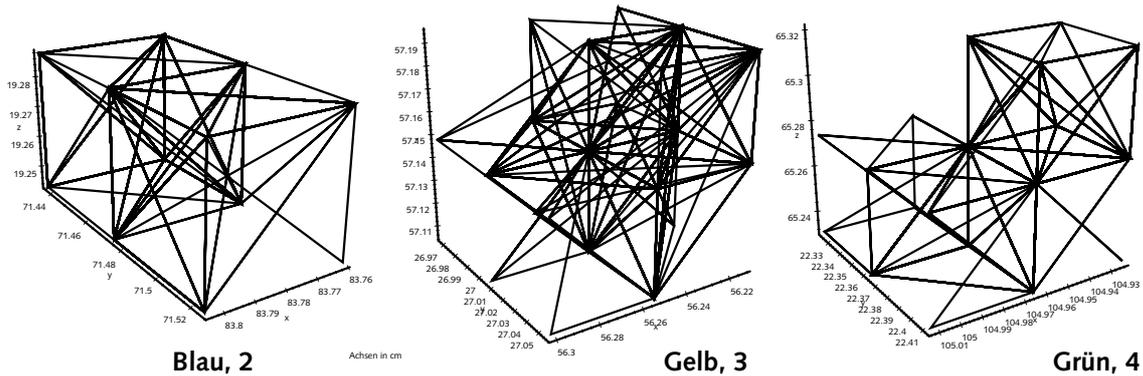


Abbildung 66: Meßdaten von ruhenden Birds, 3D-Sicht.

sichtliche Quantisierung der Werte ergibt sich aus der Tatsache, daß die Birds ihre Daten in einem Format liefern, in dem je 2 Bytes pro Dimension verwendet werden und jeweils ein Bit pro Byte für Synchronisationszwecke verschwendet wird. Das LSB (engl. „Least Significant Bit“) — also ein interner Skalenteil, im weiteren Tick genannt — hat den Wert $111,62 \mu\text{m}$, doch die beiden niederwertigsten Bits stehen immer auf Null, womit die hier zu beobachtende Quantisierungsweite von $[445,7 \mu\text{m}; 448,3 \mu\text{m}]_{99\%}^1)$ als eben 2 Bits, also 4 Skalenteile = $446,5 \mu\text{m}$ zu erklären ist. Die

	a	b	c	d	e	f	g	h	i	j
x	-2,06 *10 ⁻⁶	-5,56 *10 ⁻⁶	-6,47 *10 ⁻⁶	-6,40 *10 ⁻⁶	-3,58 *10 ⁻⁶	6,51 *10 ⁻⁶	1,079 *10 ⁻²	5,938 *10 ⁻²	-2,136 *10 ⁻²	-7,96 *10 ⁻²
y	1,070 *10 ⁻⁶	3,026 *10 ⁻⁶	4,430 *10 ⁻⁶	1,144 *10 ⁻⁷	2,369 *10 ⁻⁶	5,450 *10 ⁻⁶	3,269 *10 ⁻²	1,064 *10 ⁺⁰	9,103 *10 ⁻²	4,894 *10 ⁺²
z	3,128 *10 ⁻⁶	1,027 *10 ⁻⁶	1,702 *10 ⁻⁵	2,984 *10 ⁻⁶	2,172 *10 ⁻⁵	2,815 *10 ⁻⁷	1,390 *10 ⁻¹	4,716 *10 ⁻²	5,881 *10 ⁻¹	1,161 *10 ⁻³

Tabelle 25: Koeffizienten der Ausgleichsfunktion

Standardabweichungen für die drei mal drei Ortsparameter liegen zwischen 182 μm und 383 mm , also im Bereich von 3 bis 4 *Ticks*.

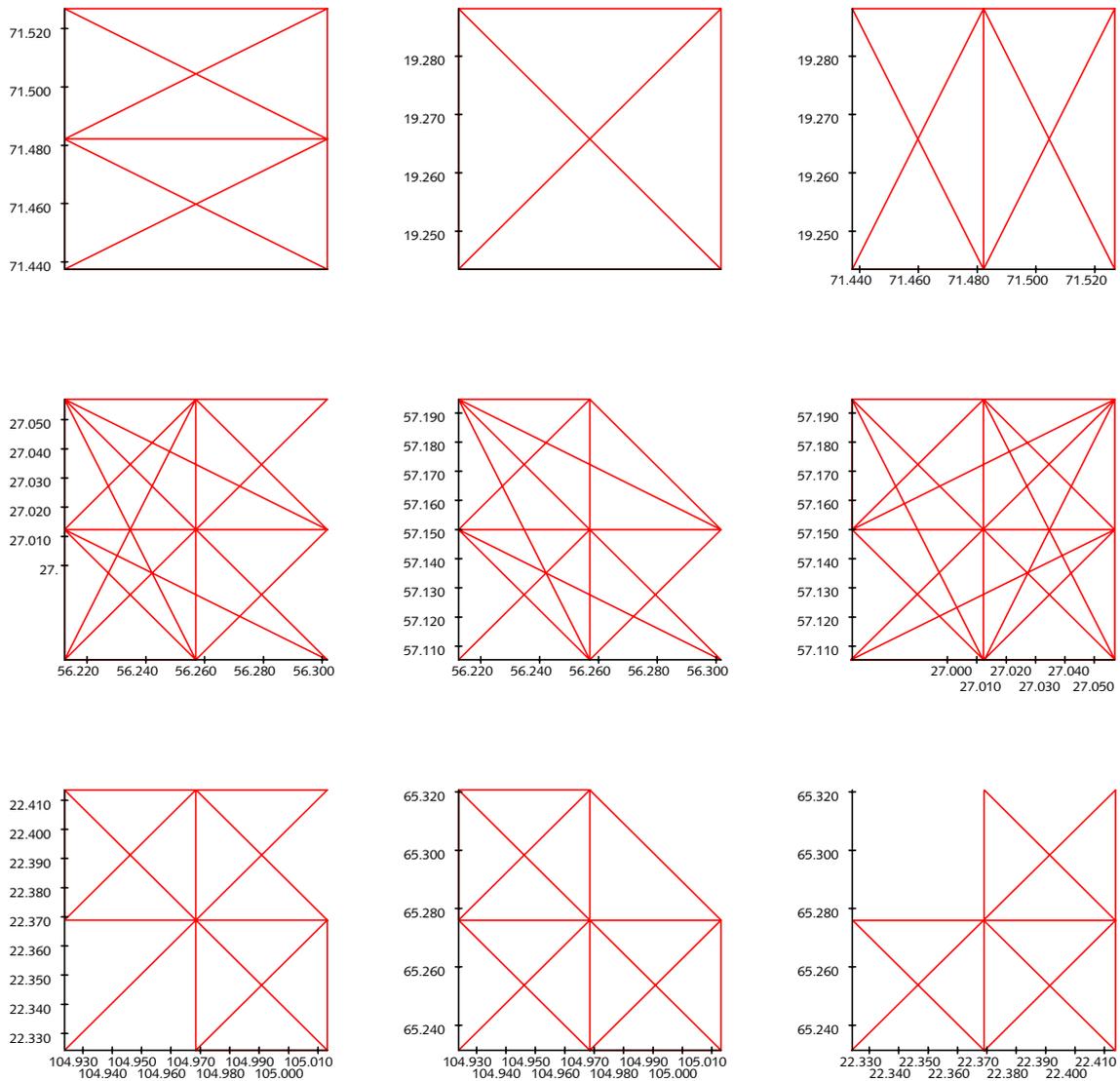
Wie in den Abbildungen zu sehen ist, ist diese Quantisierung aber nicht hinreichend, um das Signalrauschen durch die Quantisierung zu überdecken. Daher kommt ein diskreter Filter mit „low-pass“-Charakteristik und eine Rundung auf ca. 2 mm (16 *Ticks*) zum Einsatz.

1.9 Störungen bei der Zuweisung von Zeitstempeln

Da die Meßdaten in eigenen Geräten digitalisiert werden und danach erst einmal seriell zu dem Meßdatenrechner übertragen werden müssen, kommt es zu einigen Ungenauigkeiten bei der Zuweisung der Zeitstempel zu den Meßwerten. Die Zeitstempel markieren den Zeitpunkt nach dem Empfang des letzten Datums des jeweiligen Meßgerätes. Das Problem entsteht durch die Tatsache, dass die Daten zwar pseudo-parallel (mittels Thread-Technologie, siehe „Thread-sichere Programmierung“ auf Seite 167) abgefragt werden, aber andererseits nur eine CPU diese pseudo-Parallelität nur im Zeitscheibenverfahren emuliert. Ein sog. Scheduler verteilt die Zeitscheiben (nicht nur zwischen den eigenen Threads, sondern zwischen allen Threads in dem System). Dadurch kann der Moment der Datenaufnahme und der Moment der Zeitstempelgenerierung (Uhrbausteinabfrage) unterbrochen werden und somit Verzögerungen eingebaut werden. Der so entstehende Fehler insbesondere in den Geschwindigkeits-

-
1. Das Fehlerintervall ergibt sich aus Fehlern, die bei der Konvertierung von verschiedenen Darstellungen der Werte in andere Darstellungen auftraten. Hier: Internes Format, ganzzahlige Darstellung, Darstellung in cm.
-

daten ($\Delta x/\Delta t$) kann aber durch einen Tiefpaßfilter eliminiert werden, wie in Abbildung 68 auf Seite 177 motiviert wird.



Meßdaten von ruhenden Birds, verbunden gemäß des zeitlichen Verlaufes.

Spalten: Projektion auf die XY-, XZ- und YZ-Ebenen

Zeilen: Sensoren 2...4 bzw. Blau, Gelb, Grün

Abbildung 67: Meßdaten von ruhenden Birds, 2D-Projektion.

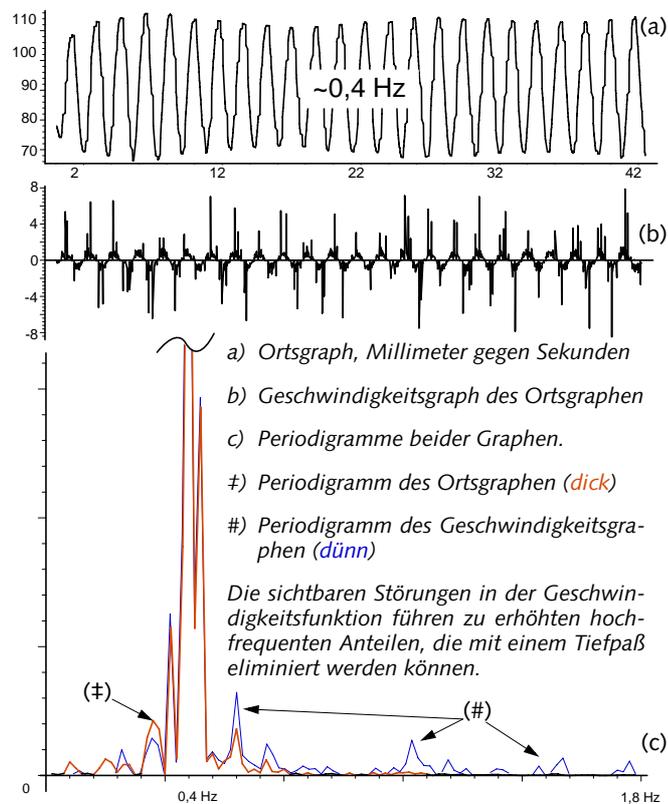


Abbildung 68: Fehler des Zeitstempels

1.10 Näherung der Handspannungsfunktion

Die Handspannungsfunktion wurde durch eine Reihe von generischen Polynomen der Grade 2 bis 5 genähert. Dabei wurden die Koeffizienten durch die Methode der kleinsten Fehlerquadrate berechnet.

In der Abbildung 69 auf Seite 178 ist ein Vergleich der Güte der Näherung bei variiertem Polynomgrad für den Zeigefinger dargestellt. Dabei ist die Fingerspannung einmal in der Originalfunktion von Harling und Edwards und ein mal in ihrer approximierten Form dargestellt. Die von dem Parameterschätzverfahren gelieferten Koeffizienten der verwendeten Funktionen sind in Tabelle 26 aufgeführt. Ein Polynom ist in einer Zeile dargestellt, zu lesen als Summe der Produkte aus Zeile und Spalte.

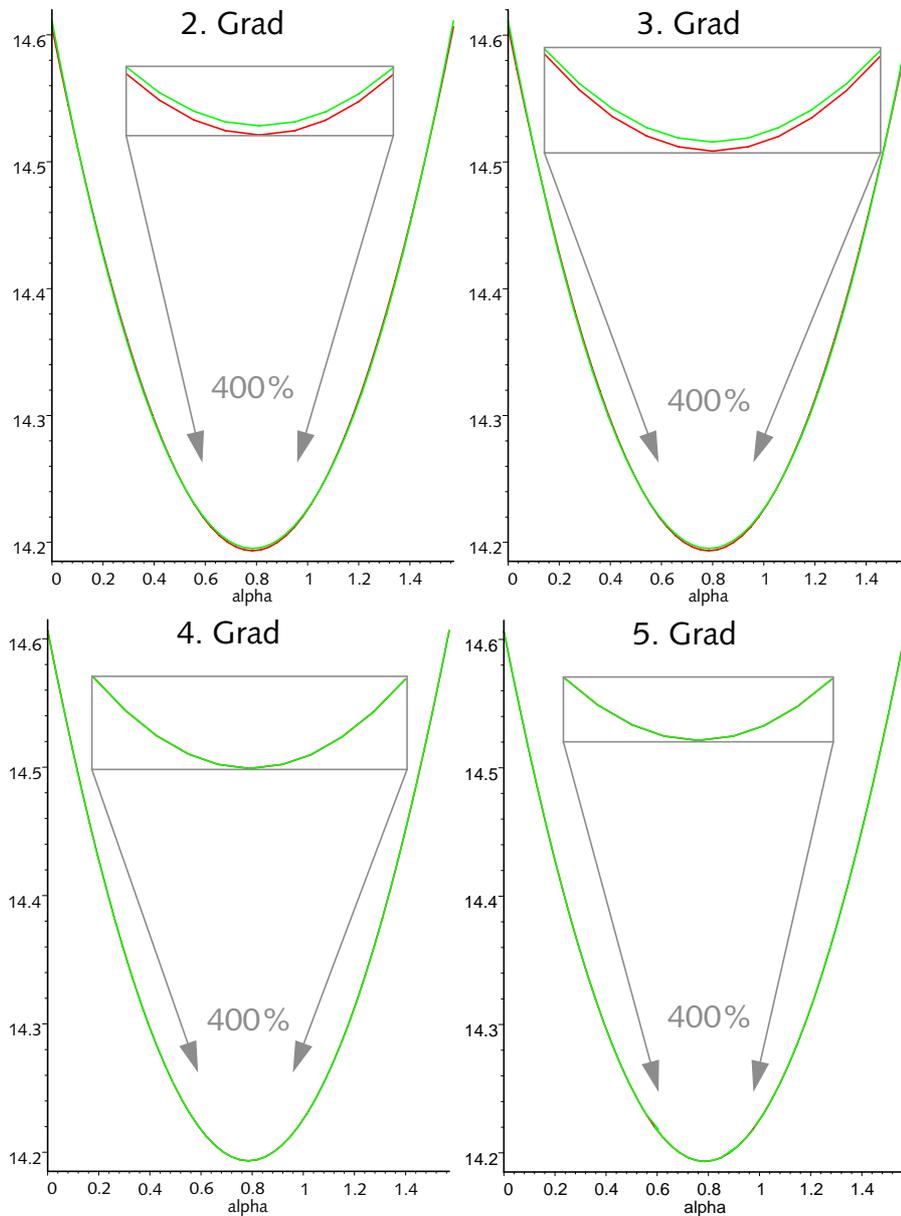


Abbildung 69: Güte der Näherungsfunktionen für den Zeigefinger

Die Graphen der Polynome 4. und 5. Grades und der der Originalfunktion sind im Rahmen der Darstellung deckungsgleich. Dies gilt analog auch für alle anderen Finger.

x5	x4	x3	x2	x1	x0
—	—	—	0,674423	-1,060103	15,81053
—	—	0,102376	0,677240	-1,061642	15,81074
—	-0,055698	0,174984	0,498085	-0,998270	15,80559
-0,000368	-0,054241	0,172935	0,499305	-0,998549	15,80560

Tabelle 26: Koeffizienten aller Polynome für die Zeigefingerspannung

Im Zuge der weiteren Vereinfachung, kann von der Fingerlänge abstrahiert werden und sie mit 1 angenommen werden. Die Koeffiziententabelle (analog zu Tabelle 26) für die universelle Fingerspannung ist in Tabelle 27 gegeben. Auch hier läßt sich be-

x5	x4	x3	x2	x1	x0
—	—	—	0,774321	-1,216636	0,299882
—	—	-0,003454	0,782611	-1,222114	0,300727
—	-0,059477	0,186873	0,582341	-1,145303	0,292823
0,167e-3	-0,060145	0,187841	0,581729	-1,145144	0,292811

Tabelle 27: Koeffizienten aller Polynome für die universelle Fingerspannung

obachten, daß die Näherungen mit Polynomen des 4. und 5. Grades die Darstellungsgenauigkeit übersteigt. In der Abbildung 70 auf Seite 180 sind, analog zu Abbildung 69 auf Seite 178 die Graphen für die Grade 3 und 4 dargestellt.

Der Vollständigkeit halber sind im folgenden die Näherungspolynome höheren Grades auch als C-Ausdrücke im Horner-Schema aufgeführt.

- Näherungspolynome 3. Grades für Daumen bis kleinen Finger:

$$(0.1461148956E2 + (-0.106100517E1 + (0.6768152139 - 0.1009233278E-2 * x) * x) * x) \quad (Gl. 18)$$

$$(0.1581074085E2 + (-0.1061642259E1 + (0.677240787 - 0.1023766886E-2 * x) * x) * x) \quad (Gl. 19)$$

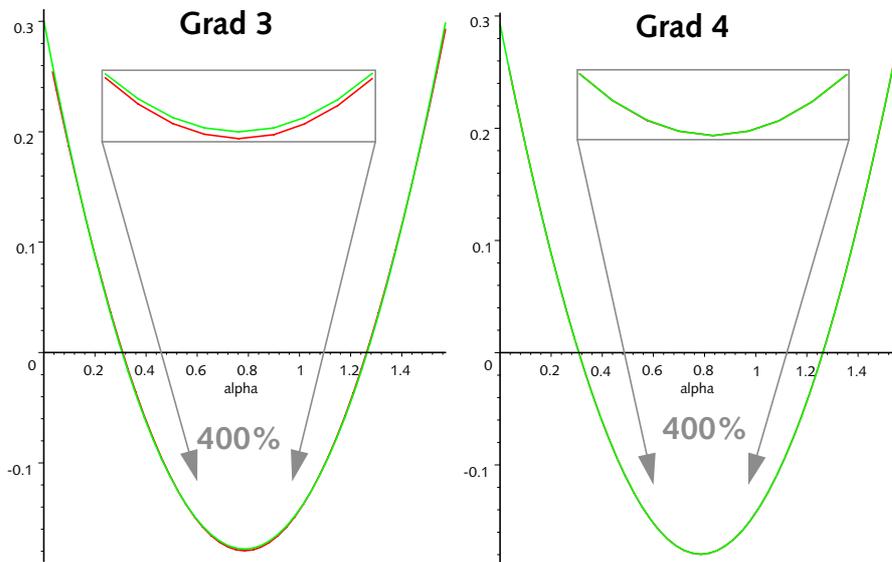


Abbildung 70: Güte der Näherungsfunktionen der universellen Fingerspannung

$$(0.1780975719E2+(-0.1062367108E1+(0.6777051045-0.1026116886E-2*x)*x)*x) \quad (Gl. 20)$$

$$(0.1621051725E2+(-0.1061794307E1+(0.677322457-0.1013108239E-2*x)*x)*x) \quad (Gl. 21)$$

$$(0.1081522648E2+(-0.1057160179E1+(0.6743176351-0.97208426E-3*x)*x)*x) \quad (Gl. 22)$$

- Näherungspolynome 4. Grades für Daumen bis kleinen Finger:

$$(0.1460637027E2+(-0.9980026849+(0.4987039682+(0.1739733678-0.5537424084E-1*x)*x)*x)*x) \quad (Gl. 23)$$

$$(0.1580559154E2+(-0.9982703429+(0.498085142+(0.1749848879-0.556989414E-1*x)*x)*x)*x) \quad (Gl. 24)$$

$$(0.178045641E2+(-0.9984563826+(0.497026218+(0.1764790226-0.5617251249E-1*x)*x)*x)*x) \quad (Gl. 25)$$

$$(0.1620534739E2+(-0.9981693889+(0.4974515635+(0.1756982313-0.5592130997E-1*x)*x)*x)*x) \quad (Gl. 26)$$

$$(0.1081034083E2+(-0.9970330797+(0.5043352531+(0.1660244422 - 0.5284700203E-1*x)*x)*x)*x) \quad (Gl. 27)$$

- Näherungspolynome 5. Grades für Daumen bis kleinen Finger:

$$(0.1460636057E2+(-0.9978264025+(0.4979328633+(0.175268556+(- 0.5629496312E-1+0.2330942469E-3*x)*x)*x)*x)*x) \quad (Gl. 28)$$

$$(0.1580560689E2+(- 0.9985493118+(0.4993054239+(0.1729352385+(-0.5424188838E-1- 0.3688741819E-3*x)*x)*x)*x)*x) \quad (Gl. 29)$$

$$(0.1780456484E2+(- 0.9984698422+(0.4970850936+(0.1763801321+(-0.561022133E-1- 0.1779726323E-4*x)*x)*x)*x)*x) \quad (Gl. 30)$$

$$(0.1620531385E2+(- 0.9975597526+(0.4947848575+(0.1801773703+(-0.5910543653E- 1+0.8061079897E-3*x)*x)*x)*x)*x) \quad (Gl. 31)$$

$$(0.1081036545E2+(- 0.9974805694+(0.5062926885+(0.1627366311+(-0.505097657E-1- 0.5917053993E-3*x)*x)*x)*x)*x) \quad (Gl. 32)$$

- Näherungspolynom 3. Grades der universellen Fingerspannung:

$$(0.300727946+(-0.1222114895E1+(0.7826113826-0.3454172875E-2 *x)*x)*x) \quad (Gl. 33)$$

- Näherungspolynom 4. Grades der universellen Fingerspannung:

$$(0.2928238904+(-0.1145303895E1+(0.5823419923+(0.1868739707- 0.5947754486E-1*x)*x)*x)*x) \quad (Gl. 34)$$

- Näherungspolynom 5. Grades der universellen Fingerspannung:

$$(0.2928115575+(-0.1145144268E1+ (0.5817294826+ (0.1878419215+ (-0.6014573717E-1+0.1670480776E-3 *x)*x)*x)*x)*x) \quad (Gl. 35)$$

Literaturverzeichnis

- Arb95 Michael A. Arbib. The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA, USA, 1995. ISBN 0-262-01148-4.
- ABF+94 P. Astheimer, K. Böhm, W. Felger, M. Göbel und S. Müller. Die virtuelle Umgebung — Eine neue Epoche der Mensch-Maschine-Kommunikation. Informatik-Spektrum, 17:281ff, 1994.
- ACM89 ACM Press. CHI'89 Conference Proceedings, New York, NY 10036, USA, 1989. ACM SIGCHI.
- Alt98 Frank Althoff. Entwicklung eines Basissystems für die Modellierung und interaktive Simulation einer artikulierten Figur. Diplomarbeit, Technische Fakultät der Universität Bielefeld, Juli 1998.
- ASW95 David F. Armstrong, William C. Stokoe und Sherman E. Wilcox. Gesture and the Nature of Language. Cambridge University Press, Cambridge, UK, 1995. ISBN 0-521-46772-1.
- Asc96 Ascension Technology Cooperation, P.O. Box 527, Burlington, Vermont 05402, USA. The Flock of Birds Position and Orientation Measurement System - Installation and Operation Guide, Juni 1996. Multiple Transmitter/ Multiple Receiver Configurations plus Extended Range Transmitter Manual.
- BPW93 Norman I. Badler, Cary B. Phillips und Bonnie Lynn Webber. Simulating Humans: Computer Graphics Animation and Control. Oxford University Press, Oxford, UK, 1993. ISBN 0-19-507359-2.
- Bal97 Dana H. Ballard. An Introduction to Natural Computation. MIT Press, Cambridge, MA, USA, 1997. ISBN 0-262-02420-9
- BBB+98 Hans-Joachim Boehme, Anja Brakensiek, Ulf-Dietrich Braumann, Markus Krabbes und Horst-Michael Gross. Neural Architecture for Gesture-Based Human-Maschine-Interaction. In Wachsmuth und Fröhlich [WF98], Seiten 219-232. ISBN 3-540-64424-5.

- Ben94 Alfred Benninghoff. Anatomie: makroskopische Anatomie, Embryologie und Histologie des Menschen, Bd. 1. Zellen- und Gewebelehre, Entwicklungsbiologie, Bewegungsapparat, Herz-Kreislauf-System, Immunsystem, Atem- und Verdauungsapparat. Urban und Schwarzenberg, München; Wien; Baltimore, 1994. ISBN 3-541-00245-X.
- BD92 Meera M. Blattner und Roger B. Dannenberg. Multimedia Interface Design. ACM Press, 1992. ISBN 0-201-54981-6.
- Blo92 John Bloomer. Power Programming with RPC. O'Reilly & Associates, Inc., Newton, MA, USA, 2. Auflage, September 1992. ISBN 0-937175-77-3.
- Bol80 Richard A. Bolt. Put-That-There: Voice and Gesture at the Graphics Interface. Proceedings of SIGGRAPH '80. ACM Press, New York, 1980.
- Bor94 Sven Bormann. Virtuelle Realität: Genese und Evaluation. Addison-Wesley, Reading, MA, USA, 1. Auflage, 1994. ISBN 3-89319-707-9.
- BC90 M. Brüwer und H. Cruse. A Network Model for Control of the Movement of a Redundant Manipulator. Biological Cybernetics, 62:549-555, 1990.
- BB78 B. Butterworth und G. Beattie. Gesture and Silence as Indicators of Planning in Speech. In Campbell und Smith, Hrsg., Recent Advances in the Psychology of Language. Plenum Press, New York, 1978.
- Bux90 Bill Buxton. The "Natural" Language of Interaction: A Perspective on Nonverbal Dialogues. In Laurel [Lau90], Kapitel New Directions, Seiten 405-416. ISBN 0-201-51797-3.
- CJW95 Yong Cao, Bernhard Jung und Ipke Wachsmuth. Situated Verbal Interaction in Virtual Design and Assembly. Zusammenfassung in [Mel95], Seiten 2061-2062.

-
- CBD93 H. Cruse, M. Brüwer und J. Dean. Control of Three- and Four-Joint Arm Movement: Strategies for Manipulator With Redundant Degrees of Freedom. *Journal of Motor Behavior*, 25(3):131-139, 1993.
- CC95 James L. Crowley und Joëlle Coutaz. Vision for Man Machine Interaction. In *Proceedings of Engineering Human Computer Interaction*, Seiten 28-45, Grand Targhee, USA, August 1995. EHCI'95, Chapman and Hall, Ltd. London, UK.
- Con72 Renato Contini. Body Segment Parameters, Part II. *Artificial Limbs*, 16(1):1-19, Spring 1972.
- CLR90 Thomas H. Cormen, Charles E. Leiserson und Ronald L. Rivest. *Introduction to Algorithms*. The MIT electrical engineering and computer science series. MIT Press, Cambridge, MA, USA, 1990. ISBN 0-262-03141-8.
- Cru86 Holk Cruse. Constraints for Joint Angle Control of the Human Arm. *Biological Cybernetics*, 54:125-132, 1986.
- CS93 Douglas E. Comer und David L. Stevens. *Internetworking with TCP/IP: Client - Server Programming and Applications; BSD socket version*, Bd. III. Prentice-Hall International, Englewood Cliffs, NJ 07632, USA, 1993. ISBN 0-13-474222-2.
- CS93 H. Cruse und U. Steinkühler. Solutions of the direct and inverse kinematic problems by a common algorithm based on the mean of multiple computations. *Biological Cybernetics*, 69:345-351, 1993.
- CWB+90 H. Cruse, E. Wischmeyer, M. Brüwer, P. Brockfeld und A. Dress. On the Cost Functions for the Control of the Human Arm Movement. *Biological Cybernetics*, 62:519-528, 1990.
- Dav90b Ernest Davis. *Representations of Commonsense Knowledge*. Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 1990. ISBN 1-55860-033-7.
- DGF59 Wilfried Taylor Dempster, W. Creighton Gabel und William J. L. Felts. *The Anthropometry of the Manual Work Space for the Seat-*
-

- ed Subject. American Journal of Physical Anthropologie, 17:289-317, 1959.
- DH55 J. Denavit und R. S. Hartenberg. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. ASME Journal of Applied Mechanics, 77:215-221, 1955.
- DHS94 Friedrich Dorsch, Hartmut Häcker und Kurt H. Stapf, Hrsg. Dorsch Psychologisches Wörterbuch. Hans Huber, Bern, Göttingen, Toronto, Seattle, 12. Auflage, 1994. ISBN 3-456-82497-1.
- DIN90 Deutsches Institut für Normung e.V. DIN, Hrsg. Bildschirmarbeitsplätze: Normen und Sicherheitsregel, Bd. 12 der DIN-Taschenbücher. Beuth, Köln, 1990. ISBN 0342-801X.
- DIN33402T1 Deutsches Institut für Normung e.V. DIN, Hrsg. DIN 33402 T1: Körpermasse des Menschen; Begriffe, Messverfahren, Kapitel 5, Seiten 153-163. Bd. 12 der [DIN90] , 1990. ISBN 0342-801X.
- DIN33402T2 Deutsches Institut für Normung e.V. DIN, Hrsg. DIN 33402 T2: Körpermasse des Menschen; Werte, Kapitel 5, Seiten 164-198. Bd. 12 der [DIN90], 1990. ISBN 0342-801X.
- DIN33402T2b Deutsches Institut für Normung e.V. DIN, Hrsg. DIN 33402 T2 Bbl 1: Körpermasse des Menschen; Werte; Anwendung von Körpermassen in der Praxis, Kapitel 5, Seiten 197-224. Bd. 12 der [DIN90] , 1990. ISBN 0342-801X.
- Dre95 Andrea Drees. Visuelle Erkennung von Handstellungen mit neuronalen Netzen. Dissertation, Technische Fakultät der Universität Bielefeld, AG Neuroinformatik, Bielefeld, August 1995.
- DS98 Kevin Dowd und Charles R. Severance. High Performance Computing. 3. Auflage, Juli 1998. ISBN 1-565992-312-X.
- Eco72 Umberto Eco. Einführung in die Semiotik. Number 105 der Uni-Taschenbücher. Wilhelm Fink Verlag, München, 1972. ISBN 3-7705-0633-2.

-
- Edw98 Alistair D. N. Edwards. Progress in Sign Language Recognition. In Wachsmuth und Fröhlich [WF98], Seiten 13-22. ISBN 3-540-64424-5.
- Efr41 David Efron. Gesture and Environments. King's Crown Press, Morningside Hights, New York, 1941.
- Efr72 David Efron. Gesture, Race and Culture, Bd. 9 of Approaches to Semiotics. Mouton, Morningside Hights, New York, 1972.
- Fau93 Oliver Faugeras. Three-dimensional computer vision: a geometric viewpoint. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-06158-9.
- FH95 Sidney Fels und Geoffrey Hinton. Glove-TalkII: An Adaptive Gesture-to-Formant Interface. In Katz et al. [KMM+95], Seiten 456-463. ISBN 0-89791-694-8.
- FHP+81 S. Frey, H.-P. Hirsbrunner, J. Pool und W. Daw. Das Berner System zur Untersuchung nonverbaler Interaktion: I. Die Erhebung des Rohdatenprotokolls. In Winkler [Win81] , Seiten 203-236. ISBN 3-476-00451-1.
- For82 C. L. Forgy. RETE - A fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. Artificial Intelligence, 19(1):17-37, 1982.
- Fra98 Liutger Franzen. Der Kommunikationsbegriff in Multiagentensystemen und seine Verwendung zum Entwurf einer agentenorientierten Middleware. Diplomarbeit, Arbeitsgruppe Wissensbasierte Systeme der Technischen Fakultät der Universität Bielefeld, September 1998.
- FSKS98 G.A. Fink, C. Schillo, F. Kummert und G. Sagerer. Incremental Speech Recognition for Multimodal Interfaces. In IECON'98 - Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society [IECON98], Seiten 2012-2017. ISBN 0-7803-4503-7.
-

- Fro95 Martin Fröhlich. Richtungserkennung dynamischer Handgesten für multimodale Mensch-Maschine Kommunikationen: Gesten für Viena. Diplomarbeit, Technische Fakultät der Universität Bielefeld, August 1995.
- FW98 Martin Fröhlich und Ipke Wachsmuth. Gesture Recognition of the Upper Limbs: From Signal to Symbol. In Wachsmuth und Fröhlich [WF98]
- Gia94 Joseph G. Giarratano. C Language Integrated Production System, Version 6.03. NASA: U.S. National Aeronautics and Space Administration, Software Technology Branch, Lyndon B. Johnson Space Center, Houston, TX 77058 U.S.A., August 1994. COSMIC Program No. MSC-22429 - MSC-22434.
- GR93 Joseph Giarratano und Gary Riley. Expert Systems: Principles and Programming. PWS Publishing Company, 2. Auflage, 1993. ISBN 0-534-93744-6.
- Goe93 Günther Görz, Hrsg. Einführung in die Künstliche Intelligenz. Addison-Wesley, Reading, MA, USA, 1. Auflage, 1993. ISBN 3-89319-507-6.
- GF99 Rachid Guerraoui und Mohamed E. Fayad. OO Distributed Programming Is *Not* Distributed OO Programming. Communications of the Association for Computing Machinery, 42(4):101-104, April 1999.
- HRF+96 A. Hardwick, J. Rush, S. Furner und J. Seton. Feeling It as well as Seeing It - Haptic Display within Gestural HCI for Multimedia Telematics Services. In Harling und Edwards [HE96b], Seiten 105-116. ISBN 3-540-76094-6.
- Har90 Stevan Harnad. The Symbol Grounding Problem. Physica D – Nonlinear Phenomena, 42:335-346, 1990.
- HE96 Philip A. Harling und Alistair D. N. Edwards. Hand Tension as a Gesture Segmentation Cue. In Harling und Edwards [HE96b], Seiten 75-87. ISBN 3-540-76094-6.

-
- HE96b Philip A. Harling und Alistair D. N. Edwards, Hrsg. Process in Gestural Interaction: Proceedings of Gesture Workshop '96, Berlin Heidelberg New York, 1997. Dep. of Computer Science, University of York, Springer-Verlag. ISBN 3-540-76094-6.
- HFF81 H.-P. Hirsbrunner, A. Florin und S. Frey. Das Berner System zur Untersuchung nonverbaler Interaktion: II. Die Auswertung von Zeitreihen visuellauditiver Information. In Winkler [Win81], Seiten 237-268. ISBN 3-476-00451-1.
- HBK99 Hermann Hienz, Britta Bauer und Karl-Freidrich Kraiss. HMM-Based Continuous Sign Language Recognition using Stochastic Grammars. In Annelies Braffort, Françoise Forest, Rachid Gherbi, Sylvie Gibet, James Richardson und Daniel Teil, Hrsg., Gesture Workshop '99, Proceedings. Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur - Centre National de la Recherche Scientifique, Université Paris-Sud, März 1999. Vorge stellt während des Gesture Workshop '99 in Gif-sur-Yvette, Frankreich. Im Druck, siehe: <http://www.limsi.fr/GW99/>.
- HG98 Hermann Hienz und Kristi Grobel. Automatic Estimations of Body Regions from Video Images. In Wachsmuth und Fröhlich [WF98], Seiten 135-145. ISBN 3-540-64424-5.
- HH96 Frank G. Hofmann und G. Hommel. Analyzing Human Gestural Motions using Acceleration Sensors. In Harling und Edwards [HE96b], Seiten 39-59. ISBN 3-540-76094-6.
- HHH98 Frank G. Hofmann, Peter Heyer und Günter Hommel. Velocity Profile Based Recognition of Dynamic Gestures with Diskrete Hidden Markov Models. In Wachsmuth und Fröhlich [WF98], Seiten 81-95. ISBN 3-540-64424-5.
- HHO+96 G. Hommel, F. Hofmann, E. Obermeier, T. Velten, R. Poser und T. Noll. Zwischenbericht. Interdisziplinäres Forschungsprojekt Gebärdenerkennung mit Sensorhandschuhen der Technischen Universität Berlin, Technische Universität Berlin, FB 13, Institut für Technische Informatik, Fachgebiet Prozeßdatenverarbeitung, Berlin, Juni 1996.
-

- HHP93 Christopher Habel, Michael Herweg und Simone Pribbenow. Wissen über Raum und Zeit, Kapitel 1.4. In Görz [Goe93], 1. Auflage, 1993. ISBN 3-89319-507-6.
- HRH+88 Hollan, Rich, Hill, Wroblewski, Wittenberg und Grudin. An Introduction to HITS: Human Interface Tool Suite. Technical Report ACA-HI-406-88, MCC, Austin, TX, USA, 1988.
- Hum95 John E. Hummel. Object Recognition. In Arbib [Arb95] , Kapitel III. — Articles, Seiten 658–660. ISBN 0-262-01148-4.
- Hut77 Ann Hutchinson. Labanotation or Kinetography Laban: The System of Analysing and Recording Movement. Theatre Arts Books, Dance Books, New York, London, 3. Auflage, 1977. ISBN 0-903102-81-1.
- HWM+92 W. Hill, D. Wroblewski, T. McCandless und R. Cohen. Architectural Qualities and Principles for multi-modal and Multimedia Interfaces. In M. Blattner und R. B. Dannenberg, Hrsg., Multimedia Interface Design. ACM Press, New York, NY 10036, USA, 1992.
- IEEE1003.1 Institute of Electrical and Electronics Engineers, Inc. (IEEE), International Organisation for Standardization (ISO), International Electrotechnical Commission (IEC). Information Technology - Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language], 2 Auflage, 1996. Reference number: ISO/IEC 9945-1:1996(E), ANSI/IEEE Std 1003.1, 1996 Edition.
- IECON98 IEEE. IECON'98 - Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Bd. 4, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, September 1998. ISBN 0-7803-4503-7.
- JLW98 Bernhard Jung, Marc Erich Latoschik und Ipke Wachsmuth. Knowledge Based Assembly Simulation for Virtual Prototype Modelling. In IECON'98 - Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society [IECON98], Seiten 2152-2157. ISBN 0-7803-4503-7.

-
- Jun98 Nils Jungclaus. Integration verteilter Systeme zur Mensch-Maschine-Kommunikation. Dissertation, Technische Fakultät der Universität Bielefeld, Oktober 1998. Veröffentlicht im „world wide web“ <http://Archiv.UB.Uni-Bielefeld.DE/disshabi/1998/0004.ps>.
- Kap84 I. A. Kapandji. Funktionelle Anatomie der Gelenke: Schematisierte und kommentierte Zeichnungen zur menschlichen Biomechanik, Band 1: Obere Extremität, Bd. 40 der Bücherei des Orthopäden. Stuttgart, 1984. ISBN 3-432-94231-1.
- KMM+95 Irvin R. Katz, Robert Mack, Linn Marks, Mary Beth Rosson und Nielsen Jakob, Hrsg. Human Factors In Computing Systems: CHI '95 Conference Proceedings, New York, NY 10036, USA, Mai 1995. ACM Press, ACM SIGCHI. ISBN 0-89791-694-8.
- KB81 Ellen Kreighbaum und Katharine M. Barthels. Biomechanics: A Qualitative Approach for Studying Human Movements. Burgess, Minneapolis, MN (USA), 1981. ISBN 0-8087-1155-5.
- Ken80 A. Kendon. Gesticulation and Speech: Two Aspects of the Process of Utterance. In M. R. Key, Hrsg., The Relation between Verbal and Non-verbal Communication. Mouton, The Hague, 1980.
- Ken86 A. Kendon. Current Issues In the Study Of Gestures. In Nespoulos, Rerron und Lecours, Hrsg., The Biological Foundations of Gestures: Motor and Semiotic Aspects. Lawrence Erlbaum Associates, Hillsday N.J., 1986.
- KHW95 G. Drew Kessler, Larry F. Hodges und Neff Walker. Evaluation of the CyberGlove as a Whole-Hand Input Device. ACM Transactions on Computer-Human Interaction, 2(4):263-283, Dezember 1995.
- KGH98 Sotaro Kita, Ingeborg van Gijn, Harry van der Hulst. Movement Phases in Sign and Co-Seech Gestures, and their Transcription by Human Coders in Wachsmuth, Fröhlich [WF98], Seiten 23–35, Springer-Verlag, 1998.
-

- KSS96 Steve Kleinman, Devang Shah und Bart Smaalders. Programming with Threads. SunSoft Press, Prentice Hall, 1996. ISBN 0-13-172389-8.
- KY95 Georg J. Klir und Bo Yuan. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1995. ISBN 0-13-101171-5.
- KSI93 The ARPA Knowledge Sharing Initiative External Interfaces Working Group. Draft specification of the KQML agent-communication language plus example agent policies and architectures. Enterprise Integration Technologies Corporation, Palo Alto, Calif., 1993. Draft accepted as specification.
- Ko94 Hyeongseok Ko. Kinematic and Dynamic Techniques for Analysing, Predicting, and Animating Human Locomotion. Dissertation, University of Pennsylvania, 1994.
- KST92 David B. Koons, Carlton J. Sparrell und Kristinn Rúnar Thórisson. Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures. In Intelligent Multimedia Interfaces [BD92], Kapitel 11, Seiten 257-276. ISBN 0-201-54981-6.
- KB82 James U. Korein und Norman I. Badler. Techniques for generating the goal-directed motion of articulated structures. IEEE Computer Graphics and Applications, 2(9):71-81, November 1982.
- KSU81 Adam Kendon, Thomas A. Sebeok und Jean Umiker-Sebeok, Hrsg. Nonverbal Communication, Interaction, and Gesture: Selections from Semiotica, Bd. 41 of Approaches to Semiotics. Mouton, The Hague, 1981. ISBN 90-279-3489-4.
- Lab95 Rudolf von Laban. Kinetografie - Labanotation: Einführung in die Grundbegriffe der Bewegungs- und Tanzschrift. Noetzel, Heinrichshofen-Bücher, Wilhelmshaven, 1995. ISBN 3-7959-0606-7.
- Lau90 Brenda Laurel, Hrsg. The Art of Human-Computer Interface Design. Addison-Wesley, Reading, MA, USA, Januar 1990. ISBN 0-201-51797-3.

-
- Lau90b Brenda Laurel. Interface agents: Metaphors with character, Kapitel New Direction, Seiten 355- 365. In [Lau90], Januar 1990. ISBN 0-201-51797-3.
- Len98 Britta Lenzmann. Benutzeradaptive und multimodale Interface-Agenten, Bd. 184 der Dissertationen zur Künstlichen Intelligenz. Dissertation, Technische Fakultät der Universität Bielefeld. Infix, Sankt Augustin, März 1998. ISBN 3-89601-184-7.
- Lin93 Thomas E. Linehan, Hrsg. Computer Graphics Visual Proceedings, Annual Conference Series, New York, NY 10036, USA, 1993. ACM Press, ACM SIGGRAPH. ISBN 0-89791-602-6.
- LJ80 G. Lakoff und M. Johnson. Metaphors We Live By. University of Chicago Press, Chicago, 1980.
- LU94 Jordan Lenarčič und Andreja Umek. Simple Model of Human Arm Reachable Workspace. IEEE Transactions on Systems, Man, and Cybernetics, 24(8):1239-1246, August 1994.
- LW96 Britta Lenzmann und Ipke Wachsmuth. Eine multimodale Eingabeagentur. In M. Thielscher und S.E. Bornscheuer, Hrsg., Fortschritte der Künstlichen Intelligenz, Seite 93. Dresden University Press, Dresden, 1996. Abstract.
- LRH85 Willem J. M. Levelt, Graham Richardson und Wido La Heij. Pointing and Voicing in Deictic Expressions. Journal of Memory and Language, 24(2):133-164, April 1985.
- LW98 Marc Erich Latoschik und Ipke Wachsmuth. Exploiting distant pointing gestures for object selection in a virtual environment. In Wachsmuth und Fröhlich [WF98].
- LFJ+98 Marc Erich Latoschik, Martin Fröhlich, Bernhard Jung und Ipke Wachsmuth. Utilize Speech and Gestures to Realize Natural Interaction in a Virtual Environment. In IECON'98 - Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society IECON98 , Seiten 2028-2033. ISBN 0-7803-4503-7.
-

- LMB92 John R. Levine, Tony Mason und Doug Brown. *lex and yacc*. O'Reilly & Associates, Inc., 981 Chestnut Street, Newton, MA 02164, USA, 2. Auflage, 1992. ISBN 1-56592-000-7.
- Loe96 S. Löbner. *Taxonomie*. Eintrag in Gerhard Strube, Barbara Becker, Christian Freksa, Udo Hahn, Klaus Opwis, Günter Palm, Hrsg., *Wörterbuch der Kognitionswissenschaft*, Seiten 720 – 721, Stuttgart, 1996, Klett-Cotta, ISBN 3-608-91705-5
- MK93 Pattie Maes und R. Kozierok. *Learning interface agents*. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, Seiten 459-465. AAAI Press/The MIT Press, 1993.
- May93 Mark T. Maybury. *Intelligent Multimedia Interfaces*. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-63150-4.
- McN92 D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, Chicago, 1992. ISBN 0-226-56132-1.
- MDB+95 Pattie Maes, Trevor Darell, Bruce Blumberg und Alex Pentland. *The ALIVE System: Full-body Interaction with Autonomous Agents*. In *Proceedings of Computer Animation Conference (April 19-21)*, Geneva, Switzerland, 1995. IEEE. ISBN 0-8186-7062-2.
- Mel95 Chris S. Mellish, Hrsg. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Bd. 2, Montréal, CA, August 1995. Morgan Kaufmann Publishers. ISBN 1-55860-363-8. ISSN 1045-0823.
- Mil98 David L. Mills. *Adaptive Hybrid Clock Discipline Algorithm for the Network Time Protocol*. *IEEE/ACM Transactions on Networking*, 6(5):505-514, Oktober 1998.
- Mil92 David L. Mills. *Network Time Protocol (Version 3): Specification, Implementation and Analysis*. Request for Comments: 1305, Internet Engineering Task Force, Network Working Group, IETF Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA,

-
- März 1992.
URL: <http://www.ietf.org/rfc/rfc1305.txt>.
- Min85 Marvin Lee Minsky. *The Society of Mind*. Simon and Schuster, New York, NY, USA, 1985. ISBN 0-671-65713-5.
- McN82 D. McNeill und E. Levy. *Conceptual Representations in Language Activity and Gesture*. In Jarvella und Klein, Hrsg., *Speech, Place, and Action*, Seiten 271-295. John Wiley, New York, NY, USA, 1982.
- Mue93 Jürgen Müller, Hrsg. *Verteilte Künstliche Intelligenz: Methoden und Anwendungen*. BI-Wissenschafts-Verlag, Mannheim, Leipzig, Wien, Zürich, 1993. ISBN 3-411-16181-7.
- NASA94 NASA: U.S. National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, TX 77058 U.S.A. *Man-System Integration Standards (MSIS): NASA-STD-3000 I-III Revision B*, August 1994.
- NC95 Laurence Nigay und Joëlle Coutaz. *A generic Platform for Addressing the Multimodal Challenge*. In Katz et al. [KMM+95], Seiten 98-105. ISBN 0-89791-694-8.
- NLB+97 Carol Neidle, Dawn MacLaughlin, Benjamin Bahan, Robert G. Lee und Judy Kegl. *The SignStream Project*. American Sign Language Linguistic Research Project. Technical Report Number 5, Boston University, Boston, MA, 1997.
- New90 Allen Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, USA, Dezember 1990. ISBN 0-674-92099-6.
- Nie83 Heinrich Niemann. *Klassifikation von Mustern*. Springer-Verlag, Berlin et al., 1983. ISBN 3-540-12642-2.
- Nor90 Donald A. Norman. *Why Interfaces don't work*, Kapitel Sermons, Seiten 209-219. In Laurel [Lau90], Januar 1990. ISBN 0-201-51797-3.
-

- NR98 Claudia Nölker und Helge Ritter. Detection of Fingertips in Human Hand Movement Sequences. In Wachsmuth und Fröhlich [WF98], Seiten 209-218. ISBN 3-540-64424-5.
- NR98b Claudia Nölker und Helge Ritter. Illumination Independent Recognition of Deictic Arm Postures. In [IECON98] - Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society IECON'98, Seiten 2006-2011. ISBN 0-7803-4503-7.
- OIAG94 Open Inventor Architecture Group, Silicon Graphics, Inc.. Open Inventor C++ Reference Manual: The Official Reference Document for Open Inventor, Release 2, Juli 1994. ISBN 0-201-62491-5.
- Pap90 Chrissostomos Papaspyrou. Gebärdensprache und universelle Sprachtheorie. Versuch einer vergleichenden generativ-transformationellen Interpretation von Gebärdensprache und Lautsprache sowie der Entwurf einer Gebärdenschrift, Bd. 8 aus Internationale Arbeiten zur Gebärdensprache und zur Kommunikation Gehörloser. Signum Press, 1990. ISBN 3-927731-07-2.
- Pau81 Richard P. Paul. Robot Manipulators. MIT Press, Cambridge, MA, USA, 1981.
- Phe86 Stephan Pheasant. Bodyspace: Anthropometry, Ergonomics and Design. Taylor & Francis, London u. a., 1986. ISBN 0-85066-325-0 (pbk).
- Pri89 Siegmund Prillwitz, Regina Leven, Heiko Zienert, Thomas Hamke und Jan Henning. HamNoSys Version 2.0: Hamburg Notation System for Sign Languages: An Introductory Guide, Bd. 5 of International Studies on Sign Language and Communication of the Deaf. Signum Press, 1989. ISBN 3-927731-01-3.
- Poe97 Ernst Pöppel. A hierarchical model of temporal perception. Trends in Cognitive Sciences, 1(2):56-61, Mai 1997.
- PS93 Ernst Pöppel und Dierk Schwender. Temporal Mechanisms of Consciousness. International Anesthesiology Clinics, 31(4):27-38, 1993.

-
- PS95 Ernst Pöppel und Kerstin Schill. Time Perception: Problems of Representation and Processing. In Arbib [Arb95], Kapitel III. — Articles, Seiten 987–990. ISBN 0-262-01148-4.
- PTV+92 William H. Press, Saul A. Teukolsky, William T. Vetterling und Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, UK, 2. Auflage, 1992.
- Rad96 F. J. Radermacher. Cognition in Systems. Cybernetics and Systems, 27:1–41, 1996.
- RH92 A. Rudnicky und A. Hauptmann. Multi-modal Interaction in Speech Systems. In M. Blattner und R. B. Dannenberg, Hrsg., Multimedia Interface Design [BD92], Seiten 147-172. ACM Press, New York, NY 10036, USA, 1992.
- RS91 B. Rimé und L. Schiaratura. Gesture and Speech. In Feldman und Rimé, Hrsg., Fundamentals of Nonverbal Behavior. Seiten 239-281. Press Syndicate of the University of Cambridge, New York, 1991.
- RY94 Johannes W. Rohen und Chihiro Yokochi. Anatomie des Menschen: photographischer Atlas der systematischen und topographischen Anatomie. Schattauer, Stuttgart, 1. korr. Nachdr., 3. Auflage, 1994. ISBN 3-7945-1501-3.
- Rui98 Jan–Peter de Ruiter. Gesture and Speech Production. Dissertation, Katholische Universität Nijmegen, MPI Series in Psycholinguistics, Band 6, Max–Planck Institut für Psycholinguistik, Februar 1998, ISBN 90-76203-05-9
- Sch97 Karl-Heinrich Schmidt. Wissensmedien für kognitive Agenten. Habilitationsschrift, Technische Fakultät der Universität Bielefeld, Oktober 1997.
- Sch95 Ernst Günter Schukat–Talamazzini. Automatische Spracherkennung — Grundlagen, statistische Modelle und effiziente Algorithmen. Künstliche Intelligenz. Vieweg, Braunschweig/Wiesbaden, 1995. ISBN 3-528-05492-1.
-

- Sho93 Y. Shoham. Agent-Oriented Programming. Artificial Intelligence, 60:51-92, 1993.
- ST91 Joseph W. Sullivan und Sherman W. Tyler. Intelligent User Interfaces. ACM Press, New York, NY 10036, USA, 1991. ISBN 0-201-50305-0.
- Sow98 Timo Sowa. Ein wissensbasierter Ansatz zur Integration zeitsensitiver Information und seine Anwendung auf die Gestenerkennung. Diplomarbeit, Technische Fakultät der Universität Bielefeld, Bielefeld, Mai 1998.
- SFL99 Timo Sowa, Martin Fröhlich und Marc Latoschik. Temporal Symbolic Integration Applied to a Multimodal System Using Gestures and Speech. In Annelies Braffort, Françoise Forest, Rachid Gherbi, Sylvie Gibet, James Richardson und Daniel Teil, Hrsg., Gesture Workshop '99, Proceedings. Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur - Centre National de la Recherche Scientifique, Université Paris-Sud, März 1999. Vorge stellt während des Gesture Workshop '99 in Gif-sur-Yvette, Frankreich. Im Druck, siehe <http://www.limsi.fr/GW99/>.
- Sri95 R. K. Srihari. Computational models for integrating linguistic and visual information: a survey. Artificial Intelligence Review, 8:349-396, 1995.
- SBC95 U. Steinkühler, W.-J. Beyn und H. Cruse. A simplified MMC model for the control of an arm with redundant degrees of freedom. Neural Processing Letters, 2(4):11-15, 1995.
- Ste94 Ulrich Steinkühler. MMC- Modelle zur Lösung kinematischer Aufgabenstellungen eines redundanten Manipulators. Dissertation, Universität Bielefeld, Fakultät für Mathematik, 1994.
- Ste92 W. Richard Stevens. Advanced Programming in the UNIX Environment. Addison-Wesley Professional Computing Series. Addison-Wesley, Reading, MA, USA, 1992. ISBN 0-201-56317-7.
- Ste94 W. Richard Stevens. The Protocols, Bd. 1 of TCP/IP Illustrated. Addison-Wesley, Reading, MA, USA, 1994.

-
- Ste97 W. Richard Stevens. UNIX Network Programming, Bd. Networking APIs: Sockets and XTI. PTR Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 2. Auflage, 1997. ISBN 0-13-490012-X.
- SCC76 William C. Stokoe, Dorothy C. Casterline und Carl G. Croneberg. A Dictionary of American Sign Language on Linguistic Principles. Linstok Press, Silver Spring, MD, USA, 2. Auflage, Februar 1976.
- Sto72 William C. Stokoe. Semiotics and Human Sign Languages. Approaches to Semiotics. Mouton, The Hague, 1972.
- SZ95 Kerstin Schill und Christoph Zetzsche. A model of visual spatio-temporal memory: The icon revisited. Psychological Research, 57:88-102, 1995.
- Tan94 Andrew S. Tanenbaum. Moderne Betriebssysteme. Hanser Studienbücher der Informatik. Hanser, Prentice-Hall International, München, Wien, Englewood Cliffs, USA, 1994. ISBN 3-446-17472-9, 0-13-517889-4.
- Thó96 Kristinn Rúnar Thórisson. Communicative Humanoids - A Computational Model of Psychosocial Dialogue Skill. Dissertation, Massachusetts Institute of Technology, September 1996.
- TM98 Jochen Triesch und Christoph von der Malsburg. Robotic Gesture Recognition. In Wachsmuth und Fröhlich [WF98], Seiten 233-244. ISBN 3-540-64424-5.
- VRR98 Axel Christian Varchmin, Robert Rae und Helge Ritter. Image Based Recognition of Gaze Direction Using Adaptive Methods. In Wachsmuth und Fröhlich [WF98], Seiten 245-258. ISBN 3-540-64424-5.
- VT94 Virtual Technologies, 2175 Park Blvd., Palo Alto, CA 94306, USA. CyberGlove: User's Manual, August 1994.
- VM99 Christian Vogler und Dimitris Metaxas. Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes. In Annelies Braffort, Françoise Forest, Rachid Gherbi, Sylvie Gibet, James Richardson und Daniel Teil, Hrsg., Gesture Workshop '99, Proceed-
-

- ings. Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur - Centre National de la Recherche Scientifique, Université Paris-Sud, März 1999. Vorgestellt während des Gesture Workshop '99 in Gif-sur-Yvette, Frankreich. Im Druck, Siehe <http://www.limsi.fr/GW99/>.
- Vyg62 L. Vygotsky. Thought and language. MIT Press, Cambridge, Mass., USA, 1962
- Wah91 Wolfgang Wahlster. User and Discourse Models for Multimodal Communication. In Intelligent User Interfaces [ST91], Seiten 45-67. ISBN 0-201-50305-0.
- WAH+89 Wolfgang Wahlster, E. André, M. Hecking und T. Rist. WIP: Knowledgebased Presentation of Information, Project Overview. Technical report, DFKI-German Research Center for Artificial Intelligence, 1989.
- WC95 Ipke Wachsmuth und Yong Cao. Interactive Graphics Design with Situated Agents. In W. Strasser und F. Wahl, Hrsg., Graphics and Robotics, Seiten 73-85. SpringerVerlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1995.
- WG92 David Weimer und S. K. Ganapathy. Interaction Techniques Using Hand Tracking and Speech Recognition. [BD92], Kapitel 7, Seiten 109-126.
- WG89 David Weimar und S. K. Ganapathy. A Synthetic Visual Environment with Hand Gestureing and Voice Input. In SIGCHI '89 Proceedings [ACM89], Seiten 235-240.
- Wex94 Alan Daniel Wexelblat. A Feature-Based Approach to Continuous-Gesture Analysis. Mater's Thesis, Massachusetts Institute of Technology Advanced Human Interface Group, Cambridge, MA, USA, Juni 1994.
- Wex95 Alan Daniel Wexelblat. An Approach to Natural Gesture in Virtual Environments. ACM Transactions on Computer-Human Interaction, 2(3):179-200, 1995.

- Wex98 Alan Daniel Wexelblat. Research Challenges in Gesture: Open Issues and Unsolved Problems. In Wachsmuth und Fröhlich [WF98], Seiten 1-12. ISBN 3-540-64424-5.
- WF98 Ipke Wachsmuth und Martin Fröhlich, Hrsg. Gesture and Sign-Language in Human-Computer Interaction: Proceedings of Bielefeld Gesture Workshop 1997, Nummer 1371 der Lecture Notes in Artificial Intelligence, Berlin Heidelberg New York, 1998. Springer-Verlag.
- WLJ+97 Ipke Wachsmuth, Britta Lenzmann, Tanja Jörding, Bernhard Jung, Marc Latoschik und Martin Fröhlich. A Virtual Interface Agent and its Agency. In Proceedings of the First International Conference on Autonomous Agents, Seiten 516-517, 1997. ACM Press, 1997. ISBN 0-89791-877-0.
- Win81 Peter Winkler, Hrsg. Methoden der Analyse von Face-to-Face-Situationen. J.B. Metzlersche Verlagsbuchhandlung, Stuttgart, 1981. ISBN 3-476-00451-1.
- Win92 Patrick Henry Winston. Artificial Intelligence. Addison-Wesley, Reading, MA, USA, 3. Auflage, 1992. ISBN 0-201-53377-4.
- Win79 David A. Winter. Biomechanics of Human Movement. John Wiley, New York, NY, USA, 1979. ISBN 0-471-03476-2.
- WJ95 M. Wooldridge, N. R. Jennings. Intelligent agents: Theory and practice. Knowledge Engineering Review, 10(2), 1995.
- Yos90 Tsuneo Yoshikawa. Foundations of Robotics: analysis and control. MIT Press, Cambridge, MA, USA, 1990. ISBN 0-262-24028-9.
- ZMR97 Shumin Zhai, Paul Milgram und Anu Rastogi. Anisotropic Human Performance in Six Degree-of-Freedom Tracking: An Evaluation of Three-Dimensional Display and Control Interfaces. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 27(4):518-528, Juli 1997.

Index

A

a priori Wissen 82

Abduktion 83

Arm 83

Hand 84

Schulter 85

Abklingfunktion 133, 141, 146, 149

linear 141

Schwellwertfunktion 141

sigmoid 141

Abklingfunktionen 134

Achse

Radialabduktions- 84

Ulnarabduktions- 84

Adduktion 83

Schulter 85

Advanced Human Interface Group 17

AHIG Klassifikation 17

Agent 77, 102, 143

Adaptivität 157

BDI 77

Effizienzgewinn 156

-ensystem 101

Flexibilität 157

Kontroll- 146

mental states 77

Offenheit 156

proaktiv 77

reaktiv 77

Robustheit 156

Skalierbarkeit 157

sozial 77

System 75

Vielfältigkeit 156

Wiederverwendbarkeit 157

Wissenserwerb 157

Agenten

Dienste 102

Interface- 63

-system 147

Verteiler 102

Agentenmodellierung 157

absichtenorientierte 158

objektorientierte 158

rollenorientierte 158

zustandsorientierte 157

Agentensystem 147

Analog-Digital-Konverter 81

Anatomie 82

ANN 92

Antebrachium 84

Pronation 84

Supination 84

Anteversion

Schulter 85

Anthropometrie 82

Antwort 15

Anytime 87, 89

Apex 21, 23, 62

Applikation

Schicht 73

Architektur

System 71

Arm

Abduktion 83

Art

Gesten- 3

Aufgabenteilung 156

Ausgleichsfunktion 174–175

Ausschlußmechanismus 171

Außenrotation

Schulter 85

autonom 77

B

Ballard

- data 36
- dynamics 36
- optimization 36
- programs 36

Basic Objects 170

Basishypothese 142

Basiswissen 138, 140

Befestigungsproblem 81, 85

belief update 129

benutzerunabhängig 72

Berner System 43, 45

Bibliothek 78, 105, 139

- CLIPS 140
- Software 71

Bildung

- gesten 15

Bildverarbeitungsverfahren 79

Biometrie 79

Birds 171

BNF 130, 132

Bool 92

Brustbein 83

Brustkorb 83

C

C++ 102

cache miss 168

Chomsky-Hierarchie 132

Clavicula 83

CLIPS 138–139, 166

- applyDecayFunction 141
- Bibliothek 140
- Hypothesis 140, 166–167
- Interval 140, 167

newInstance 140

Parser 140

Regel-Konstruktor 141

Timeless-Hypothesis 140, 166

Colossus 1

constraint 97

COOL 139–140, 166

Cubitus 84

Flexion 84

CyberGlove

CG1801 81

D

DAC 81

Daten

Komplexität 99

Datenhandschuh 81

Datenreduktion 151

Datentelegramm 108

degrees of freedom 81

Depression 83

direct manipulation interface 16

Diskurs 73

DNS 107

DOOP 101

dorsal

Extension 84

Dorsalextension 84

Hand 84

E

Echtzeit 89

Echtzeitverarbeitung 127

EDVAC 1

EIA/TIA

232C 106, 171

422 106

485 106

-
- Eingabearchitektur** 64
Elevation 83
Ellenbeuge 84
 Flexion 84
Ellenbogen 87
 Gelenk 84
Ellenbogengelenk 84
ENIAC 1
Ergonomie 90
Erkenntniss
 syntagmatische 94
ERT 171
Exkoskelett 81
Expertensystem 97
Extension
 dorsal 84
- F**
Fakten
 Wissen 97
Fakten- und Regelbasis 140
Fakten-Wissen 97
Felder
 Gebärden 61
 magnetische 80
Fingerspannung 135
 C-Ausdrücke 137
 Näherung 137, 177
 universelle 179, 181
Fingerspannungen 135
fitness 36
Flexion
 Ellenbeuge (Cubitus) 84
 Palmar 84
„Finger-fliegen“ 16
Flock of Birds 106, 108
Freiheitsgrad 81
Fuzzy Logic 147
- G**
Gebärde 11
 -felder 61
Gebärden 61
Gebärdensfelder 61
Gegenwart
 subjektive 32
Gelenk
 Ellenbogen 84
Geste 5, 11
 Erkennungssystem 97
 komplette 58
 -nart 3
 -nbildung 15
 -ntype 3
Gesten 5
 Grammatik 94
Gestenart 3
Gestenbildung 15
Gestenerkennung
 Prototyp A 143
 System 71
Gestenerkennungssystem 97
Gestenfamilie
 coverbal gesture 11
 coverbale Gesten 11–12
Gestengrammatik 94
Gestenkategorien
 autonom 16
 Butterworth'sch 17
 Butterworth'schen Gesten 16
 deiktisch 16–17
 emblematisch 16, 18
 Gestikulieren 15, 18
 ideographisch 15, 18
 ikonisch 15, 17–18
 kinetographisch 15, 18
 Lakoff'sch 16–18
-

- metaphorisch 15, 18
- mimetisch 18
- modalisierend 17
- modus-setzend 18
- objektbezogen 18
- pantomimisch 17–18
- physiographisch 15, 18
- selbstregulierend 18
- sprachmarkierend 15, 18
- symbolisch 16–18
- Taktschlag 17
- Taktschläge 15, 18
- Taktstock 15
- Gesteklasse**
 - EH-Gesten 12
 - open / empty hand gesture 12
- Gesteklassifizierungshierarchie** 20
 - Art 20
 - Familie 20
 - Gattung 20
 - Ordnung 20
- Gestenplatz** 64, 69
- Gestestamm**
 - epistemische Funktion 12
 - ergodische Funktion 12
 - semiotische Funktion 12
- Gestentyp** 3
- Gestikulationsraum** 120
 - Abstand zum Körper 121
 - Bewegungen 123
 - geradlinige 123
 - Bewegungsart 125
 - Geschwindigkeit 125
 - Handflächenorientierung 126
 - Höheneinteilung 120
 - Ortsangabe 120
 - Ruck 125
 - transversale Segmente 121
- Gestlet** 59–60
- Gestlets** 60
- Gleichzeitig** 95
- Glitches** 87
- Grammatik**
 - Gesten 94
- H**
- Hamburger Notationssystem für Gebärdensprachen** 47
- Hamilton–Kreise** 114
 - knotendisjunkt 114
- HamNoSys** 47–54, 61, 119, 121, 158
 - Aktionen 52–53
 - Anfangskonfiguration 49
 - Beidhändige Gesten 53
 - Bewegungsmodalitäten 52
 - Bezeichner 158
 - Fingeransatzrichtung 50
 - Grundformen 49
 - Hand
 - colateral 53
 - dominant 53
 - nicht-dominant 53
 - Handflächenorientierung 50–51
 - Lokationsparameter 51
 - Modifikatoren 51
 - Semantik 119
 - Verbindungsformen 50
- Hand**
 - Abduktion 84
 - Dorsalextension 84
 - Palmarflexion 84
- Handspannung** 134
 - Näherung 177
- Hash-Tabelle** 147
- Hidden-Markov-Modell** 61
- Hierarchie**

-
- Symbolklasse 98
 - HMM** 62
 - HNS+** 98, 119, 130, 144, 158
 - Ausdrücke 130
 - Semantik 132
 - Syntax 130
 - Horner-Schema** 138, 179
 - Hypothese** 93, 97, 141, 144
 - Basishypothese 128
 - nklassen 128
 - eindeutig 129
 - vollständig 129
 - Klasse 95
 - komplexe 128
 - ngewicht 146–147
 - nklasse 144
 - nkonstituenten 144
 - nnamen 144
 - Sprach- 149
 - Hypothesen** 127
 - Hypothesengewicht** 146–147
 - Hypothesenklasse** 95
 - I**
 - ICONIC** 58
 - IFP-GS** 60–62
 - Ikone** 27
 - Implementierung** 138
 - Innenrotation**
 - Schulter 85
 - Integration**
 - multimediale 147
 - Schicht 73
 - sfenster 147
 - temporale 132
 - Wissen 97
 - Integrationsfenster** 147
 - Integrationsrahmen** 126, 132, 138
 - Integrationsrahmens** 126
 - Integrationsverfahren** 93, 126
 - Integrationswissen** 97
 - Integrator** 134, 140, 146
 - Integratoren** 151
 - Intelligenz** 3, 75
 - Invasivität** 85
 - IP** 107
 - ISR** 25
 - J**
 - Jack** 90
 - Jacobi-Matrix** 89
 - K**
 - Kategorisierungssystemen** 15
 - Kendons Kontinuum** 13
 - Embleme 14
 - Gestikulation 13
 - pantomimischen Gesten 14
 - Sprachhafte Gesten 13
 - Zeichensprache 14
 - KiBa–BaKi–Effekt** 33
 - Kinematik** 89
 - klassisch
 - algebraisch 89
 - geometrisch 89
 - iterativ 89
 - Kinetographie Laban, Siehe auch Labeanotation** 39
 - KINOTE** 61–62
 - Klasse** 20
 - Hypothese 95
 - Symbol
 - Hierarchie 98
 - Klassen**
 - Merkmale 91
 - Klassenhierarchie** 96
-

- Klassifikation** 91
 - direkt 92
 - statistische 92
 - Knowledge**
 - Query and Manipulation Language 102
 - Kohärenzprotokolle** 168
 - Kommunikation**
 - natürliche 3
 - Kommunikationsstruktur**
 - aufgabenorientierte 151
 - Kommunkation**
 - Paradigma 77
 - Komplexität**
 - Daten 99
 - Konfidenz** 93
 - Konfidenzwert** 133
 - Kontrollagent** 146
 - Konzepte**
 - vage 119
 - Konzeption** 76
 - System 71
 - Koordinierung**
 - zeitliche 23
 - Körper**
 - Modell 86, 91
 - Körpermodell** 86, 90–91, 109
 - dynamisch 119
 - dynamische 91
 - dynamisches 87, 110
 - geometrisch 119
 - geometrische 91
 - geometrisches 86
 - Längenbezeichner 116
 - Skalierungsfaktoren 164
 - Korrespondenz**
 - Problem 96
 - Korrespondenzproblem** 96
 - KQML** 102, 104
 - Kristallisationspunkt** 14
 - Artificial Intelligence** 1
 - Künstliche Intelligenz** 1
 - künstliches neuronales Netz** 92
 - L**
 - Labanotation** 39
 - Linnaeische Hierarchie**
 - Art 19
 - Familie 19
 - Gattung 19
 - Klasse 19
 - Ordnung 19
 - Reich 19
 - Stamm 19
 - Linnaeischen Hierarchie** 19
 - Locked Basic Objects** 170
 - Logik**
 - Bool 92
 - Luntanumsäule** 84
 - LWP** 104
 - M**
 - Manipulator** 110
 - MAS** 102
 - Medium** 155
 - Mensch-Maschine-Interaktion** 66
 - HCI** 3
 - Mensch-Maschine-Kommunikation** 3
 - Mensch-Maschine-Schnittstelle** 5
 - Merkmal**
 - Schicht 73
 - Merkmale** 99
 - Klassen 91
 - Merkmalerkenner** 119
 - Meßdaten** 176
 - Meßpunkte** 173
-

-
- Meßverfahren** 78
 - Abtastfrequenz 80
 - Genauigkeit 78–79
 - Geschwindigkeit 79
 - Robustheit 87
 - Meta-Rule** 144–145
 - Minsky** 75
 - Mittelhandknochen** 84
 - MMC** 110, 114, 152
 - Gewichtsfaktoren 115
 - Gleichungen 112
 - Komplexitätsabschätzung 113
 - Kopplungsfaktor 113
 - Laufzeit 115
 - Mittelwertsterme 113
 - Randbedingung 115
 - Segmente 112
 - Segmentkette 112
 - Segmentlängen 115
 - MMK** 3
 - Modalität** 155
 - Modell**
 - Körper 86, 91
 - Programmierung 76
 - Radermacher 74
 - Schichten 72, 75
 - Zeit 74
 - modular** 72
 - Modularität** 155
 - monolithisch** 72
 - Moore School of Engineering** 90
 - mounting problem** 81
 - MT-Hot** 105
 - Multiagentensystem** 102
 - Multicast** 107
 - Adressen 107
 - IP 107
 - Multimedia System** 155
 - multimediales System** 155
 - multimodales System** 155
 - Mutex** 168
 - N**
 - Nachrichten** 102, 147
 - Nachzeitig** 95
 - Necker-Würfel** 33
 - Netz**
 - künstlich, neuronal 92
 - NFS** 106
 - NN** 92
 - Notation**
 - Denavit-Hardenberg 88
 - NTP** 109
 - O**
 - Objekt**
 - entferntes 101
 - Observable** 144
 - Observablen** 72
 - Operator**
 - assert 95
 - Organisation**
 - horizontale 72
 - syntagmatische 21
 - vertikale 71
 - P**
 - page fault** 168
 - Palmar**
 - Flexion 84
 - Palmarflexion** 84
 - Hand 84
 - Paradigma**
 - Kommunkation 77
 - Strukturierung 77
 - Parallelität** 156
-

Performanz

Agentensystem 147

Performativ 102

Performative 102

Periodigramm 177

persistenten Instanz 128

Phase

Ausführungs- 21

Rückstellungs- 21

Vorbereitungs- 21

Photometrie 79

Physiologie 82

portabler Code 156

POSIX 104

Nintendo Powerglove 64

Principle of least surprise 156

Problem

Korrespondenz 96

Segmentierung 96

Sicherheits- 101

Zuverlässigkeits- 101

Produktionssystem 138–139

Programmierung

Modell 76

objektorientierte 101

Pronation

Unterarm (Antebrachium) 84

Prototyp 27, 143, 147

provide 102

Pseudogelenk

Ellenbogen 88

Schulter 88

Schulter-Wirbelsäule 88

Pseudoinverse 89

Pseudosegmente 88, 117

Put–That–There 57

R

Radialabduktion

Hand 84

Rahmen 143

Rahmensystem 143

Randbedingung 97

Gleichzeitigkeit 95

komplexe 96

Nachzeitigkeit 95

Randbedingungen 95

zeitliche 149

Rauschen 173

Readily Inferable Meaning 29

RIM 29

Regel

-Meta 144

Wissen 97

Regel-Konstruktor 96, 140–141

PARALLEL 142, 153

SEQUENCE 142, 153

Regelung

System 74

Regelungssystem 74

Reich 20

Relation 96

Spezialisierung 96

Verallgemeinerung 96

Repräsentation 132, 139

adäquate 92

Gesten- 92

Harnad'sche, symbolische 27

ikonische 27

kategoriale 27

interne 25, 91

interne, symbolische 25

lexikalischer Teil 25

prozeduraler Teil 25

semantischer Teil 25

-
- struktureller Teil 25
 - symbolische 93
 - require** 102, 148
 - RETE-Verfahren** 139
 - Retroversion**
 - Schulter 85
 - RGS** 132
 - Richly Grounded Symbol** 29
 - RPC** 106
 - Ruck** 125

 - S**
 - Scapula** 83
 - Schicht**
 - Applikation 73
 - Integration 73
 - Merkmal 73
 - Modell 75
 - Sensor 72, 74
 - Symbolerkennung 73
 - Theorie 73
 - Wissensverarbeitung 73
 - Schichten**
 - Modell 72
 - Schichtenmodell** 75
 - Schlag** 22
 - Schlag Siehe auch Apex** 21
 - Schlüßelbein** 83
 - Schlüsselbeine** 88
 - Schmidt'sche Hierarchie** 26
 - Marken 26
 - Notationssystem 26
 - semantisch endlich differenziert 26
 - Sprache 26
 - Symbolschema 26
 - Symbolsystem 26
 - syntaktisch disjunktiv 26
 - syntaktisch endlich differenziert 26
 - Zeichen 26
 - Schnittstelle**
 - Mensch–Maschine 71
 - Schnittstellen** 156
 - Schulter** 86
 - Abduktion 85
 - Adduktion 85
 - Anteversion 85
 - Außenrotation 85
 - Innenrotation 85
 - Retroversion 85
 - Schulterblatt** 83
 - Schultergelenk** 83, 85
 - Schultergürtel** 82
 - Schwellwertfunktion** 141
 - Segmentierung** 134
 - Problem 96
 - Segmentierungshinweis** 134
 - Segmentierungshinweise** 138
 - Segmentierungsproblem** 96
 - Semantik** 152
 - Sensor**
 - Schicht 72, 74
 - Sensoren**
 - Positions 80
 - Positions- 86, 88, 90
 - virtuelle 90, 152
 - SGIM** 66, 158
 - shared objects** 76, 105
 - Signale** 99
 - Signifikanzschwelle** 133
 - SignStream** 38
 - SMP** 105
 - Society of Mind** 75
 - Sprachanalyse** 147
 - Sprachhypothese** 149
 - Sprechvorgang** 15
 - Stamm** 20
-

- Sternum** 83
- Stimulus** 15
- Strukturierung**
 - Paradigma 77
- Supination** 84
 - Unterarm (Antebrachium) 84
- Switching Hub** 108
- Symbol**
 - definition 145
 - Vorrat 99
- Symboldefinition** 145
- Symbole** 99
 - komplexe 129
- Symbolerkennung**
 - Schicht 73
- Symbolklasse**
 - Hierarchie 98
- Symbolsystem** 25
- SyncWriter** 38
- System**
 - Agenten 75
 - Erkennung
 - Geste 97
 - Gestenerkennung 71
 - multimedial 155
 - multimodal 155
 - multimodales 155
 - Regelung 74
- Systematik** 19
- Systeme**
 - intentionale 77
- T**
- Tabelle**
 - Hash- 147
- Taxonomie** 19, 152
- Theorie**
 - Schicht 73
- Thorax** 83
- Thread** 104–105, 167, 175
 - Ausschlußmechanismus 171
 - Barriere 170
 - deadlock 167, 169
 - kritischer Abschnitt 167
 - Pipelining 170
 - recursive locking 171
- Tick-Tack Effekt** 33
- Tiefpaßfilter** 176
- Transkriptionssystem** 38
- Transskription** 37
- Typ**
 - geste 3
- U**
- UDP** 108
- Ulnarabduktion**
 - Hand 84
- Universelle Rechenmaschine** 1
- Unsicherheit** 93
- Unterarm** 84
 - Pronation 84
 - Supination 84
- V**
- Vagheit** 93
- Verdeckungsproblem** 81
- Verteilungstransparenz** 101
- Vertrauenswert** 93
- Viena** 62, 64
- Virtual Environments and Agents** 62
- Virtuellen Wissensfabrik** 66
- VisBoS** 79
- Visualisierung** 146
- VPL DataGlove** 58

W**Werkzeugebene** 57**WIMP** 6**Wirbelsäule** 88**Wissen** 94

Fakten- 97

Integration 97

kategorisches 129–130

partiell 94

prozedural 139

räumliches 129

Regel- 97

unsicher 92, 120

vage 92, 119

Zeitintervalle 128

Wissensbasis 144**Wissensverarbeitung**

Schicht 73

Wordklassen 148**X****XDR** 106, 108**XTRA** 58**Z****Zeit** 31

Aufeinanderfolge 31

Bänder 35

Dauer 31

Drei-Sekunden-Phänomen 32

Fenster 96

Gegenwart 31

Gleichzeitigkeit 31

Fenster 32

Intervalle 128

Kognition 31

Kontinuität 35

Modell 74

Ordnungsschwelle 32

Reproduktionsexperiment 34

Ungleichzeitigkeit 31

Wahrnehmungsgestalt 32

Wahrnehmungsweise 33

Zeitfenster 96**Zeitkalkül** 95**Zeitstempel** 109, 133

